



Solaris のシステム管理 (基本編)

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-3491-10
2003 年 12 月

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2、AutoClient、JumpStart、Sun Ray、SunBlade、PatchPro、Sun Cobalt、SunOS、Solstice、Solstice AdminSuite、Solstice DiskSuite、Solaris Solve、Java、JavaStation、OpenWindows、NFS、iPlanet、および Netra は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されず、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *System Administration Guide: Basic Administration*

Part No: 817-2874-10

Revision A



031007@6671



目次

はじめに	27
1 Solaris 管理ツール (製品概要)	33
Solaris 管理ツールの新機能	33
Solaris 管理ツールのサポート状況のマトリックス	34
Solaris 9 管理ツールの機能説明	35
Solaris 8 管理ツールの機能説明	36
以前の Solaris 管理ツールの機能説明	38
Solaris 管理コマンドの有効性	38
Solaris 9 システム管理コマンド	39
Solaris 8 システム管理コマンド	40
以前の Solaris 管理コマンドの説明	40
Solaris 管理ツールの詳細情報	41
2 Solaris 管理コンソールの操作 (手順)	43
Solaris 管理コンソール (概要)	43
Solaris 管理コンソールについて	43
Solaris 管理コンソールツール	44
Solaris 管理コンソールを使用する理由	46
Solaris 管理コンソールの構成	46
Solaris 管理コンソールウィンドウの変更	47
Solaris 管理コンソールのマニュアル	48
役割によるアクセス制御の程度	48
スーパーユーザー (ルート) になるか役割を引き受ける	50
▼ スーパーユーザー (ルート) になるか役割を引き受ける方法	50

Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)	52
コンソールに最初にログインした場合	53
プライマリ管理者の役割を作成する	53
▼ 最初の役割 (プライマリ管理者) を作成する方法	55
▼ プライマリ管理者の役割を引き受ける方法	55
Solaris 管理コンソールを起動する	56
▼ スーパーユーザーまたは役割としてコンソールを起動する方法	56
ネームサービス環境で Solaris 管理ツールを使用する (作業マップ)	58
RBAC セキュリティファイル	58
ネームサービス環境で Solaris 管理コンソールを使用するための前提条件	60
管理範囲	60
/etc/nsswitch.conf ファイル	60
▼ 特定環境用のツールボックスを作成する方法	61
▼ ツールボックスにツールを追加する方法	63
▼ ネームサービス環境で Solaris 管理コンソールを起動する方法	64
Solaris 管理コンソールにツールを追加する	64
▼ レガシーツールをツールボックスに追加する方法	64
▼ 別製品のツールをインストールする方法	65
Solaris 管理コンソールの障害追跡	66
▼ Solaris 管理コンソールの障害追跡の方法	66
3 ユーザーアカウントとグループの管理	69
4 ユーザーアカウントとグループの管理 (概要)	71
ユーザーとグループの管理における新機能	71
Solaris 管理コンソールツール群	72
Solaris Directory Services	72
プロジェクトでユーザーおよびリソースを管理する	72
ユーザーアカウントとグループとは	73
ユーザーアカウント管理のガイドライン	74
ネームサービス	74
ユーザー (ログイン) 名	74
ユーザー ID 番号	75
パスワード	78
パスワードの有効期限を設定する	79
ホームディレクトリ	79
ユーザーの作業環境	80

グループを管理するガイドライン	80
ユーザーアカウントとグループを管理するツール	81
Solaris ユーザー管理ツールで実行できる作業	82
Solaris 管理コンソールによるホームディレクトリの管理	86
ユーザーアカウントの変更	86
ユーザーアカウントの削除	87
カスタマイズしたユーザー初期設定ファイルの追加	87
パスワードの管理	87
ユーザーアカウントを無効にする	87
ユーザーアカウントとグループ情報の格納場所	88
passwd ファイルのフィールド	88
shadow ファイルのフィールド	90
group ファイルのフィールド	91
ユーザーの作業環境のカスタマイズ	93
サイト初期設定ファイルの使用方法	95
ローカルシステムへの参照を避ける	95
シェル機能	96
シェル環境	96
PATH 変数	99
ロケール変数	100
デフォルトのファイルアクセス権 (umask)	101
ユーザー初期設定ファイルとサイト初期設定ファイルの例	102
例 — サイト初期設定ファイル	103
5 ユーザーアカウントとグループの管理 (手順)	105
ユーザーアカウントの設定 (作業マップ)	105
ユーザー情報を収集する方法	106
▼ユーザー初期設定ファイルをカスタマイズする方法	107
▼Solaris 管理コンソールのグループツールを使ってグループを追加する方法	109
▼Solaris 管理コンソールのユーザーツールを使ってユーザーを追加する方法	110
例 — Solaris 管理コンソールのグループツールを使ってユーザーを追加する	110
CLI ツールを使ってグループやユーザーを追加する方法	111
▼ユーザーのホームディレクトリを共有する方法	112
▼ユーザーのホームディレクトリをマウントする方法	113
ユーザーアカウントの管理 (作業マップ)	115

Solaris ユーザー登録	115
Solaris Solve へのアクセス	116
Solaris ユーザー登録の問題の障害追跡	117
▼ Solaris ユーザー登録をやり直す方法	117
▼ ユーザー登録を無効にする方法	118
6	サーバーとクライアントサポートの管理 119
7	サーバーとクライアントサポートの管理 (概要) 121
サーバーおよびクライアント管理の新機能	121
ディスクレスクライアントサポート	122
サーバーとクライアントタスクの操作手順	122
サーバー、クライアント、およびアプライアンスとは	123
クライアントサポートとは	124
システムタイプの概要	124
サーバー	125
スタンドアロンシステム	125
ディスクレスクライアント	126
AutoClient システム	126
アプライアンス	127
システムタイプ選択のガイドライン	127
ディスクレスクライアント管理の概要	128
OS サーバーおよびディスクレスクライアントのサポート情報	129
ディスクレスクライアント管理機能	129
OS サーバーに必要なディスク容量	132
8	ディスクレスクライアントの管理 (手順) 135
ディスクレスクライアントの管理 (作業マップ)	135
ディスクレスクライアントの管理	137
▼ ディスクレスクライアントの追加の準備	138
▼ ディスクレスクライアントサポートの OS サービスの追加方法	140
▼ ディスクレスクライアントの追加方法	141
▼ ディスクレスクライアントの起動方法	143
▼ ディスクレスクライアントサポートの削除方法	143
▼ ディスクレスクライアントの OS サービスを削除する方法	144
ディスクレスクライアント OS サービスにパッチを適用する	145
ディスクレスクライアントの OS パッチの表示	145

	▼ ディスクレスクライアントの OS パッチの追加方法	146
	ディスクレスクライアント問題の障害追跡	148
9	システムのシャットダウンとブート	151
10	システムのシャットダウンとブート (概要)	153
	システムのシャットダウンとブートに関する新機能	153
	PXE ネットワークブート	154
	シャットダウンとブートについての参照先	155
	シャットダウンとブートの用語	155
	システムのシャットダウンに関するガイドライン	156
	システムのブートに関するガイドライン	156
	ネットワークからのシステムのブート	157
	システムをシャットダウンする場合	157
	システムをブートする場合	158
11	実行レベルとブートファイル (手順)	161
	実行レベル	161
	システムの実行レベルを確認する方法	162
	/etc/inittab ファイル	163
	例 — デフォルトの inittab ファイル	164
	システムが実行レベル 3 になると実行される処理	165
	実行制御スクリプト	166
	実行制御スクリプトのまとめ	167
	実行制御スクリプトを使用してサービスを起動または停止する	171
	▼ 実行制御スクリプトを使用してサービスを起動または停止する方法	171
	実行制御スクリプトを追加する	172
	▼ 実行制御スクリプトを追加する方法	172
	実行制御スクリプトを無効にする	173
	▼ 実行制御スクリプトを無効にする方法	173
	x86: ブートファイル	174
12	システムのシャットダウン (手順)	177
	システムのシャットダウン	177
	システムシャットダウンコマンド	178
	システムのダウン時間をユーザーに通知する	179

- ▼ システムにログインしているユーザーを知る方法 179
 - ▼ サーバーをシャットダウンする方法 179
 - ▼ スタンドアロンシステムをシャットダウンする方法 183
 - すべてのデバイスの電源を落とす 184
 - ▼ すべてのデバイスの電源を落とす方法 185
- 13 SPARC: システムのブート (手順) 187**
- SPARC: システムのブート (作業マップ) 187
 - SPARC: ブート PROM の使用 189
 - SPARC: システムの PROM リビジョンを確認する方法 189
 - ▼ SPARC: システム上のデバイスを確認する方法 189
 - ▼ SPARC: デフォルトのブートデバイスを変更する方法 191
 - SPARC: システムをリセットする方法 193
 - SPARC: システムのブート 193
 - ▼ SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法 194
 - ▼ SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法 195
 - ▼ SPARC: システムを対話式でブートする方法 196
 - ▼ SPARC: システムをネットワークからブートする方法 197
 - ▼ SPARC: 復元を目的としてシステムを停止する方法 198
 - ▼ SPARC: 復元を目的としてシステムをブートする方法 199
 - ▼ SPARC: カーネルデバッグ (kadb) を使ってシステムをブートする方法 201
 - SPARC: クラッシュダンプを強制してシステムをリポートする 202
 - ▼ SPARC: クラッシュダンプを強制してシステムをリポートする方法 202
- 14 x86: システムのブート (手順) 205**
- x86: システムのブート (作業マップ) 205
 - x86: Solaris Device Configuration Assistant (デバイス構成用補助) のブート 207
 - ▼ x86: Solaris Device Configuration Assistant (デバイス構成用補助) をブートする方法 207
 - x86: システムのブート 207
 - ▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法 208
 - ▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法 209
 - ▼ x86: システムを対話式でブートする方法 210
 - ▼ x86: システムをネットワークからブートする方法 212
 - ▼ x86: 復元を目的としてシステムを停止する方法 213

	▼ x86: 復元を目的としてシステムをブートする方法	213
	▼ x86: カーネルデバッガ (kadb) を使ってシステムをブートする方法	218
x86:	クラッシュダンプを強制してシステムをリブートする	219
	▼ x86: クラッシュダンプを強制してシステムをリブートする方法	219
15	ブートプロセス (参照情報)	221
	SPARC: ブート PROM	221
	SPARC: ブートプロセス	222
	x86: PC BIOS	222
	x86: ブートサブシステム	223
	x86: Solaris リリースのブート	224
	x86: デバイス識別段階で表示される画面	225
	x86: ブート段階で表示されるメニュー	227
	x86: ブートプロセス	228
16	リムーバブルメディアの管理	231
17	リムーバブルメディアの管理 (概要)	233
	リムーバブルメディアの管理における新機能	233
	リムーバブルメディアの管理についての参照先	234
	リムーバブルメディアの機能と利点	234
	自動マウントと手動によるマウントの比較	235
	ボリューム管理を使って行える操作	236
18	リムーバブルメディアへのアクセス (手順)	239
	リムーバブルメディアへのアクセス (作業マップ)	239
	リムーバブルメディアへのアクセス (概要)	240
	リムーバブルメディア名の使用	240
	リムーバブルメディア上のデータにアクセスするためのガイドライン	242
	▼ 新しいリムーバブルメディアドライブを追加する方法	243
	ボリューム管理 (vol1d) の終了と起動	243
	▼ リムーバブルメディア上の情報にアクセスする方法	244
	▼ リムーバブルメディア上の情報をコピーする方法	245
	▼ 音楽用 CD/DVD を再生する方法	245
	▼ リムーバブルメディアが使用中かどうかを調べる方法	247
	▼ リムーバブルメディアを取り出す方法	247

	リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)	248
	▼ ローカルメディアを他のシステムで使用可能にする方法	248
	▼ リモートシステム上のリムーバブルメディアにアクセスする方法	252
19	リムーバブルメディアのフォーマット (手順)	255
	リムーバブルメディアのフォーマット (作業マップ)	255
	リムーバブルメディアのフォーマットの概要	256
	リムーバブルメディアのフォーマットのガイドライン	256
	リムーバブルメディアのハードウェア面での考慮事項	257
	▼ リムーバブルメディアを読み込む方法	259
	▼ リムーバブルメディアをフォーマットする方法 (rmformat)	261
	▼ ファイルシステムの追加用にリムーバブルメディアをフォーマットする方 法	262
	▼ リムーバブルメディア上のファイルシステムを検査する方法	263
	▼ リムーバブルメディア上の不良ブロックを修復する方法	264
	リムーバブルメディアに読み取り/書き込み保護とパスワードによる保護を適用す る	265
	▼ リムーバブルメディアの書き込み保護を有効または無効にする方法	265
	▼ Iomega メディアの読み取り/書き込み保護とパスワードを有効または無効にす る方法	265
20	CD への書き込み (手順)	267
	オーディオ CD やデータ CD の取り扱い	267
	CD メディアに関するよく使われる用語	268
	データ CD やオーディオ CD への書き込み	269
	RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する	270
	▼ RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方 法	270
	CD デバイスを確認する方法	270
	▼ CD メディアをチェックする方法	271
	データ CD を作成する	272
	▼ データ CD 用の ISO 9660 ファイルシステムを作成する方法	272
	▼ マルチセッションのデータ CD を作成する方法	273
	オーディオ CD を作成する	275
	▼ オーディオ CD を作成する方法	275
	▼ オーディオトラックを CD から抽出する方法	276
	▼ CD をコピーする方法	277
	▼ CD-RW メディアを消去する方法	278

21	ソフトウェアの管理	279
22	ソフトウェアの管理 (概要)	281
	Solaris 9 Update リリースの新しいソフトウェア管理機能	281
	pkgadd と patchadd による署名付きパッケージおよびパッチのサポート	282
	prodreg コマンドの拡張機能	282
	Solaris 9 リリースの新しいソフトウェア管理機能	282
	署名付きパッチ	282
	Solaris Product Registry 3.0	283
	パッチアナライザ	283
	Solaris 管理コンソールのパッチマネージャ	283
	ソフトウェア管理作業についての参照先	284
	ソフトウェアパッケージの概要	284
	署名付きパッケージおよびパッチ	285
	ソフトウェアパッケージを管理するためのツール	290
	ソフトウェアパッケージの追加または削除 (pkgadd)	291
	ソフトウェアパッケージの追加にあたっての重要な注意点 (pkgadd)	292
	パッケージの削除に関するガイドライン (pkgrm)	292
	パッケージ追加時のユーザーの対話操作を省略する (pkgadd)	293
	管理ファイルの使用	293
	応答ファイルの使用 (pkgadd)	294
23	ソフトウェアの管理 (手順)	295
	ソフトウェアパッケージを管理するためのコマンド	295
	Solaris Web Start プログラムによるソフトウェアの追加	296
	▼ Solaris Web Start プログラムを使ってソフトウェアをインストールする方 法	297
	Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)	298
	▼ インストールまたはアンインストールしたソフトウェアの情報を Solaris Product Registry の GUI を使って表示する方法	300
	▼ Solaris Product Registry の GUI を使ってソフトウェアをインストールする方 法	301
	▼ Solaris Product Registry の GUI を使ってソフトウェアをアンインストールする 方法	302
	Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作 業マップ)	303
	▼ インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)	304

▼	ソフトウェア属性を表示する方法 (prodreg)	307
▼	ソフトウェアコンポーネント間の依存関係を確認する方法 (prodreg)	309
▼	損傷を受けたソフトウェア製品を特定する方法 (prodreg)	310
▼	ソフトウェアをアンインストールする方法 (prodreg)	313
▼	損傷を受けたソフトウェアをアンインストールする方法 (prodreg)	317
▼	損傷を受けたソフトウェアコンポーネントを再インストールする方法 (prodreg)	320
	署名付きパッケージの追加と削除 (作業マップ)	322
▼	信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)	322
▼	証明書の情報を表示する方法 (pkgadm listcert)	324
▼	証明書を削除する方法 (pkgadm removecert)	325
▼	プロキシサーバーをセットアップする方法	325
▼	署名付きパッケージを追加する方法 (pkgadd)	326
	パッケージコマンドによるソフトウェアパッケージの管理 (作業マップ)	327
▼	ソフトウェアパッケージを追加する方法 (pkgadd)	328
	ソフトウェアパッケージをスプールディレクトリに追加する	331
	インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)	333
▼	インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)	334
	ソフトウェアパッケージを削除する	335
▼	ソフトウェアパッケージを削除する方法 (pkgrm)	335
	Admintool によるソフトウェアパッケージの追加と削除 (作業マップ)	337
▼	Admintool を使ってソフトウェアパッケージを追加する方法	337
▼	Admintool を使ってソフトウェアパッケージを削除する方法	338
24	Solaris パッチの管理 (概要)	341
	パッチについて	341
	署名付きパッチについて	342
	Solaris パッチへのアクセス	342
	Solaris パッチの番号付け	343
	Solaris パッチの管理用ツール	344
	Solaris パッチ管理ツール	344
25	Solaris パッチの管理 (手順)	349
	Solaris 環境でのパッチ管理 (作業マップ)	349
	パッチの追加に必要なディスク容量があるかどうかを特定する	350

署名付きパッチまたは署名のないパッチの選択	351
patchadd コマンドによる署名付きパッチの追加 (作業マップ)	351
信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)	352
▼ 署名付き Solaris パッチのダウンロードと追加を手動で行う方法 (patchadd)	353
▼ 署名付き Solaris パッチのダウンロードと追加を自動で行う方法 (patchadd)	354
smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)	355
Solaris パッチ管理ツールの使用 (smpatch)	356
署名付きパッチツールのパッケージ要件を確認する方法 (smpatch)	357
▼ Solaris パッチ管理ツールのダウンロードおよびインストール方法 (smpatch)	357
▼ Java キーストアに Sun の証明書をインポートする方法	359
▼ Java キーストアのパスワードの変更方法	360
▼ パッチ環境の設定方法 (smpatch)	360
smpatch コマンドによる署名付きパッチの管理 (作業マップ)	362
▼ 署名付きパッチをダウンロードし、追加する方法 (smpatch)	362
▼ 署名付きパッチを削除する方法 (smpatch)	365
署名付きパッチに関する問題の障害追跡 (smpatch)	366
パッチツールのログファイルの表示	366
▼ 隔離されたパッチの解決方法	367
▼ Java キーストアからインポート済みの証明書を削除する方法	368
署名のない Solaris パッチの管理 (作業マップ)	369
署名のない Solaris パッチの情報を表示する	369
Solaris パッチの情報を表示する方法	369
署名のない Solaris パッチの追加	370
▼ 署名のない Solaris パッチのダウンロード方法	370
▼ 署名のない Solaris パッチの追加方法	371
署名のない Solaris パッチの削除	372
▼ 署名のない Solaris パッチを削除する方法	372
26 デバイスの管理	375
27 デバイスの管理 (手順)	377
デバイス管理作業についての参照先	377
デバイスドライバについて	378
デバイスの自動構成	378

自動構成の機能と利点	379
標準サポートされていないデバイスを使用する場合	380
デバイス構成情報の表示	380
driver not attached メッセージ	381
システムデバイスの識別	381
システム構成情報を表示する方法	382
デバイス情報を表示する方法	383
システムへ周辺デバイスを追加する	384
▼ 周辺デバイスを追加する方法	385
▼ デバイスドライバを追加する方法	386
28 デバイスの動的構成 (手順)	387
動的再構成とホットプラグ機能	387
接続点	388
x86: PCI アダプタカードの取り外し	390
cfgadm コマンドによる SCSI ホットプラグ (作業マップ)	391
cfgadm コマンドによる SCSI ホットプラグ	392
▼ SCSI デバイスに関する情報を表示する方法	392
▼ SCSI コントローラの構成を解除する方法	393
▼ SCSI コントローラを構成する方法	393
▼ SCSI デバイスを構成する方法	394
▼ SCSI コントローラを切り離す方法	395
▼ SPARC: SCSI コントローラを接続する方法	396
▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法	396
▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法	397
▼ SPARC: SCSI デバイスを取り外す方法	398
SPARC: SCSI 構成に関する問題の障害追跡	399
▼ 失敗した SCSI 構成解除操作の解決方法	401
cfgadm コマンドによる PCI ホットプラグ (作業マップ)	401
x86: cfgadm コマンドによる PCI ホットプラグ	402
▼ x86: PCI スロット構成情報を表示する方法	402
▼ x86: PCI アダプタカードを取り外す方法	403
▼ x86: PCI アダプタカードを取り付ける方法	403
x86: PCI 構成に関する問題の障害追跡	404
Reconfiguration Coordination Manager (RCM) スクリプトの概要	405
RCM スクリプトについて	406
RCM スクリプトで実行できること	406

	RCM スクリプト処理の動作方法	406
	RCM スクリプトでの作業	407
	アプリケーション開発者 RCM スクリプト (作業マップ)	408
	システム管理者 RCM スクリプト (作業マップ)	409
	RCM スクリプトに名前を付ける	409
	RCM スクリプトのインストールまたは削除	409
	▼ RCM スクリプトのインストール方法	410
	▼ RCM スクリプトの削除方法	410
	▼ RCM スクリプトのテスト方法	410
	テープバックアップ用の RCM スクリプトの例	411
29	USB デバイスの使用 (概要/手順)	415
	USB デバイスの概要	415
	よく使用される USB 関連の略語	417
	USB バスの説明	417
	Solaris 環境における USB について	420
	USB キーボードとマウス	420
	USB ホストコントローラとルートハブ	421
	SPARC: USB 電源管理	422
	USB ケーブルに関するガイドライン	422
	USB 大容量ストレージデバイスの使用 (作業マップ)	423
	USB 大容量ストレージデバイスの使用	424
	非準拠 USB 大容量ストレージデバイスの使用	424
	USB デバイスのホットプラグ	424
	▼ vold を使用して USB 大容量ストレージデバイスを追加する方法	425
	▼ vold を使用しないで USB 大容量ストレージデバイスを追加する方法	425
	▼ vold を使用して USB 大容量ストレージデバイスを取り外す方法	426
	▼ vold を使用しないで USB 大容量ストレージデバイスを取り外す方法	426
	vold を使用してまたは使用しないで USB 大容量ストレージデバイスをマウントする	427
	vold を使用して USB 大容量ストレージデバイスをマウントまたはマウント解除する方法	428
	vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法	429
	▼ USB カメラを追加する方法	430
	USB オーディオデバイスの使用 (作業マップ)	431
	USB オーディオデバイスの使用	431
	複数の USB オーディオデバイスのホットプラグ	433

- ▼ USB オーディオデバイスを追加する方法 433
- ▼ システムの主オーディオデバイスを識別する方法 433
- ▼ 主 USB オーディオデバイスを変更する方法 435
- ▼ 使用されていない USB オーディオデバイスのリンクを削除する方法 436
- USB オーディオデバイスに関する問題の障害追跡 437
 - USB スピーカ障害に対処する 437
- cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ) 438
- cfgadm コマンドを使った USB デバイスのホットプラグ 438
 - USB デバイス情報を表示する方法 439
 - ▼ USB デバイスの構成を解除する方法 441
 - ▼ USB デバイスの構成方法 441
 - ▼ 論理的に USB デバイスを接続解除する方法 441
 - ▼ 論理的に USB デバイスを接続する方法 442
 - ▼ 論理的に USB デバイスのサブツリーを接続解除する方法 442
 - ▼ USB デバイスのリセット方法 443

- 30 デバイスへのアクセス (概要) 445**
 - デバイスへのアクセス 445
 - デバイス情報が作成される方法 445
 - デバイスの管理方法 446
 - デバイス名の命名規則 446
 - 論理ディスクデバイス名 447
 - ディスクサブディレクトリの指定 447
 - スライスの指定 448
 - x86: 直接コントローラでアクセスされるディスク 449
 - SPARC: バス指向コントローラでアクセスされるディスク 449
 - x86: SCSI コントローラでアクセスされるディスク 450
 - 論理テープデバイス名 450
 - 論理リムーバブルメディアデバイス名 451

- 31 ディスクの管理 453**

- 32 ディスクの管理 (概要) 455**
 - Solaris 9 8/03 リリースの新しいディスク管理機能 455
 - SPARC: EFI ディスクラベルによるマルチテラバイトボリュームのサポート 456
 - Solaris 9 リリースの新しいディスク管理機能 460
 - Solaris ボリュームマネージャとソフトパーティション 460

	ディスク管理作業についての参照先	460
	ディスク管理の概要	461
	ディスク関連の用語	461
	ディスクスライスについて	462
	SPARC: ディスクスライス	463
	x86: ディスクスライス	464
	raw データスライスの使用	466
	複数のディスク上のスライス配置	466
	使用するスライスの決定	467
	format ユーティリティ	467
	format ユーティリティを使用する場合	468
	format ユーティリティ使用上のガイドライン	469
	ディスクのフォーマット	470
	ディスクラベルについて	471
	パーティションテーブル	471
	パーティションテーブル情報の表示	472
	ディスクをスライスに分割する	474
	free hog スライスの使用方法	474
33	ディスクの管理 (手順)	475
	ディスクの管理 (作業マップ)	475
	システム上のディスクの確認	476
	▼ システム上のディスクを確認する方法	477
	ディスクのフォーマット	478
	▼ ディスクがフォーマット済みかを調べる方法	479
	▼ ディスクをフォーマットする方法	480
	ディスクスライスの表示	481
	▼ ディスクスライス情報を表示する方法	481
	ディスクラベルの作成と検査	483
	▼ ディスクラベルを作成する方法	483
	▼ ディスクラベルを検査する方法	486
	破損したディスクラベルの復元	487
	▼ 破損したディスクラベルを復元する方法	487
	Sun 製品以外のディスクの追加	489
	format.dat のエントリの作成	490
	▼ format.dat のエントリを作成する方法	490
	SCSI ディスクドライブの自動構成	491

- ▼ SCSI ドライブを自動構成する方法 492
- 欠陥セクターの修復 493
 - ▼ 表面解析を使用して欠陥セクターを調べる方法 494
 - ▼ 欠陥セクターを修復する方法 495
- ディスク管理のヒント 495
 - format セッションのデバッグ 495
 - prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける 496
- 34 SPARC: ディスクの追加 (手順) 499**
 - SPARC: システムディスクまたは二次ディスクの追加 (作業マップ) 499
 - SPARC: システムディスクまたは二次ディスクの追加 500
 - ▼ SPARC: システムディスクを接続してブートする方法 501
 - ▼ SPARC: 二次ディスクを接続してブートする方法 502
 - ▼ SPARC: ディスクスライスとディスクラベルを作成する方法 503
 - ▼ SPARC: ファイルシステムを作成する方法 508
 - ▼ SPARC: システムディスクにブートブロックをインストールする方法 508
- 35 x86: ディスクの追加 (手順) 511**
 - x86: システムディスクまたは二次ディスクの追加 (作業マップ) 511
 - x86: システムディスクまたは二次ディスクの追加 512
 - ▼ x86: システムディスクを接続してブートする方法 513
 - ▼ x86: 二次ディスクを接続してブートする方法 513
 - x86: fdisk パーティションの作成上のガイドライン 515
 - ▼ x86: Solaris fdisk パーティションを作成する方法 515
 - ▼ x86: ディスクスライスとディスクラベルを作成する方法 521
 - ▼ x86: ファイルシステムを作成する方法 523
 - ▼ x86: システムディスクにブートブロックをインストールする方法 523
- 36 format ユーティリティ (参照情報) 525**
 - format ユーティリティを使用する上での推奨事項および要件 525
 - format のメニューとコマンドの説明 526
 - partition メニュー 528
 - x86: fdisk メニュー 529
 - analyze メニュー 530
 - defect メニュー 531
 - format.dat ファイル 532

format.dat	ファイルの内容	533
format.dat	ファイルの構文	533
format.dat	ファイル中のキーワード	533
	パーティションまたはスライステーブル (format.dat)	536
format	ユーティリティの代替データファイルを指定する	536
format	コマンドへの入力規則	537
format	コマンドへ番号を指定する	537
format	コマンドへブロック番号を指定する	537
format	のコマンド名を指定する	538
format	コマンドへディスク名を指定する	539
format	ユーティリティのヘルプを利用する	539
37	ファイルシステムの管理	541
38	ファイルシステムの管理 (概要)	543
	Solaris 9 8/03 リリースの新しいファイルシステム機能	543
	SPARC: マルチテラバイトの UFS ファイルシステムのサポート	543
	Solaris 9 リリースの新しいファイルシステム機能	550
	拡張ファイル属性	550
	UFS スナップショット	550
	UFS 直接入出力の並行処理の向上	551
	mkfs コマンドのパフォーマンスの向上	551
	UDF ファイルシステムの新しい labelit コマンドオプション	551
	ファイルシステム管理作業についての参照先	552
	ファイルシステムの概要	552
	ファイルシステムのタイプ	553
	ディスクベースのファイルシステム	553
	ネットワークベースのファイルシステム	554
	仮想ファイルシステム	555
	ファイルシステム管理用のコマンド	558
	ファイルシステムコマンドによるファイルシステムタイプの判断	559
	汎用コマンドと専用コマンドのマニュアルページ	560
	デフォルトの Solaris ファイルシステム	560
	スワップ空間	561
	UFS ファイルシステム	562
	UFS ログイン	562
	UFS ファイルシステムの計画	563

- UFS 直接入出力 564
- ファイルシステムのマウントとマウント解除 564
 - マウントされたファイルシステムテーブル 566
 - 仮想ファイルシステムテーブル 567
 - NFS 環境 568
 - 自動マウントまたは AutoFS 568
- ファイルシステムのタイプを調べる 569
 - ファイルシステムのタイプを調べる方法 569

- 39 ファイルシステムの作成 (手順) 571**
 - UFS ファイルシステムの作成 571
 - newfs コマンドのデフォルトのパラメータ 572
 - ▼ UFS ファイルシステムを作成する方法 573
 - 一時ファイルシステム (TMPFS) の作成 574
 - ▼ TMPFS ファイルシステムを作成する方法 575
 - ループバックファイルシステム (LOFS) の作成 576
 - ▼ LOFS ファイルシステムを作成する方法 576

- 40 ファイルシステムのマウントとマウント解除 (手順) 579**
 - ファイルシステムのマウントの概要 579
 - ファイルシステムのマウントとマウント解除に使用するコマンド 580
 - 汎用マウントオプション 581
 - /etc/vfstab ファイルのフィールドの説明 583
 - ファイルシステムのマウント 584
 - どのファイルシステムがマウントされているかを調べる方法 585
 - ▼ /etc/vfstab ファイルにエントリを追加する方法 585
 - ▼ 1 つのファイルシステムをマウントする方法 (/etc/vfstab ファイル) 586
 - ▼ UFS ファイルシステムのマウント方法 (mount コマンド) 588
 - ▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド) 589
 - ▼ NFS ファイルシステムのマウント方法 (mount コマンド) 590
 - ▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド) 591
 - ファイルシステムのマウント解除 592
 - ファイルシステムをマウント解除する場合の前提条件 593
 - ファイルシステムのマウント解除を確認する方法 593
 - ▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法 593

▼1つのファイルシステムをマウント解除する方法 594

- 41 **CacheFS** ファイルシステムの使用 (手順) 597
 - CacheFS ファイルシステムの概観 (作業マップ) 597
 - CacheFS ファイルシステムの概要 598
 - CacheFS ファイルシステムの機能 598
 - CacheFS ファイルシステムの構造と動作 599
 - CacheFS ファイルシステムの作成とマウント (作業マップ) 600
 - ▼キャッシュを作成する方法 601
 - ファイルシステムをキャッシュにマウントする 602
 - ▼CacheFS ファイルシステムのマウント方法 (mount) 602
 - ▼CacheFS ファイルシステムのマウント方法 (/etc/vfstab) 604
 - ▼CacheFS ファイルシステムのマウント方法 (AutoFS) 605
 - CacheFS ファイルシステムの保守 (作業マップ) 606
 - CacheFS ファイルシステムの保守 607
 - CacheFS ファイルシステムの変更 607
 - ▼CacheFS ファイルシステムに関する情報を表示する方法 608
 - CacheFS ファイルシステムの整合性チェック 608
 - ▼必要に応じてキャッシュの整合性チェックを指定する方法 609
 - ▼CacheFS ファイルシステムを削除する方法 609
 - ▼CacheFS ファイルシステムの整合性をチェックする方法 611
 - CacheFS ファイルシステムのバックアップ (作業マップ) 612
 - CacheFS ファイルシステムのバックアップ 612
 - キャッシュにファイルをバックアップする方法 613
 - バックアップされたファイルの情報を表示する方法 614
 - バックアップリストの使用 615
 - バックアップリストを作成する方法 615
 - バックアップリストを使ってファイルをキャッシュにバックアップする方法 616
 - キャッシュからファイルまたはバックアップリストのバックアップを解除する 616
 - キャッシュからファイルまたはバックアップリストのバックアップを解除する方
法 617
 - cachefspack エラーの障害追跡 618
 - CacheFS の統計情報の収集 (作業マップ) 621
 - CacheFS の統計情報の収集 622
 - CacheFS ログインを設定する方法 623
 - CacheFS ログファイルの場所を調べる方法 624
 - CacheFS ログインを停止する方法 624
 - 作業セット (キャッシュ) のサイズを表示する方法 625

	CacheFS の統計情報の表示	626
	CacheFS 統計情報を表示する方法	626
42	追加スワップ空間の構成 (手順)	629
	スワップ空間について	629
	スワップ空間と仮想メモリー	630
	スワップ空間と TMPFS ファイルシステム	630
	ダンプデバイスとしてのスワップ空間	631
	スワップ空間の追加が必要かどうかを調べる方法	631
	スワップ関連のエラーメッセージ	632
	TMPFS 関連のエラーメッセージ	632
	スワップ空間の割り当て方法	632
	/etc/vfstab ファイル	633
	スワップ空間の計画	633
	スワップリソースの監視	634
	スワップ空間の追加	635
	スワップファイルの作成	636
	▼スワップファイルを作成して使用可能にする方法	636
	スワップファイルの削除	638
	▼不要になったスワップ空間を削除する方法	638
43	UFS ファイルシステムの整合性チェック (手順)	639
	ファイルシステムの整合性	639
	ファイルシステムの状態はどのように記録されるか	640
	fsck コマンドでチェックして修復される内容	642
	不整合が発生する原因	642
	整合性がチェックされる UFS 構成要素	643
	fsck 要約メッセージ	648
	UFS ファイルシステムを対話式でチェックして修復する	649
	▼ファイルシステムのチェックが必要かを調べる方法	649
	▼ファイルシステムを対話式でチェックする方法	650
	UFS ファイルシステムの修復	651
	▼UFS ファイルシステムを修復する方法	651
	fsck コマンドで修復できない UFS ファイルシステムの修正	652
	不正なスーパーブロックの復元	653
	▼不正なスーパーブロックを復元する方法	653
	fsck コマンドの構文とオプション	655

44	UFS ファイルシステム (参照情報)	657
	ルート (/) と /usr ファイルシステムのデフォルトディレクトリ	657
	プラットフォームに依存するディレクトリ	665
	UFS ファイルシステムのシリンダグループの構造	666
	ブートブロック	666
	スーパーブロック	666
	i ノード	667
	データブロック	668
	空きブロック	668
	カスタムファイルシステムパラメータ	669
	論理ブロックサイズ	669
	フラグメントサイズ	670
	最小空き容量	671
	回転待ち	671
	最適化のタイプ	671
	i ノード数 (ファイルの数)	672
	UFS ファイルとファイルシステムの最大サイズ	673
	UFS サブディレクトリの最大数	673
	カスタマイズされたファイルシステムを作成するためのコマンド	673
	newfs コマンドの構文、オプション、引数	673
	汎用 mkfs コマンド	676
45	ファイルとファイルシステムのバックアップおよび復元	677
46	ファイルシステムのバックアップと復元 (概要)	679
	ファイルシステムのバックアップと復元の新機能	679
	UFS スナップショット	679
	バックアップと復元についての参照先	680
	ファイルシステムのバックアップと復元とは	680
	ファイルシステムをバックアップする理由	681
	バックアップを作成するファイルシステムの計画	682
	バックアップタイプの選択	684
	テープデバイスの選択	684
	ファイルシステムのバックアップおよび復元の概要 (作業マップ)	685
	バックアップスケジュールを設定する際のガイドライン	686
	バックアップ頻度	687
	ダンプレベルを使用して増分バックアップを作成する	687

	バックアップスケジュールの例	688
	例 — 日単位累積、週単位累積バックアップ	688
	例 — 日単位累積、週単位増分バックアップ	689
	例 — 日単位増分、週単位累積バックアップ	690
	例—サーバーの月単位バックアップスケジュール	691
	バックアップスケジュールに関するその他の推奨事項	694
47	ファイルとファイルシステムのバックアップ (手順)	697
	ファイルとファイルシステムのバックアップ (作業マップ)	697
	ファイルシステムバックアップの実行準備	698
	▼ ファイルシステム名を検索する方法	698
	▼ 完全バックアップに必要なテープ数を判別する	699
	ファイルシステムのバックアップ	699
	▼ ファイルシステムのバックアップをテープに作成する方法	700
48	UFS スナップショットの使用 (手順)	707
	UFS スナップショットの使用 (作業マップ)	707
	UFS スナップショットの概要	708
	なぜ UFS スナップショットを使用するか	708
	UFS スナップショットのパフォーマンス上の問題	709
	UFS スナップショットの作成と削除	709
	▼ UFS スナップショットを作成する方法	710
	▼ UFS スナップショットの情報を表示する方法	711
	UFS スナップショットの削除	712
	▼ UFS スナップショットを削除する方法	712
	UFS スナップショットのバックアップ	713
	▼ UFS スナップショットの完全バックアップを作成する方法 (ufsdump)	713
	▼ UFS スナップショットの増分バックアップの作成方法 (ufsdump)	714
	▼ UFS スナップショットのバックアップ方法 (tar)	714
	UFS スナップショットのバックアップからのデータの復元	715
49	ファイルとファイルシステムの復元 (手順)	717
	ファイルおよびファイルシステムのバックアップを復元する (作業マップ)	717
	ファイルとファイルシステムを復元するための準備	718
	ファイルシステム名の確認	719
	必要なテープデバイスのタイプの決定	719
	テープデバイス名の決定	719

ファイルシステムの復元	719
▼使用するテープを決定する方法	720
▼対話式でファイルを復元する方法	721
▼例 — 特定のファイルを復元する	723
▼ファイルシステム全体を復元する方法	725
▼ルート (/) と /usr を復元する方法	728
50 UFS バックアップおよび復元コマンド (参照情報)	731
ufsdump コマンドの機能	731
デバイス特性の判断	731
メディアの終わりの検出	732
ufsdump データのコピー	732
/etc/dumpdates ファイルの役割	732
バックアップデバイス (<i>dump-file</i>) 引数	733
バックアップを作成するファイルを指定する	735
テープの性質を指定する	735
ufsdump の制限	735
ufsdump コマンドのオプションと引数	736
ufsdump のデフォルトオプション	736
ufsdump コマンドのオプション	736
ufsdump とセキュリティに関する注意事項	739
ufsrestore コマンドのオプションと引数	739
51 UFS ファイルとファイルシステムのコピー (手順)	745
ファイルシステムをコピーするためのコマンド	746
ファイルシステムをディスク間でコピーする	747
ファイルシステムのリテラルコピーを作成する	747
▼ディスクをコピーする方法 (dd)	748
cpio を使用してファイルシステム間でディレクトリをコピーする	750
▼ファイルシステム間でディレクトリをコピーする方法 (cpio)	751
ファイルとファイルシステムをテープにコピーする	752
tar を使用してファイルをテープにコピーする	753
▼ファイルをテープにコピーする方法 (tar)	754
▼テープ上のファイルのリストを表示する方法 (tar)	755
▼テープからファイルを取り出す方法 (tar)	755
pax コマンドを使用してファイルをテープにコピーする	756
▼ファイルをテープにコピーする方法 (pax)	756

cpio コマンドを使用してファイルをテープにコピーする	757
▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)	757
▼ テープ上のファイルのリストを表示する方法 (cpio)	759
▼ テープからすべてのファイルを取り出す方法 (cpio)	759
▼ テープから特定のファイルを取り出す方法 (cpio)	760
ファイルをリモートテープデバイスにコピーする	761
▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd)	761
▼ ファイルをリモートテープデバイスから取り出す方法	763
ファイルとファイルシステムをフロッピーディスクにコピーする	764
ファイルをフロッピーディスクにコピーする際の注意事項	764
▼ ファイルを1枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)	764
▼ フロッピーディスク上のファイルのリストを表示する方法 (tar)	765
▼ ファイルをフロッピーディスクから取り出す方法 (tar)	766
ファイルを複数のフロッピーディスクにアーカイブする方法	767
ファイルを別のヘッダー形式でコピーする	767
SunOS の旧リリース用のアーカイブを作成する方法	768
bar コマンドで作成したファイルを取り出す	768
▼ bar ファイルをフロッピーディスクから取り出す方法	769
52 テープドライブの管理 (手順)	771
使用するメディアの選択	771
バックアップデバイス名	772
テープドライブの巻き戻しオプションを指定する	773
テープドライブに別の密度を指定する	774
テープドライブの状態を表示する	774
▼ テープドライブの状態を表示する方法	774
磁気テープカートリッジの取り扱い	775
磁気テープカートリッジのたるみを直す方法	775
磁気テープカートリッジを巻き戻す方法	776
ドライブの管理とメディア処理のガイドライン	776
索引	777

はじめに

本書『Solaris のシステム管理 (基本編)』は、Solaris™ システム管理に関する重要な情報を提供するマニュアルの一部です。本書は、SPARC® および x86 の両方のプラットフォームにおけるシステム管理について解説しています。

本書では、次のことを前提としています。

- SunOS 5.9 オペレーティングシステムがすでにインストールされていること
- 使用するネットワークソフトウェアの設定が完了していること

SunOS 5.9 オペレーティングシステムは、多くの機能と Solaris 共通デスクトップ環境 (CDE) を含む Solaris 製品の一部です。また、SunOS 5.9 は、AT&T System V リリース 4 オペレーティングシステムに準拠しています。

システム管理者にとって重要な Solaris 9 リリースの新機能については、各章のはじめにある新機能に関する節を参照してください。

注 - Solaris オペレーティングシステムは、SPARC と x86 という 2 種類のハードウェア (プラットフォーム) で動作します。また、Solaris オペレーティング環境は、64 ビットのアドレス空間でも 32 ビットのアドレス空間でも動作します。このマニュアルで説明する情報は、章、節、注、箇条書き項目、図、表、例、コード例などで特に明記しないかぎり、両方のプラットフォーム、およびアドレス空間に適用されます。

対象読者

このマニュアルは、Solaris 9 リリースを実行するシステムの管理者を対象にしています。このマニュアルを読むには、UNIX® のシステム管理について 1～2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも知識の習得に役立ちます。

Solaris システム管理マニュアルセットの構成

システム管理マニュアルセットに含まれる各マニュアルとその内容は、次のとおりです。

マニュアル名	内容
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、リムーバブルメディア、ソフトウェアの管理 (パッケージとパッチ)、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Solaris のシステム管理 (上級編)』	印刷サービスの管理、端末とモデムの設定、システムリソースの管理 (ディスク割り当て、アカウントティング、および crontab ファイルの管理)、システムプロセスの管理、および Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク、IPv4 と IPv6、DHCP、IP セキュリティ、モバイル IP、および IP ネットワークのマルチパス化
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、および LDAP のネーミングとディレクトリサービスの管理
『Solaris のシステム管理 (ネーミングとディレクトリサービス : FNS、NIS+ 編)』	FNS および NIS+ のネーミングとディレクトリサービスの管理
『Solaris のシステム管理 (資源管理とネットワークサービス)』	リソース管理、リモートファイルシステム、メール、SLP、PPP
『Solaris のシステム管理 (セキュリティサービス)』	監査、PAM、RBAC、および SEAM

このリリースの Solaris オペレーティングシステムに含まれているオープンソースソフトウェアのライセンス用語、特性、著作権の記述を表示する場合、デフォルトのパスは `/usr/share/src/freeware-name` または `/usr/sfw/share/src/freeware-name` です。Solaris オペレーティング環境がデフォルト以外の場所にインストールされている場合は、所定のパスを修正して、インストールした場所にあるファイルにアクセスします。

Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、http://docs.sun.com です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用しません。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>system%</code>
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	<code>system% su</code> <code>password:</code>
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、 <code>rm filename</code> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	<code>sun% grep '^#define \</code> <code>XV_VERSION_STRING'</code>

コード例は次のように表示されます。

- C シェル

```
machine_name% command y|n [filename]
```

- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

一般規則

このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が適宜併記されています。

このマニュアルでは、「x86」という用語は、Intel 32 ビット系列のマイクロプロセッサチップ、および AMD が提供する互換マイクロプロセッサチップを意味します。

- このマニュアル中の手順を実行したり、例 (コマンド入力、コードなど) を使用する場合には、二重引用符 ("), 左一重引用符 ('), 右一重引用符 (') をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。
- /sbin、/usr/sbin、/usr/bin、/etc ディレクトリにあるコマンドについては、このマニュアルでは絶対パス名で表記していない場合があります。ただし、それ以外のあまり一般的でないディレクトリにあるコマンドについては、このマニュアル中の例では絶対パスで表記します。

- このマニュアル中の例は、SunOS ソフトウェアが標準的にインストールされていることを前提としています。つまり、バイナリ互換パッケージがインストールされていることや /usr/ucb が検索パスに設定されていることは、前提としていません。



注意 - /usr/ucb を検索パスに設定する場合は、パスの一番最後に設定してください。ps コマンドや df コマンドなどは、SunOS コマンドと /usr/ucb コマンドとで形式やオプションがそれぞれ異なります。

第 1 章

Solaris 管理ツール (製品概要)

この章では、Solaris 管理ツールの製品概要について説明します。

- 33 ページの「Solaris 管理ツールの新機能」
- 34 ページの「Solaris 管理ツールのサポート状況のマトリックス」
- 35 ページの「Solaris 9 管理ツールの機能説明」
- 36 ページの「Solaris 8 管理ツールの機能説明」
- 38 ページの「以前の Solaris 管理ツールの機能説明」
- 38 ページの「Solaris 管理コマンドの有効性」
- 41 ページの「Solaris 管理ツールの詳細情報」

Solaris 管理ツールの新機能

Solaris 9 リリースでは、次のツールが新たに追加または変更されました。

- ディスクレスクライアントサポート
- Solaris DHCP
- リソース管理
- Solaris 管理コンソールツール群
- Solaris ボリュームマネージャ (旧 Solstice™ DiskSuite)
- Solaris パッチマネージャ
- Product Registry

次の表に、各ツールの概要と詳しい説明の参照先を示します。

表 1-1 Solaris 9 で新たに追加または変更された Solaris 管理ツール

Solaris 管理ツール	説明	参照先
ディスクレスクライアントサポート	ディスクレスクライアントシステムを管理するためのコマンド行インタフェースを提供する	第 8 章
リソース管理	利用可能なシステムリソースをアプリケーションでどのように使用するかを管理できる	『Solaris のシステム管理 (資源管理とネットワークサービス)』
Solaris DHCP	ネットワークで DHCP を管理する際のパフォーマンス、容量、柔軟性が改善される	『Solaris のシステム管理 (IP サービス)』の「Solaris DHCP (概要)」
Solaris 管理コンソール ¹	各種の GUI ベースシステム管理ツールの起動ポイントとして機能する	このマニュアルとコンソールのオンラインヘルプ
Solaris ボリュームマネージャ (旧 Solstice™ DiskSuite)	信頼性の高い記憶装置の管理を行い、Solaris 管理コンソールから起動される。コマンド行インタフェースも使用できる	『Solaris ボリュームマネージャの管理』
Solaris パッチマネージャ	このツールを使うと、署名付きまたは署名なしのパッチをシステムに追加できる	第 24 章
Solaris Product Registry	prodreg コマンドには、Solaris Product Registry グラフィカルユーザーインタフェースと同様に、browse、info、unregister、uninstall の各サブコマンドが含まれている	303 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)」

¹ このツールを Sun Management Center (SunMC) と混同しないでください。Sun Management Center 製品については、<http://www.sun.com/solaris/sunmanagementcenter/docs> を参照してください。

Solaris 管理ツールのサポート状況のマトリックス

この節では、主としてユーザー、グループ、クライアント、ディスク、プリンタ、シリアルポートの管理に使われるツールについて説明します。

次の表に、Solaris 管理用の各種 GUI ツールと、それらの現在のサポート状況を示します。

表 1-2 Solaris 管理ツールのサポート状況を示すマトリックス

	Solaris 2.6 以前のリリース	Solaris 7	Solaris 8	Solaris 9
admintool	O	O	O	O
Solstice AdminSuite 2.3	O	O	X	X
Solstice AdminSuite 3.0	O (Solaris 2.6 リリースのみ)	O	O	X
Solaris Management Tools 1.0	O	O	O	X
Solaris Management Tools 2.0	X	X	O (Solaris 8 01/01、4/01、7/01、10/01、2/02 の各リリースのみ)	X
Solaris Management Tools 2.1	X	X	X	O

コンソールにテキストベースの端末を使用しているシステムで管理作業を行う場合は、Solaris 管理コンソールのコマンドを使用してください。詳細については、表 1-6 を参照してください。

Solaris 9 管理ツールの機能説明

次の表に、Solaris 9 リリースで使用できるツールを示します。

表 1-3 Solaris 9 管理ツールの機能説明

機能またはツール	admintool でのサポート	Solaris 管理コンソール 2.1 でのサポート
AutoClient サポート	なし	なし
コンピュータとネットワークツール	なし	あり

表 1-3 Solaris 9 管理ツールの機能説明 (続き)

機能またはツール	admintool でのサポート	Solaris 管理コンソール 2.1 でのサポート
ディスクレスクライアントサポート	なし	あり。ディスクレスクライアント CLI を使用できる
ディスクツール	なし	あり
拡張ディスクツール (Solaris ボリュームマネージャ)	なし	あり
ジョブスケジューラ	なし	あり
ログビューア	なし	あり
メール別名サポート	なし	あり
マウントと共有ツール	なし	あり
ネームサービスサポート	なし	ユーザー、グループ、ネットワーク情報に対してのみ
パッチツール	なし	あり
パフォーマンスツール	なし	あり
プリンタサポート	あり	Solaris Print Manager は個別に使用できる
プロジェクトツール	なし	あり
RBAC サポート	なし	あり
RBAC ツール	なし	あり
シリアルポートツール	あり	あり
ソフトウェアパッケージツール	あり	なし
システム情報ツール	なし	あり
ユーザー/グループツール	あり	あり

Solaris 8 管理ツールの機能説明

次の表に、Solaris 8 リリースと Solaris 8 の各種アップデートリリース (更新版) で使用できるツールを示します。

表 1-4 Solaris 8 管理ツールの機能説明

機能またはツール	admintool でのサ ポート	Solstice AdminSuite 3.0 で のサポート (Solaris 8、Solaris 8 6/00、 Solaris 8 10/00 の み)	Solaris Management Console 1.0 でのサ ポート	Solaris Management Console 2.0 でのサ ポート (Solaris 8 1/01、4/01、7/01、 10/01、2/02 のみ)
AutoClient/ディス クレスクライアント サポート	なし	なし (ただし、 AutoClient CLI は個別に使用で きる)	なし	なし (ただし、 ディスクレスク ライアント CLI と AutoClient CLI は個別に使 用できる)
ディスクツール	なし	なし	なし	あり
ジョブスケジューラ	なし	なし	なし	あり
ログビューア	なし	あり	なし	あり
メール別名サポート	なし	あり	なし	あり
マウントと共有ツ ール	なし	あり	なし	あり
ネームサービスサ ポート	なし	あり	なし	ユーザー、グ ループ、ネット ワーク情報に対 してのみ
プリンタサポート	あり	Solaris Print Manager は使用 できる	あり	なし (ただし、 Solaris Print Manager は使用 できる)
ソフトウェア パッケージツール	あり	なし	あり	なし
RBAC サポート	なし	あり (権利のみ サポート)	なし	あり
RBAC ツール	なし	RBAC CLI は個 別に使用できる	なし	あり
シリアルポートツ ール	あり	あり	あり	あり
ユーザー/グループ ツール	あり	あり	あり	あり

以前の Solaris 管理ツールの機能説明

次の表に、Solaris 8 より前のリリースで使用できるツールを示します。

表 1-5 以前の Solaris 管理ツールの機能説明

機能またはツール	admintool でのサポート	Solstice AdminSuite 2.3 でのサポート	Solstice AdminSuite 3.0 でのサポート(Solaris 2.6 のみ)
AutoClient/ディスクレスクライアントサポート	なし	あり	なし (ただし、AutoClient CLI は個別に使用できる)
ディスクツール	なし	あり	なし
ログビューア	なし	なし	あり
メール別名サポート	なし	あり	あり
マウントと共有ツール	なし	あり	あり
ネームサービスサポート	なし	あり	あり
プリンタサポート	あり	あり	Solaris Print Manager は使用できる
RBAC サポート	なし	なし	あり (権限のみサポート)
RBAC ツール	なし	なし	RBAC CLI は個別に使用できる
シリアルポートツール	あり	あり	あり
ユーザー/グループツール	あり	あり	あり

Solaris 管理コマンドの有効性

次の一連の表に、Solaris 管理ツールと同じ作業を実行するコマンドを示します。ディスクレスクライアントサポートについては、第 8 章を参照してください。

Solaris 9 システム管理コマンド

次の表に、Solaris 管理ツールと同じ機能を備えたコマンドを示します。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。コマンドの中には、ローカルシステム専用のものもあります。また、ネームサービス環境で動作するものもあります。それぞれのマニュアルページで、-D オプションを参照してください。

表 1-6 Solaris 管理コマンドの説明

コマンド	説明	マニュアルページ
smc	Solaris 管理コンソールを起動する	smc (1M)
smcron	crontab ジョブを管理する	smcron (1M)
smdiskless	ディスクレスクライアントサポートを管理する	smdiskless (1M)
smexec	exec_attr データベースのエントリを管理する	smexec (1M)
smgroup	グループエントリを管理する	smgroup (1M)
smlog	WBEM ログファイルを管理/表示する	smlog (1M)
smmultiuser	複数のユーザーアカウントに対する一括操作を管理する	smmultiuser (1M)
smosservice	OS サービスとディスクレスクライアントサポートを追加する	smosservice (1M)
smprofile	prof_attr および exec_attr データベースのプロファイルを管理する	smprofile (1M)
smrole	役割アカウントの役割とユーザーを管理する	smrole (1M)
smserialport	シリアルポートを管理する	smserialport (1M)
smuser	ユーザーエントリを管理する	smuser (1M)

次の表は、コマンド行から RBAC を管理するためのコマンドです。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。なお、これらのコマンドは、ネームサービス環境で RBAC 情報を管理する場合には使用できません。

表 1-7 RBAC コマンドの説明

コマンド	説明	参照ページ
auths	ユーザーに付与されている権限を表示する	auths (1)
profiles	ユーザーの実行プロファイルを表示する	profiles (1)
roleadd	システムに新しい役割を追加する	roleadd (1M)
roles	ユーザーに付与されている役割を表示する	roles (1)

次の表は、コマンド行からユーザー、グループ、RBAC 機能を管理するためのコマンドです。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。なお、これらのコマンドは、ネームサービス環境でユーザーやグループの情報を管理する場合には使用できません。

表 1-8 Solaris ユーザー/グループコマンドの説明

コマンド	説明	参照ページ
useradd, usermod, userdel	ユーザーを追加、変更、削除する	useradd (1M)、 usermod (1M)、 userdel (1M)
groupadd, groupmod, groupdel	グループを追加、変更、削除する	groupadd (1M)、 groupmod (1M)、 groupdel (1M)

Solaris 8 システム管理コマンド

Solaris 8 リリースでは、表 1-7 および 表 1-8 に記載されているコマンドをすべて使用できます。

以前の Solaris 管理コマンドの説明

次の表は、Solstice AdminSuite™ 2.3 と Solstice AutoClient™ 2.3 の GUI ツールと同等の機能を備えたコマンドです。これらのコマンドを使用する場合は、スーパーユーザーになるか、sysadmin グループのメンバーになる必要があります。

注 – Solstice AdminSuite 2.3 と Solstice AutoClient 2.3 のコマンドのマニュアルページは、オンラインで使用できません。このため、これらのマニュアルページを見る場合は、Solstice AdminSuite 2.3 および Solstice AutoClient 2.3 ソフトウェアにアクセスできなければなりません。

以前の Solaris リリースでは、表 1-8 に記載されているコマンドもすべて使用できます。

表 1-9 Solstice AdminSuite™ 2.3/Solstice AutoClient™ 2.1 のコマンドの説明

コマンド	説明	参照先
admhostadd、 admhostmod、 admhostdel、 admhostls	AdminSuite ソフトウェアを使って設定したクライアント/サーバーシステムのサポートを追加、変更、削除、表示する	『Solstice AdminSuite 2.3 管理者ガイド』と『Solstice AutoClient 2.1 管理者ガイド』
admuseradd、 admusermod、 admuserdel、 admuserls、 admgroupadd、 admgroupmod、 admgroupdel、 admgrouppls	ユーザーやグループを追加、変更、削除、表示する	『Solstice AdminSuite 2.3 管理者ガイド』

Solaris 管理ツールの詳細情報

次の表に、Solaris 管理ツールの詳細情報の参照先を示します。

表 1-10 Solaris 管理ツールの詳細情報

ツール	有効な Solaris のバージョン	参照先
Solaris 管理コンソール 2.1 のツール群	Solaris 9 リリース	このマニュアルとコンソールのオンラインヘルプ
Solaris Management Console 2.0 のツール群	Solaris 8 1/01、4/01、7/01、10/01、2/02 の各リリース	Solaris 管理コンソールのオンラインヘルプ
Solaris Management Console 1.0 のツール群	Solaris 2.6、Solaris 7、Solaris 8 の各リリース	『Solaris Easy Access Server 3.0 のインストール』

表 1-10 Solaris 管理ツールの詳細情報 (続き)

ツール	有効な Solaris のバージョン	参照先
admintool	Solaris 9、Solaris 8、以前の Solaris リリース	admintool (1M)
AdminSuite 2.3	Solaris 2.4、Solaris 2.5、Solaris 2.5.1、Solaris 2.6、Solaris 7 の各リリース	『Solstice AdminSuite 2.3 管理者ガイド』
AminSuite 3.0	Solaris 8、Solaris 8 6/00、Solaris 8 10/00 の各リリース	『Solaris Easy Access Server 3.0 のインストール』
AutoClient 3.0.1	Solaris 8 リリース	ローカルサービスプロバイダに連絡する
	Solaris 9 リリース – フラッシュ (Flash) インストール機能の使用を検討する	『Solaris 9 12/03 インストールガイド』
ディスクレスクライアント CLI	Solaris 8 1/01、4/01、7/01、10/01、2/02、Solaris 9 の各リリース	第 8 章

第 2 章

Solaris™ 管理コンソールの操作 (手順)

この章では、システム管理作業を行うための Solaris 管理ツールの概要について説明します。この章は、Solaris 管理コンソールの起動、コンソールで使用する役割によるアクセス制御 (RBAC) の設定、ネームサービス環境での Solaris 管理ツールの操作から構成されています。

Solaris 管理コンソールによるシステム管理作業の実行手順については、以下を参照してください。

- 52 ページの「Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)」
- 58 ページの「ネームサービス環境で Solaris 管理ツールを使用する (作業マップ)」

Solaris 管理コンソール問題の障害追跡については、66 ページの「Solaris 管理コンソールの障害追跡」を参照してください。

Solaris 管理コンソール (概要)

この節では、Solaris 管理コンソールについて説明します。

Solaris 管理コンソールについて

Solaris 管理コンソールは、「ツールボックス」というコレクションに格納されている GUI ベースの管理ツール用のコンテナです。このコンソールには、多数の基本管理ツールが含まれているデフォルトのツールボックスがあります。基本管理ツールには、ユーザー、プロジェクト、cron ジョブを管理するためのツール、ファイルシステムをマウント / 共有するためのツール、ディスクやシリアルポートを管理するためのツールなどがあります。それぞれの Solaris 管理ツールについては、表 2-1 を参照してください。

いつでも既存のツールボックスにツールを追加したり、新しいツールボックスを作成することができます。

Solaris 管理コンソールには、主に次の3つの構成要素があります。

- Solaris 管理コンソールクライアント
「コンソール」と呼ばれるビジュアルインタフェースで、管理作業を行うための GUI ツールが含まれています。
- Solaris 管理コンソールサーバー
コンソールと同じマシン上またはリモートマシン上にあり、コンソールによる管理を可能にする「バックエンド」機能を提供します。
- Solaris 管理コンソールツールボックスエディタ
コンソールによく似たアプリケーションで、ツールボックスを追加または変更したり、ツールボックスにツールを追加したり、ツールボックスの適用範囲を広げたりする場合に使用します。たとえば、ネームサービスのドメインを管理するためのツールボックスを追加する場合に使用します。

コンソールを起動すると、デフォルトのツールボックスが表示されます。

Solaris 管理コンソールツール

次の表に、Solaris 管理コンソールのデフォルトのツールボックスに含まれているツールと、各ツールの説明の参照先を示します。

表 2-1 Solaris 管理コンソールツール群

カテゴリ	ツール	説明	参照先
システム状態	システム情報	日付、時間、時間帯などのシステム情報を監視および管理する	『Solaris のシステム管理 (上級編)』の「システム情報の表示と変更 (手順)」
	ログビューア	Solaris 管理コンソールツールのログとシステムログを監視および管理する	『Solaris のシステム管理 (上級編)』の「ソフトウェアの問題解決 (概要)」
	プロセス	システムプロセスを監視および管理する	『Solaris のシステム管理 (上級編)』の「プロセスとシステムのパフォーマンス」

表 2-1 Solaris 管理コンソールツール群 (続き)

カテゴリ	ツール	説明	参照先
システム構成	パフォーマンス	パフォーマンスを監視する	『Solaris のシステム管理 (上級編)』の「システムパフォーマンスの管理 (概要)」
	ユーザー	ユーザー、権利、役割、グループ、メーリングリストを管理する	73 ページの「ユーザーアカウントとグループとは」と『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」
	プロジェクト	/etc/project データベースのエントリを作成および管理する	『Solaris のシステム管理 (資源管理とネットワークサービス)』の「プロジェクトとタスク」
サービス	コンピュータとネットワーク	コンピュータとネットワークの情報を作成および監視する	Solaris 管理コンソールのオンラインヘルプ
	パッチ	パッチを管理する	第 24 章
	スケジュールされたジョブ	スケジュールされる cron ジョブを作成および管理する	『Solaris のシステム管理 (上級編)』の「システムタスクを自動的に実行する方法」
記憶装置	マウントと共有	ファイルシステムをマウントおよび共有する	第 38 章
	ディスク	ディスクパーティションを作成および管理する	第 32 章
	拡張記憶装置	ボリューム、ホットスベアの集合、状態データベースの複製、ディスクセットを作成および管理する	『Solaris ボリュームマネージャの管理』
デバイスとハードウェア	シリアルポート	端末とモデムを設定する	『Solaris のシステム管理 (上級編)』の「端末とモデムの管理 (概要)」

コンテキストヘルプは、ツールの起動後に使用できるようになります。コンテキストヘルプよりも詳細なオンライン情報については、コンソールの「ヘルプ (Help)」メニューから利用できる拡張ヘルプトピックを参照してください。

Solaris 管理コンソールを使用する理由

このコンソールには、管理者にさまざまなメリットをもたらすツールセットが用意されています。コンソールは、次の処理を実行します。

- あらゆる経験レベルをサポートする

ダイアログボックス、ウィザード、コンテキストヘルプなどのグラフィカルインタフェースを使用すれば、経験の浅い管理者でも作業を完了することができます。また、経験を積んだ管理者であれば、何十または何百にも及ぶシステム上に分散している何百もの構成パラメータを管理する場合に、vi よりもこのコンソールを使用した方が便利で安全性が高いことに気がつきます。

- システムへのユーザーアクセスを管理する

デフォルトではどのユーザーもこのコンソールにアクセスできますが、初期設定を変更できるのはスーパーユーザーに限られます。『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」に説明されているように、特定のシステム変更を許可されているユーザー (通常は管理者) に割り当てられる「役割」という特殊なユーザーアカウントを作成できます。

RBAC の主なメリットは、ジョブの実行に必要な作業だけに役割を制限できることです。Solaris 管理ツールを使用する場合に、RBAC は必須ではありません。何も変更せずスーパーユーザーとしてすべてのツールを実行できます。

- コマンド行インタフェースを提供する

管理者は、必要に応じてコマンド行インタフェース (CLI) から Solaris 管理ツールを操作することができます。コマンドの中には、特に GUI ツールのユーザーを管理するコマンドなどの機能をまねて作られたものもあります。表 1-6 に、これらの新しいコマンドの名前と簡単な説明が記載されています。各コマンドのマニュアルページもあります。

専用のコマンドを持たない Solaris 管理ツール (マウントツールや共有ツールなど) では、標準の UNIX コマンドを使用します。

RBAC の機能、メリット、ユーザーサイトでのそれらのメリットの活かし方については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

RBAC を Solaris 管理ツールと組み合わせて使用方法については、52 ページの「Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)」を参照してください。

Solaris 管理コンソールの構成

次の図は、ユーザーツールが開いた状態でのコンソールです。

コンソールの主要部分は、次の 3 つの区画で構成されます。

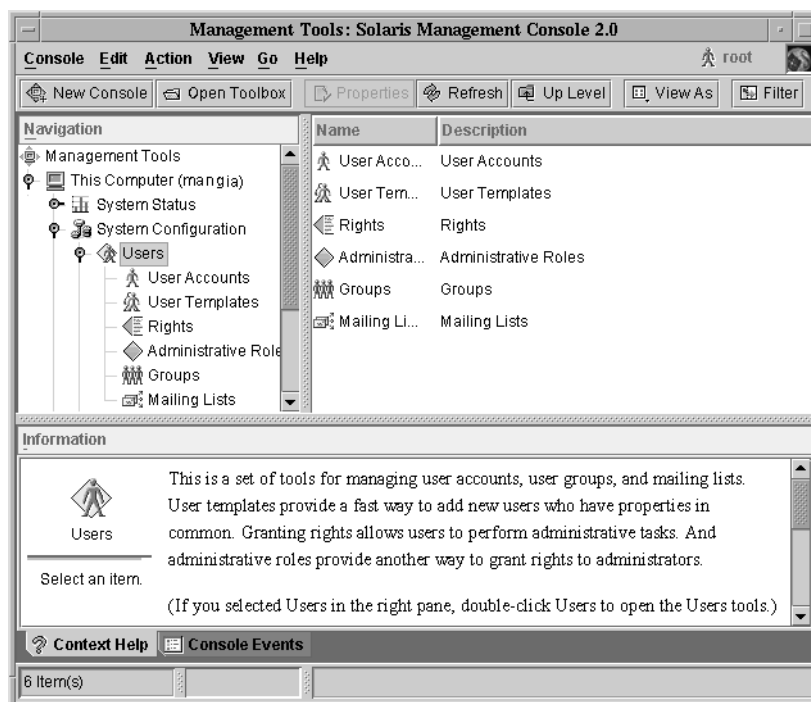


図 2-1 Solaris 管理コンソール–ユーザーツール

- ナビゲーション区画 (左側) – ツール (ツールセット)、フォルダ、他のツールボックスへのアクセス用。ナビゲーション区画内にあるアイコンは、ノードと呼ばれ、それらがフォルダやツールボックスである場合は拡張可能です。
- 表示区画 (右側) – ナビゲーション区画で選択したノードに関する情報の表示用。選択したフォルダの内容、従属ツール、選択したツールに関連付けられたデータを表示します。
- 情報区画 (下側) – コンテキストヘルプまたはコンソールイベントの表示用。

Solaris 管理コンソールウィンドウの変更

コンソールウィンドウのレイアウトは、非常に柔軟に設定することができます。次の機能を使用して、コンソールウィンドウのレイアウトを変更できます。

- 「表示 (View)」メニュー – 「表示 (View)」メニューの「表示 (Show)」オプションを使用すると、オプションのバーと区画を表示または非表示にすることができます。「表示 (View)」メニューの他のオプションは、表示区画内にあるノードの表示を制御します。

- 「コンソール (Console)」メニュー - 「設定の変更 (Preferences)」オプションを使用すると、次の設定ができます。初期ツールボックス、区画の向き、選択肢のクリックまたはダブルクリック、ツールバーのテキストまたはアイコン、フォント、デフォルトのツールロード、認証プロンプト、拡張ログイン。
- 「コンテキストヘルプ (Context Help)」と「コンソールイベント (Console Events)」のトグル - 情報区画の一番下にあるアイコンを使用すると、コンテキストヘルプとコンソールイベントの表示を切り替えることができます。

Solaris 管理コンソールのマニュアル

コンソールやそのツールの使用方法に関するマニュアルは、主にオンラインヘルプシステムから利用できます。オンラインヘルプは、コンテキストヘルプと拡張ヘルプトピックの2つの形式で利用できます。

- コンテキストヘルプは、コンソールツールの使用状況に対応しています。
タブ、入力フィールド、ラジオボタンなどでカーソルをクリックすると、該当するヘルプが情報区画に表示されます。情報区画を閉じたり、開き直したりするには、ダイアログボックスやウィザードの疑問符ボタンをクリックします。
- 拡張ヘルプトピックは、「ヘルプ (Help)」メニューから使用可能であり、一部のコンテキストヘルプから相互参照リンクのクリックにより使用することもできます。
これらのトピックは個別のビューアに表示され、コンテキストヘルプよりも詳細な情報が含まれています。具体的には、各ツールの概要、各ツールの機能説明、特定のツールで使用されるファイル、障害追跡などです。

各ツールの概要については、表 2-1 を参照してください。

役割によるアクセス制御の程度

46 ページの「Solaris 管理コンソールを使用する理由」に説明されているように、Solaris 管理ツールを使用する主な利点は、役割によるアクセス制御 (RBAC) を使用できることです。RBAC を使用すると、管理者はジョブの実行に必要なツールとコマンドだけを使用できます。

組織のセキュリティ要件に応じて、さまざまなレベルの RBAC を使用できます。

RBAC の方法	説明	参照先
RBAC を使用しない	すべての作業をスーパーユーザーとして実行できる。ユーザーとしてログインできる。Solaris 管理ツールを選択するときは、ユーザーとして root に入力し、root のパスワードを入力する	50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」
ルートを役割とする	匿名のルートログインを削除し、ユーザーが root としてログインできないようにする。この方法では、ユーザーが root の役割を持つ前にユーザー自身としてログインする必要がある この方法は他の役割を使用しているかどうかに関係なく適用できるので注意が必要	『Solaris のシステム管理 (セキュリティサービス)』の「root を役割にする」
単一の役割のみ	プライマリ管理者の役割を使用する。これは、root アクセス権だけを持つ場合とほぼ同じ	53 ページの「プライマリ管理者の役割を作成する」
推奨される役割	簡単に設定できる 3 つの役割 (プライマリ管理者、システム管理者、オペレータ) を使用する。これらの役割は、さまざまな責任レベルの管理者がいて、それぞれのジョブ機能が推奨される役割にほぼ合っている組織に適している	『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」
カスタムの役割	組織のセキュリティの必要性に応じて、独自の役割を追加できる	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の計画」

スーパーユーザー (ルート) になるか役割を引き受ける

ほとんどの管理作業 (ユーザー、ファイルシステム、プリンタの追加など) では、まず root (UID=0) としてログインするか、役割を引き受ける (RBAC を使用している場合) 必要があります。root アカウント (「スーパーユーザーアカウント」とも呼ばれる) は、システムを変更したり、緊急時にユーザーファイルの保護を無効にしたりする場合に使用します。

システムの変更がむやみに行われないように、スーパーユーザーアカウントと役割は、管理作業を実行するためだけに使用してください。スーパーユーザーアカウントに関連するセキュリティ問題は、小さな作業を行うときにもユーザーがシステムへの完全なアクセス権を持つてしまうことにあります。

RBAC が実装されていない環境では、スーパーユーザーとしてシステムにログインすることも、su コマンドを使ってスーパーユーザーアカウントに変更することもできます。RBAC が実装されている場合は、コンソールを介して役割を引き受けることも、su を使用して役割を指定することもできます。

コンソールを使用して管理作業を行う場合は、次のいずれかを実行できます。

- ユーザーとしてコンソールにログインし、root ユーザー名とパスワードを入力する
- ユーザーとしてコンソールにログインし、役割を引き受ける

RBAC の大きなメリットは、特定の機能しか使用できないように役割を作成できることです。RBAC を使用している場合は、スーパーユーザーになるのではなく役割を引き受けることで、制限付きのアプリケーションを実行できます。

プライマリ管理者の役割の作成手順については、55 ページの「最初の役割 (プライマリ管理者) を作成する方法」を参照してください。役割を使用できるように RBAC を構成する方法については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

▼ スーパーユーザー (ルート) になるか役割を引き受ける方法

スーパーユーザーになるか役割を引き受ける場合は、次のいずれかの方法を使用します。各方法では、スーパーユーザーのパスワードまたは役割のパスワードを知っている必要があります。

1. スーパーユーザーになる場合は、次のいずれかの方法を選択します。

- a. ユーザーとしてログインし、**Solaris** 管理コンソールを起動し、**Solaris** 管理ツールを選択して **root** としてログインする方法。

この方法では、コンソールから任意の管理作業を実行できます。

Solaris 管理コンソールの起動方法については、64 ページの「ネームサービス環境で **Solaris** 管理コンソールを起動する方法」を参照してください。

- b. システムコンソールでスーパーユーザーとしてログインする方法。

```
hostname console: root
```

```
Password: root-password
```

```
#
```

ポンド記号 (#) は、**Bourne** シェルにおける、スーパーユーザーアカウント用のプロンプトです。

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

- c. ユーザーとしてログインし、コマンド行で **su** コマンドを実行してスーパーユーザーアカウントに変更する方法。

```
% su
```

```
Password: root-password
```

```
#
```

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

- d. スーパーユーザーとしてリモートでログインする方法。この方法は、デフォルトでは使用できません。システムコンソールのスーパーユーザーとしてリモートログインするには、**/etc/default/login** ファイルを変更する必要があります。このファイルの変更方法については、『**Solaris** のシステム管理 (セキュリティサービス)』の「システムのセキュリティ (手順)」を参照してください。

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

2. 役割を引き受ける場合は、次のいずれかの方法を選択します。

- a. ユーザーとしてログインし、コマンド行で **su** コマンドを実行して役割に変更する方法。

```
% su role
```

```
Password: role-password
```

```
$
```

この方法では、設定した役割がアクセスできるすべてのコマンドとツールを使用できます。

- b. ユーザーとしてログインし、**Solaris** 管理コンソールを起動し、**Solaris** 管理ツールを選択して、役割を引き受ける方法。

Solaris 管理コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

この方法では、引き受けた役割がアクセスできる Solaris 管理ツールを使用できます。

Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)

この作業マップでは、スーパーユーザーアカウントを使用するのではなく、役割によるアクセス制御 (RBAC) のセキュリティ機能を使用して管理作業を実行する場合に行う作業について説明します。

注 - この節の内容は、コンソールを RBAC と組み合わせて使用する方法について書かれています。最初にコンソールを使って RBAC を設定する方法について説明するため、RBAC の概要や作業にも触れています。

RBAC の詳細や、RBAC を他のアプリケーションと組み合わせて使用する方法については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

作業	説明	参照先
1. コンソールを起動する	ユーザーアカウントをすでに設定してある場合は、まずユーザーとしてコンソールを起動し、次に root としてコンソールにログインする。ユーザーアカウントを設定していない場合は、まずスーパーユーザーになり、次にコンソールを起動する	56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」
2. 自分のユーザーアカウントを追加する	自分のユーザーアカウントが存在しない場合はそれを追加する	Solaris 管理コンソールのオンラインヘルプ
3. プライマリ管理者の役割を作成する	プライマリ管理者の役割を作成し、自分をこの役割に追加する	55 ページの「最初の役割 (プライマリ管理者) を作成する方法」
4. プライマリ管理者の役割を引き受ける	プライマリ管理者の役割を作成後、その役割を引き受ける	55 ページの「プライマリ管理者の役割を引き受ける方法」

作業	説明	参照先
5. (省略可能) root を役割にする	root を役割にし、他のユーザーが su コマンドを使用して root になれないようにルートの役割に自分を追加する	『Solaris のシステム管理 (セキュリティサービス)』の「root を役割にする」
6. (省略可能) 他の管理役割を作成する	他の管理役割を作成し、各役割に適切な権利を付与する。次に、各役割に該当するユーザーを追加する	『Solaris のシステム管理 (セキュリティサービス)』の「管理役割ツールを使用して役割を作成する方法」

次の節では、Solaris 管理コンソールと RBAC のセキュリティ機能の使い方に関する概要とその手順について説明します。

コンソールに最初にログインした場合

管理者としてコンソールに最初にログインした場合は、まずユーザー (自分自身) としてコンソールを起動し、次にスーパーユーザーとしてログインします。この方法では、コンソールのすべてのツールに完全にアクセスできます。

ここで、RBAC を使用しているかどうかに応じて、一般的な手順を示します。

- RBAC を使用しない – RBAC を使用しない場合は、スーパーユーザーとして作業を続けます。他のすべての管理者も、ジョブを実行するのに root アクセス権が必要になります。
- RBAC を使用する – 次の手順を実行する必要があります。
 - ユーザーアカウントがない場合は、設定する
 - プライマリ管理者という役割を作成する
 - 作成中の役割にプライマリ管理者の権利を割り当てる
 - この役割にユーザーアカウントを割り当てる

プライマリ管理者の役割の作成手順については、55 ページの「最初の役割 (プライマリ管理者) を作成する方法」を参照してください。

役割を使用できるように RBAC を構成する方法については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

プライマリ管理者の役割を作成する

管理役割は、特殊なユーザーアカウントの 1 つです。この役割を引き受けたユーザーは、定義済みの管理作業を実行することができます。

プライマリ管理者の役割は、スーパーユーザーと同様に、すべての管理機能の実行が許可されています。

スーパーユーザー、またはプライマリ管理者の役割を引き受けたユーザーは、他の管理者が実行できる作業を定義することができます。「管理役割を追加 (Add Administrative Role)」ウィザードを使用すると、役割を作成し、その役割に権利を付与し、その役割を引き受けられるユーザーを指定できます。権利とは、特定のアプリケーションを使用するため、またはアプリケーション内にある特定の機能を実行するためのコマンド、つまり承認と他の権利 (その使用は管理者が付与または拒否できる) をまとめて名前を付けたものです。

プライマリ管理者の役割を作成するときは、次の情報の入力を求めるプロンプトが表示されます。

表 2-2 コンソールを使用して役割を追加するための項目の説明

項目	説明
役割名	管理者が特定の役割にログインするために使用する名前を選択する
役割の正式名称	(省略可能) この役割の名前をフルネームでわかりやすく入力する
説明	この役割の詳細な説明
役割 ID 番号	この役割に割り当てられている ID 番号を選択する。この番号は、UID の ID セットと同じ
役割シェル	ユーザーが端末またはコンソールのウィンドウにログインするか、そのウィンドウで役割を引き受けるときに実行するシェルを選択する
役割のメーリングリストを作成	項目をチェックすると、役割と同じ名前でメーリングリストを作成する。メーリングリストを使用すると、その役割に割り当てられているすべてのユーザーに電子メールを送信できる
役割パスワードとパスワードを確認	役割のパスワードを設定および再入力する
有効な権利と許可された権利	「有効な権利 (Available Rights)」のリストから権利を選択し、「許可された権利 (Granted Rights)」のリストに追加することにより、この役割に権利を割り当てる
ホームディレクトリの選択	この役割の専有ファイルが格納されるホームディレクトリサーバーを選択する
ユーザーの役割への割り当て	特定のユーザーが特定の作業を行うための役割を持てるようにユーザーを役割に追加する

役割によるアクセス制御の詳細と、役割を使用して安全な環境を作成する方法については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

▼ 最初の役割 (プライマリ管理者) を作成する方法

この手順では、プライマリ管理者の役割を作成し、それをユーザーアカウントに割り当てる方法について説明します。ユーザーアカウントはすでに作成してあるものとします。

1. ユーザーとしてコンソールを起動します。

```
% /usr/sadm/bin/smc &
```

コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

ユーザーアカウントを作成する必要がある場合は、コンソールのオンラインヘルプを参照してください。

2. ナビゲーション区画で「このコンピュータ (Computer)」アイコンをクリックします。
3. 「System Configuration」→「ユーザー (Users)」→「管理役割 (Administrative Roles)」の順にクリックします。
4. 「アクション (Action)」→「管理役割を追加 (Add Administrative Role)」の順にクリックします。
「管理役割を追加 (Add Administrative Role)」ウィザードが開きます。
5. 次の手順に従って、「管理役割を追加 (Add Administrative Role)」ウィザードでプライマリ管理者の役割を作成します。
 - a. 役割名、役割の正式名、備考欄、役割 ID 番号、役割シェル、役割のメーリングリストを作成するかどうかを指定する。「次へ (Next)」をクリックする。
 - b. 役割のパスワードを設定し、確認する。「次へ (Next)」をクリックする。
 - c. 「有効な権利 (Available Rights)」欄からプライマリ管理者の権利を選択後、「追加 (Add)」をクリックし、「許可された権利 (Granted Rights)」欄に追加する。「次へ (Next)」をクリックする。
 - d. 役割のホームディレクトリを選択する。「次へ (Next)」をクリックする。
 - e. この役割を引き受けることができるユーザーのリストに自分を割り当てる。「次へ (Next)」をクリックする。
必要に応じて、表 2-2 で役割の各項目の説明を参照してください。
6. 「完了 (Finish)」をクリックする。

▼ プライマリ管理者の役割を引き受ける方法

プライマリ管理者の役割を作成し終わったら、まずユーザー (自分自身) としてコンソールにログインし、次にプライマリ管理者の役割を引き受けます。

役割を引き受けるときは、その役割の権利を含むすべての属性を引き受けます。同時に、自分自身のユーザープロパティはすべて放棄します。

1. コンソールを起動します。

```
% /usr/sadm/bin/smc &
```

コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

2. ユーザー名とパスワードを使ってログインします。
引き受けることができる役割のリストが表示されます。
3. プライマリ管理者の役割にログインし、役割のパスワードを入力します。

Solaris 管理コンソールを起動する

次の手順では、コンソールを起動し、Solaris 管理ツールにアクセスする方法について説明します。

▼ スーパーユーザーまたは役割としてコンソールを起動する方法

自分自身のユーザーアカウントを使ってユーザーとしてコンソールを起動する場合は、Solaris 管理ツールへのアクセスが制限されます。十分なアクセス権を得るために、まずユーザー (自分自身) としてログインし、次に引き受けることができる役割のいずれかでログインできます。プライマリ管理者の役割を引き受けることができる場合は、スーパーユーザーの場合と同様に、すべての Solaris 管理ツールにアクセスできます。

1. CDE 環境などのウィンドウ環境で操作していることを確認します。
2. 次のいずれかの方法でコンソールを起動します。

- コマンド行から、次のように入力する

```
% /usr/sadm/bin/smc &
```

コンソールが初めて起動するときは 1 ~ 2 分かかります。

- CDE フロントパネルの「ツール (Tool)」メニューから起動する
 - CDE のアプリケーションマネージャまたはファイルマネージャの「Solaris 管理コンソール」アイコンをダブルクリックして起動する
- 「Solaris 管理コンソール」ウィンドウが表示されます。

注 - Solaris 管理コンソールのスタートアップメッセージを表示する場合は、自分のウィンドウ環境でコンソールを開いてください。Solaris 管理コンソールを起動する前に、Solaris 管理コンソールサーバーを手動で起動しないでください。このサーバーは、Solaris 管理コンソールを起動すると、自動的に起動します。コンソールに関する問題の障害追跡については、66 ページの「Solaris 管理コンソールの障害追跡」を参照してください。

3. ナビゲーション区画の「管理ツール (Management Tools)」アイコンの下にある「このコンピュータ (This Computer)」アイコンをダブルクリックします。カテゴリのリストが表示されます。
4. (省略可能) 適切なツールボックスを選択します。

デフォルト以外のツールボックスを使用する場合は、ナビゲーション区画から該当するツールボックスを選択します。あるいは、コンソールメニューから「ツールボックスを開く (Open Toolbox)」を選択し、任意のツールボックスをロードします。

各種ツールボックスの使用方法については、61 ページの「特定環境用のツールボックスを作成する方法」を参照してください。
5. カテゴリのアイコンをダブルクリックして、特定のツールにアクセスします。特定の作業の実行方法を確認する場合は、オンラインヘルプを使用します。
6. ツールのアイコンをダブルクリックします。

「ログイン (Log-In)」ポップアップウィンドウが表示されます。
7. スーパーユーザーまたは役割のどちらでこのツールを使用するかを決めます。
 - スーパーユーザーとしてログインし、スーパーユーザーとして作業する場合は、手順 8 に進む
 - ユーザー (自分自身) としてログインし、プライマリ管理者の役割を引き受ける場合は、手順 9 と 10 に進む
8. スーパーユーザーとしてログインしている場合は、**root** のパスワードを入力します。
9. ユーザー (自分自身) としてログインしている場合は、バックスペースキーを使って **root** のユーザー名を削除します。次に、適切なユーザー **ID** とユーザーパスワードを入力します。

ユーザーが引き受けることができる役割のリストが表示されます。
10. プライマリ管理者の役割、またはそれと同等の役割を選択し、役割のパスワードを入力します。

プライマリ管理者の役割の作成手順については、55 ページの「最初の役割 (プライマリ管理者) を作成する方法」を参照してください。

ツールのメインメニューが表示されます。

ネームサービス環境で Solaris 管理ツールを使用する (作業マップ)

デフォルトでは、Solaris 管理ツールはローカル環境で動作するように設定されます。たとえば、マウントと共有ツールを使用すると、特定のシステム上でディレクトリをマウントおよび共有できますが、NIS や NIS+ 環境ではできません。ただし、ネームサービス環境では、ユーザーツールやコンピュータとネットワークツールを使って情報を管理できます。

ネームサービス環境でコンソールのツールを使用する場合は、まずネームサービスのツールボックスを作成し、次にそのツールボックスにツールを追加する必要があります。

作業	説明	参照先
1. 前提条件を確認する	ネームサービス環境でコンソールを使用する前に、前提条件が満たされていることを確認する	60 ページの「ネームサービス環境で Solaris 管理コンソールを使用するための前提条件」
2. ネームサービス用のツールボックスを作成する	新規ツールボックスウィザードを使用して、ネームサービスツール用のツールボックスを作成する	61 ページの「特定環境用のツールボックスを作成する方法」
3. ネームサービスツールボックスにツールを追加する	作成したネームサービスツールボックスにユーザーツール、または他のネームサービスツールを追加する	63 ページの「ツールボックスにツールを追加する方法」
4. 直前に作成したツールボックスを選択する	直前に作成したツールボックスを選択して、ネームサービス情報を管理する	64 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」

RBAC セキュリティファイル

Solaris 管理コンソールで使用する RBAC セキュリティファイルは、Solaris 9 リリースへのアップグレードまたはそのインストール時に作成されます。Solaris 管理コンソールのパッケージをインストールしない場合、RBAC を使用するのに必要なデータがない状態で RBAC セキュリティファイルがインストールされます。Solaris 管理コンソールのパッケージについては、66 ページの「Solaris 管理コンソールの障害追跡」を参照してください。

Solaris 9 リリースの RBAC セキュリティファイルは、ネームサービス環境で Solaris 管理コンソールのツールを使用できるように、作成したネームサービスに組み込まれます。

ローカルサーバー上のセキュリティファイルは、標準アップグレードの一環として、ypmake コマンド、nispopulate コマンド、または同様の LDAP コマンドによってネームサービス環境に作成されます。サポートされているネームサービスは次のとおりです。

- NIS
- NIS+
- LDAP
- files

注 - NIS+ 環境では projects データベースはサポートされていません。

RBAC セキュリティファイルは、Solaris 9 リリースへのアップグレードまたはそのインストール時に作成されます。

次の表に、Solaris 9 でインストールされる定義済みのセキュリティファイルとその簡単な説明を示します。

表 2-3 RBAC セキュリティファイル

ローカルファイル名	テーブルまたはマップ名	説明
/etc/user_attr	user_attr	ユーザーと役割を承認と権利プロファイルに関連付ける
/etc/security/auth_attr	auth_attr	承認とその属性を定義し、関連付けられたヘルプファイルを識別する
/etc/security/prof_attr	prof_attr	権利プロファイルを定義し、権利プロファイルによって割り当てられた承認のリストを表示し、関連付けられたヘルプファイルを識別する
/etc/security/exec_attr	exec_attr	権利プロファイルに割り当てられている特権付きの操作を定義する

アップグレードに失敗した場合は、次のときに smattrpop コマンドで RBAC セキュリティファイルを作成してください。

- 権利プロファイルを作成または変更するとき
- usr_attr ファイルをカスタマイズして、ユーザーと役割を追加する必要があるとき

詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

ネームサービス環境で Solaris 管理コンソールを使用するための前提条件

次の表に、ネームサービス環境で Solaris 管理コンソールを使用する前に行う必要のある操作を示します。

前提条件	参照先
Solaris 9 をインストールする	『Solaris 9 12/03 インストールガイド』
ネームサービス環境を設定する	『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』
管理範囲を選択する	60 ページの「管理範囲」
ネームサービスデータにアクセスできるように /etc/nsswitch.conf ファイルが構成されていることを確認する	60 ページの「/etc/nsswitch.conf ファイル」

管理範囲

Solaris 管理コンソールでは、選択した管理ツールで使用するネームサービス環境を指すときに「管理範囲 (management scope)」という用語を使用します。ユーザーおよびコンピュータとネットワークツールの管理範囲は、LDAP、NIS、NIS+、files のの中から選択します。

コンソールセッション時に選択する管理範囲は、/etc/nsswitch.conf ファイルで識別される一次ネームサービスと一致している必要があります。

/etc/nsswitch.conf ファイル

各システムの /etc/nsswitch.conf ファイルは、そのシステムのネームサービスルックアップ (ここからデータが読み取られる) のポリシーを示します。

注 - コンソールからアクセスされるネームサービス (コンソールのツールボックスエディタで指定する) が /etc/nsswitch.conf ファイルの検索パスに含まれていることを確認する必要があります。指定のネームサービスがその検索パスにない場合、ツールが予測のつかない動作をしてエラーまたは警告が発生する可能性があります。

ネームサービス環境で Solaris 管理ツールを使用するときは、1 回の操作で多数のユーザーに影響を及ぼす可能性があります。たとえば、NIS ネームサービスのユーザーを削除すると、そのユーザーは NIS を使用しているすべてのシステムで削除されます。

ネットワーク内のさまざまなシステムで `/etc/nsswitch.conf` 構成が異なっていると、予期しない結果が生じる可能性があります。これを回避するために、Solaris 管理ツールで管理するすべてのシステムには、一貫したネームサービス構成を設定する必要があります。

▼ 特定環境用のツールボックスを作成する方法

Solaris オペレーティング環境を管理するためのアプリケーションは「ツール」と呼ばれ、それらのツールは「ツールボックス」というコレクションに格納されます。ツールボックスは、ローカルサーバー (コンソールがある場所) にもリモートマシンにも配置できます。

ツールボックスエディタを使用すると、新しいツールボックスの追加、既存のツールボックスへのツールの追加、またはツールボックスの適用範囲の変更を行うことができます。たとえば、ドメインをローカルファイルからネームサービスに変更できます。

注 - ツールボックスエディタは、通常のユーザーとして起動できます。ただし、変更を加えたり、変更内容をデフォルトのコンソールツールボックス (`/var/sadm/smc/toolboxes`) に保存する場合は、`root` としてツールボックスエディタを起動する必要があります。

1. ツールボックスエディタを起動します。

```
# /usr/sadm/bin/smc edit &
```

2. 「ツールボックス (Toolbox)」メニューから「開く (Open)」を選択します。
3. 「ツールボックスを開く (Toolboxes)」ウィンドウで「このコンピュータ (This Computer)」アイコンを選択します。
4. 「開く (Open)」を選択します。
「このコンピュータ (This Computer)」ツールボックスがウィンドウに開きます。
5. ナビゲーション区画で「このコンピュータ (This Computer)」アイコンを再度選択します。
6. 「アクション (Action)」メニューから「フォルダを追加 (Add Folder)」を選択します。
7. 「フォルダ (Folder)」ウィザードを使用して、ネームサービス環境用の新しいツールボックスを追加します。

- a. 名前 (**Name**) と説明 (**Description**) – 「名前 (**Full Name**)」ウィンドウに名前を入力する。「次へ (**Next**)」をクリックする。
たとえば、NIS 環境の場合は「NIS tools」と入力する。
 - b. 「説明 (**Description**)」ウィンドウに説明を入力する。「次へ (**Next**)」をクリックする。
たとえば、「tools for NIS environment」と入力する。
 - c. アイコン (**Icons**) – デフォルト値を使用する。「次へ (**Next**)」をクリックする。
 - d. 管理範囲 (**Management Scope**) – 「上書き (**Override**)」を選択する。
 - e. 「管理範囲 (**Management Scope**)」プルダウンメニューから該当するネームサービスを選択する。
 - f. 必要に応じて、ネームサービスのマスター名を「サーバー (**Server**)」フィールドに追加する。
 - g. サーバーが管理しているドメインを「ドメイン(**Domain**)」フィールドに追加する。
 - h. 「完了 (**Finish**)」をクリックする。
左側のナビゲーション区画に新しいツールボックスが表示されます。
8. 新しいツールボックスのアイコンを選択します。
 9. 「ツールボックス (**Toolbox**)」メニューから「別名保存 (**Save As**)」を選択します。
 10. 「ローカルツールボックス (**Local Toolbox**)」ダイアログボックスの「フォルダ (**Folder**)」からフォルダを選択し、「ファイル名 (**Filename**)」にツールボックスのファイル名を入力します。接尾辞 **.tbx** を使用します。

`/var/sadm/smc/toolboxes/this_computer/toolbox-name.tbx`
 11. 「保存 (**Save**)」をクリックします。
コンソールウィンドウのナビゲーション区画に新しいツールボックスが表示されます。

次に進む手順

ネームサービスツールボックスを作成し終わったら、その中にネームサービスを入れることができます。詳細については、63 ページの「ツールボックスにツールを追加する方法」を参照してください。

▼ ツールボックスにツールを追加する方法

コンソールに添付されているデフォルトのツールのほかに、コンソールから起動できるツールがいくつか開発されています。これらのツールが使用可能になったら、既存のツールボックスに追加できます。

また、ローカル管理またはネットワーク管理用に新しくツールボックスを作成し、その新しいツールボックスにツールを追加することもできます。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、ツールボックスエディタを起動します。

```
# /usr/sadm/bin/smc edit &
```

3. ツールボックスを選択します。
ネームサービス環境で作業する場合は、直前に作成したツールボックスをツールボックスエディタで選択します。
詳細については、61 ページの「特定環境用のツールボックスを作成する方法」を参照してください。
4. 「アクション (Action)」メニューから「ツールを追加 (Add Tool)」を選択します。
5. 「ツールを追加 (Add Tool)」ウィザードを使用して新しいツールを追加します。
 - a. 「サーバーの選択 (Server Selection)」 - ネームサービスのマスターを「サーバー (Server)」ウィンドウに追加する。「次へ (Next)」をクリックする。
 - b. 「ツールの選択 (Tools Selection)」 - 追加するツールを「ツール (Tools)」ウィンドウから選択する。「次へ (Next)」をクリックする。
このツールボックスがネームサービス用のツールボックスである場合は、ネームサービス環境で作業するツールを選択します。たとえば、ユーザーツールを選択します。
 - c. 「ツールクラス名 (Name) と説明 (Description)」 - デフォルト値を使用する。「次へ (Next)」をクリックする。
 - d. 「アイコン (Icons)」 - カスタムアイコンを作成していない場合は、デフォルト値を使用する。「次へ (Next)」をクリックする。
 - e. 「管理範囲 (Management Scope)」 - デフォルト値「親から継承 (Inherit from Parent)」を使用する。「次へ (Next)」をクリックする。
 - f. 「ツールの読み込み (Tool Loading)」 - デフォルト値「選択された時にツールを読み込む (Load tool when selected)」を使用する。「完了 (Finish)」をクリックする。
6. 「ツールボックス (Toolbox)」メニューから「保存 (Save)」を選択して、更新したツールボックスを保存します。

「ローカルツールボックス (Local Toolbox)」ウィンドウが表示されます。

▼ ネームサービス環境で Solaris 管理コンソールを起動する方法

ネームサービスのツールボックスを作成し、そのツールボックスにツールを追加し終わったら、Solaris 管理コンソールを起動し、作成したツールボックスを開いてネームサービス環境を管理できます。

1. 次の前提条件を満たしていることを確認します。
 - a. ログインしているシステムがネームサービス環境で動作するように設定されているかを確認する。
 - b. `/etc/nsswitch.conf` ファイルがユーザーのネームサービス環境に適合するように設定されているかを確認する。
2. Solaris 管理コンソールを起動します。

詳細については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。
3. ネームサービス用に作成したツールボックスを選択します。選択したツールボックスがナビゲーション区画に表示されます。

ネームサービス用のツールボックスの作成方法については、61 ページの「特定環境用のツールボックスを作成する方法」を参照してください。

Solaris 管理コンソールにツールを追加する

この節では、レガシーツールまたは別製品のツールをコンソールに追加する方法について説明します。これらのツールに認証を追加する場合は、『Solaris のシステム管理 (セキュリティサービス)』の「レガシーアプリケーションのセキュリティ保護」を参照してください。

▼ レガシーツールをツールボックスに追加する方法

レガシーツールとは、特に Solaris 管理ツールとして設計されていないアプリケーションのことです。コンソールのツールボックスには、X アプリケーション、コマンド行インタフェース、HTML の 3 種類のレガシーツールアプリケーションを追加できます。ツールボックスに追加した各ツールは、Solaris 管理コンソールから起動できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、**Solaris** 管理コンソールのツールボックスエディタを起動します。

```
# /usr/sadm/bin/smc edit &
```
3. レガシーアプリケーションを追加したいツールボックスを開きます。
選択したツールボックスはツールボックスエディタで開きます。
4. レガシーアプリケーションを追加するツールボックス内のノードを選択します。
レガシーアプリケーションは、ツールボックスの最上位のノードにも別のフォルダにも追加できます。
5. 「アクション (Action)」 → 「従来のアプリケーションを追加 (Add Legacy Application)」の順にクリックします。
「従来のアプリケーションのウィザード (Legacy Application Wizard: General)」の最初のパネルが表示されます。
6. ウィザードの指示に従います。
7. エディタでツールボックスを保存します。

▼ 別製品のツールをインストールする方法

コンソールから起動できるツールパッケージを新たに追加する場合は、この手順に従います。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 新しいツールパッケージをインストールします。

```
# pkgadd ABCDtool
```
3. コンソールを再起動して、コンソールに新しいツールを認識させます。
 - a. コンソールサーバーを停止する。

```
# /etc/init.d/init.wbem stop
```
 - b. コンソールサーバーを起動する。

```
# /etc/init.d/init.wbem start
```
4. コンソールを起動して、新しいツールが表示されることを確認します。
詳細については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

Solaris 管理コンソールの障害追跡

この障害追跡の手順を使用する前に、次のパッケージがインストールされていることを `pkginfo(1)` コマンドで確認してください。

SUNWmc	Solaris Management Console 2.1 (サーバー構成要素)
SUNWmcc	Solaris Management Console 2.1 (クライアント構成要素)
SUNWmccom	Solaris Management Console 2.1 (共通の構成要素)
SUNWmcdev	Solaris Management Console 2.1 (開発キット)
SUNWmcex	Solaris Management Console 2.1 (例)
SUNWwbmc	Solaris Management Console 2.1 (WBEM 構成要素)

これらのパッケージには、基本的な Solaris 管理コンソール起動ツールが用意されています。Solaris 管理コンソールとそのすべてのツールを使用するには、`SUNWprog` クラスタをインストールする必要があります。

▼ Solaris 管理コンソールの障害追跡の方法

Solaris 管理コンソールを起動すると、クライアントとサーバーが自動的に起動します。

コンソールが表示可能で、ツールの実行に問題がある場合は、サーバーが実行していない可能性があります。あるいは、サーバーがエラー状態にあり、いったん停止してから再起動することによって解決できる可能性もあります。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 以下のコマンドでコンソールサーバーが稼動しているかどうかを調べます。

```
# /etc/init.d/init.wbem status
```

コンソールサーバーが稼動している場合は、次のようなメッセージが表示されず。

```
SMC server version 2.1.0 running on port 898.
```

3. コンソールサーバーが稼動していない場合は、以下のコマンドで起動します。

```
# /etc/init.d/init.wbem start
```

すぐに、次のようなメッセージが表示されます。

```
SMC server is ready.
```

4. サーバーが稼動しているのに問題が引き続き発生している場合は、いったんコンソールサーバーを停止してから、再起動します。
 - a. コンソールサーバーを停止する。

```
# /etc/init.d/init.wbem stop
```

次のようなメッセージが表示されます。

```
Shutting down SMC server on port 898.
```

- b.** コンソールサーバーを起動する。

```
# /etc/init.d/init.wbem start
```


第3章

ユーザーアカウントとグループの管理

以下の各章では、ユーザーおよびグループの管理方法について説明します。

第4章	ユーザーアカウントおよびグループ管理の概要について説明します。
第5章	ユーザーアカウントおよびグループの管理手順について説明します。

第 4 章

ユーザーアカウントとグループの管理 (概要)

この章では、ユーザーアカウントとグループを管理するためのガイドラインと管理計画の概要について説明します。また、ユーザーの作業環境のカスタマイズについても説明します。

この章の内容は次のとおりです。

- 71 ページの「ユーザーとグループの管理における新機能」
- 73 ページの「ユーザーアカウントとグループとは」
- 74 ページの「ユーザーアカウント管理のガイドライン」
- 80 ページの「グループを管理するガイドライン」
- 81 ページの「ユーザーアカウントとグループを管理するツール」
- 88 ページの「ユーザーアカウントとグループ情報の格納場所」
- 93 ページの「ユーザーの作業環境のカスタマイズ」

ユーザーアカウントおよびグループの管理手順については、第 5 章を参照してください。

ユーザーとグループの管理における新機能

この節では、ユーザーとグループを管理するための Solaris 9 リリースの新機能について説明します。

Solaris 管理コンソールツール群

Solaris 管理コンソールでは、Solaris 管理ツール群を利用でき、このツール群を使用してすべてのユーザーおよびグループ機能を管理できます。Solaris 管理コンソールの詳細については、第 2 章を参照してください。特定のユーザーおよびグループ管理作業の実行方法については、82 ページの「Solaris ユーザー管理ツールで実行できる作業」を参照してください。

Solaris Directory Services

Sun ONE Directory Server および他の LDAP ディレクトリサーバーを使用すると、LDAP (Lightweight Directory Access Protocol) ディレクトリサービスのユーザーおよびグループ情報を管理できます。ユーザーおよびグループ情報は、NIS、NIS+、またはファイル環境でも管理できます。

LDAP の設定の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

Sun ONE Directory Server を使ってユーザーおよびグループを管理する方法については、<http://docs.sun.com> の『iPlanet Directory Server 5.1 管理者ガイド』を参照してください。

プロジェクトでユーザーおよびリソースを管理する

Solaris 9 リリースでは、ユーザーおよびグループを「プロジェクト (project)」(システム使用率またはリソースアロケーションチャージバックの基礎として使用される、作業負荷の構成要素を示す識別子) のメンバーにすることができます。プロジェクトは、Solaris リソース管理機能の一部で、システムリソースの管理に使用されます。

Solaris 9 リリースを実行するシステムにログインするには、ユーザーはプロジェクトのメンバーになる必要があります。デフォルトでは、ユーザーは Solaris 9 リリースのインストール時に `group.staff` プロジェクトのメンバーになり、他のプロジェクト情報は設定されていません。

ユーザーのプロジェクト情報は、`/etc/project` ファイルに格納され、このファイルは、ローカルシステム (ファイル)、NIS ネームサービス、または LDAP ディレクトリサービスに保存できます。Solaris 管理コンソールを使用すると、プロジェクト情報を管理できます。

`/etc/project` ファイルは、ユーザーがログインするために必要ですが、プロジェクトを使用しない場合は管理する必要はありません。

プロジェクトの使用法および設定方法の詳細については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「プロジェクトとタスク」を参照してください。

ユーザーアカウントとグループとは

基本的なシステム管理作業の1つに、サイトの各ユーザーにユーザーアカウントを設定することがあります。通常ユーザーアカウントには、ユーザーがスーパーユーザーのパスワードを知らなくても、システムにログインして、システムを使用するのに必要な情報が含まれています。ユーザーアカウント情報は、次の要素で構成されています。

構成要素	説明
ユーザー名	ユーザーがシステムにログインするのに使用する名前。ログイン名とも呼ばれます。
パスワード	ユーザーがシステムにアクセスするために、ユーザー名とともに入力しなければならない文字の組み合わせ。
ユーザーのホームディレクトリ	通常、ログイン時にユーザーのディレクトリになるディレクトリ。通常、ユーザーのホームディレクトリにはそのユーザーの大部分のファイルが含まれます。
ユーザー初期設定ファイル	ユーザーがシステムにログインするときに、そのユーザーの動作環境の設定を制御するシェルスクリプト。

また、ユーザーアカウントを設定するとき、ユーザーをあらかじめ定義されたユーザーグループに追加できます。グループは一般に、ファイルまたはディレクトリへのグループアクセス権を設定して、グループ内のユーザーだけがファイルとディレクトリにアクセスできるようにするために使用されます。

たとえば、ごく少数のユーザーだけにアクセスさせたい機密ファイルを入れるディレクトリを作成できます。topsecret プロジェクトに携わるユーザーを含む topsecret という名前のグループを設定します。そして、topsecret ファイルの読み取り権を topsecret グループに対して設定します。こうすれば、topsecret グループ内のユーザーだけが、ファイルを読み取ることができます。

また、「役割」という特別な種類のユーザーアカウントは、指定したユーザーに特別な特権を与えるときに使用します。詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

ユーザーまたはグループは、1つまたは複数のプロジェクトのメンバーになることができます。プロジェクトは、システムリソースのチャージバックに使用される識別子です。プロジェクトの使用方法については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「プロジェクトとタスク」を参照してください。

ユーザーアカウント管理のガイドライン

次の節では、ユーザーアカウントを作成するガイドラインと計画方法について説明します。

ネームサービス

大規模なサイトでユーザーアカウントを管理する場合、LDAP、NIS、NIS+などのネームサービスまたはディレクトリサービスを使用できます。ネームサービスまたはディレクトリサービスを使うと、ユーザーアカウント情報を各システムの /etc 内のファイルに格納するのではなく、一元管理できます。ユーザーアカウントにネームサービスまたはディレクトリサービスを使用すれば、サイト全体のユーザーアカウント情報をシステムごとにコピーしなくても、同じユーザーアカウントのままシステム間を移動できます。ネームサービスまたはディレクトリサービスを使用すると、ユーザーアカウント情報を集中化および一元化して管理できます。

ユーザー (ログイン) 名

ユーザーは、ユーザー名 (ログイン名とも呼ばれる) を使って、自分のシステムと、適切なアクセス権を持つリモートシステムにアクセスできます。作成するユーザーアカウントそれぞれに、ユーザー名を選択しなければなりません。

ユーザー名または役割名を作成するときは、次のガイドラインに従ってください。

- 複数のドメインにまたがることもあるユーザーの組織内で、固有であること。
- 2～8文字の英数字を使用すること。最初の文字は英字でなければならない、少なくとも1文字は小文字にする必要があります。

ユーザー名には、ピリオド (.)、アンダースコア (_)、ハイフン (-) を使用できますが、これらの文字により障害が発生するソフトウェアもあるため、使用はお勧めできません。

ユーザー名を探しやすいように、ユーザー名の標準的な割り当て方法を使用することを検討してください。また、ユーザー名はユーザーが覚えやすいものにしてください。単純な規則の例としては、ユーザーのファーストネームの頭文字とラストネームの最初の7文字を使用します。たとえば、Ziggy Ignatz は zignatz になります。他のユーザー名と重複する場合は、ユーザーのファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の6文字を使用します。たとえば、Ziggy Top Ignatz は ztignatz になります。さらに重複する場合、次の方法を検討してください。

- ファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の5文字を使用する

- 固有の名前になるまで 1、2、3 などの数字を使用する

注 - それぞれの新しいユーザー名は、システムまたは NIS や NIS+ のドメインに登録されているメール別名(エイリアス)とは異なるものでなければなりません。そうしないと、メールは実際のユーザーではなく別名に送られることがあります。

ユーザー ID 番号

ユーザー名に関連するものとして、ユーザー識別 (UID) 番号があります。ユーザーがログインしようとするシステムは、UID 番号によってユーザー名を識別したり、ファイルとディレクトリの所有者を識別したりします。多数の異なるシステム上で、ある個人用に複数のユーザーアカウントを作成する場合は、常に同じユーザー名とユーザー ID を使用してください。そうすれば、そのユーザーは、所有権の問題を起こすことなく、システム間で簡単にファイルを移動できます。

UID 番号は、2147483647 以下の整数でなければなりません。UID 番号は、通常のユーザーアカウントと特殊なシステムアカウントの両方に必要です。次の表にユーザーアカウントとシステムアカウントに予約されている UID 番号を示します。

表 4-1 予約済みの UID 番号

ユーザー ID 番号	ユーザー/ログインアカウント	説明
0 - 99	root、daemon、bin、sys など	システムアカウント
100 - 2147483647	通常のユーザー	汎用アカウント
60001 と 65534	nobody および nobody4	匿名ユーザー
60002	noaccess	アクセス権のないユーザー

0 から 99 までの UID 番号は予約されていますが、これらの番号でユーザーを追加することはできません。ただし、0 から 99 までの番号を通常のユーザーアカウントには使用しないでください。システム上の定義により、root には常に UID 0、daemon には UID 1、擬似ユーザー bin には UID 2 が設定されます。また、UID が passwd ファイルの先頭にくるように、uucp ログインや、who、tty、ttytype などの擬似的なユーザーログインには低い UID を与えるようにしてください。

ユーザー (ログイン) 名と同様に、固有の UID を割り当てる方法を決めてください。企業によっては、従業員に固有の番号を割り当て、管理者がその従業員番号にある番号を加えて固有の UID 番号を作成している場合もあります。

セキュリティ上のリスクを最小限に抑えるために、削除したアカウントの UID を再利用することは避けてください。どうしても UID を再利用する必要がある場合、はじめから作りなおして、新しいユーザーが前のユーザーの属性に影響されないようにしてください。たとえば、前のユーザーがプリンタの拒否リストに含まれていたためプリンタにアクセスできなかった場合、その属性を新しいユーザーに適用することが正しいとは限りません。

大きな数値のユーザー ID とグループ ID の使用

UID と GID の値の制限が符号付き整数の最大値 (つまり、2147483647) に引き上げられました。

ただし、60000 より大きな数値の UID と GID は機能的に完全でなく、多くの Solaris の機能と互換性がありません。したがって、60000 を超える UID と GID を使用することは避けてください。

次の表では、Solaris の以前のリリースとの相互運用性に関する問題について説明します。

表 4-2 60000 より大きな数値の UID または GID の相互運用性に関する問題

項目	製品またはコマンド	問題または注意
NFS™ 互換性	SunOS™ 4.0 NFS ソフトウェアおよびその互換リリース	NFS サーバーとクライアントのコードは、大きな UID と GID を 16 ビットに切り捨てる。この状況では、SunOS 4.0 およびその互換リリースが動作しているシステムが、大きな UID と GID を使用している環境で使用されると、セキュリティの問題が発生する可能性がある。この問題を回避するには、SunOS 4.0 およびその互換リリースが動作しているシステムにパッチをあてる必要がある
ネームサービスの互換性	NIS ネームサービスおよびファイルベースのネームサービス	60000 より大きな数値の UID を持つユーザーは、Solaris 2.5 およびその互換リリースが動作しているシステムでは、ログインしたり、su コマンドを使用したりできるが、そのユーザーの UID と GID は 60001 (nobody) に設定される
	NIS+ ネームサービス	60000 より大きな数値の UID を持つユーザーは、Solaris 2.5 およびその互換リリースと NIS+ ネームサービスが動作しているシステムではアクセスが拒否される

表 4-3 大きな UID または GID の制限の要約

UID または GID の値	制限
60003 以上	<ul style="list-style-type: none"> ■ この分類のユーザーが、Solaris 2.5 およびそれ以前の互換リリースと NIS ネームサービスまたはファイルネームサービスが動作しているシステムにログインすると、nobody の UID および GID を取得する
65535 以上	<ul style="list-style-type: none"> ■ Solaris 2.5 およびその互換リリースが NFS バージョン 2 ソフトウェアと一緒に動作しているシステムでは、この分類の UID は 16 ビットに切り捨てられるため、セキュリティの問題が発生する可能性がある ■ この分類のユーザーがデフォルトのアーカイブフォーマットで cpio コマンドを使用してファイルをコピーすると、ファイルごとにエラーメッセージが表示される。そして、UID と GID はアーカイブにおいて nobody に設定される。 ■ SPARC システム: この分類のユーザーが SunOS 4.0 およびその互換バージョンで動作可能なアプリケーションを実行すると、一部のシステムコールから EOVERFLOW が戻されて、そのユーザーの UID と GID は nobody にマップされる ■ x86 システム: この分類のユーザーが SVR3 互換のアプリケーションを実行すると、一部のシステムコールから EOVERFLOW が返される場合がある ■ x86 システム: この分類のユーザーが、マウントされた System V ファイルシステムでファイルまたはディレクトリを作成しようとした場合、System V ファイルシステムは EOVERFLOW エラーを返す
100000 以上	<ul style="list-style-type: none"> ■ ps -l コマンドは最大 5 桁の UID を表示する。したがって、99999 より大きな UID または GID を含むときは、出力される列が揃わない
262144 以上	<ul style="list-style-type: none"> ■ この分類のユーザーが -H odc フォーマットで cpio コマンドまたは pax -x cpio コマンドを使用してファイルをコピーすると、ファイルごとにエラーメッセージが戻される。そして、UID と GID はアーカイブにおいて nobody に設定される
1000000 以上	<ul style="list-style-type: none"> ■ この分類のユーザーが ar コマンドを使用すると、そのユーザーの UID と GID はアーカイブにおいて nobody に設定される
2097152 以上	<ul style="list-style-type: none"> ■ この分類のユーザーが tar コマンド、cpio -H ustar コマンド、または pax -x tar コマンドを使用すると、そのユーザーの UID と GID は nobody に設定される

パスワード

ユーザーを追加するときにそのユーザーのパスワードを指定できます。または、ユーザーが最初にログインするときにパスワードを指定するよう強制できます。ユーザーのパスワードは、次の構文に準拠している必要があります。

- パスワード長は、`/etc/default/passwd` ファイル内の `PASSLENGTH` 変数に指定された値に一致している必要がある。`PASSLENGTH` にはデフォルトで 6 が設定されている
- パスワードの最初の 6 文字には、2 文字以上の英字と 1 文字以上の数字または特殊文字を含める必要がある

ユーザー名は公表されますが、パスワードを知っているのは各ユーザーだけでなければなりません。各ユーザーアカウントには、6～8 文字の英数字と特殊文字を組み合わせたパスワードを割り当ててください。

コンピュータシステムのセキュリティを強化するには、ユーザーにパスワードを定期的に変更するよう指示してください。高いレベルのセキュリティを確保するには、ユーザーに 6 週間ごとにパスワードを変更するよう要求してください。低いレベルのセキュリティなら、3 ヶ月に 1 度で十分です。システム管理用のログイン (`root` や `sys` など) は、毎月変更するか、`root` のパスワードを知っている人が退職したり交替したりするたびに交換してください。

コンピュータセキュリティが破られる原因の多くは、正当なユーザーのパスワードが解読される場合です。ユーザーについて何か知っている人が簡単に推測できるような固有名詞、名前、ログイン名、パスワードを使わないよう各ユーザーに対して指示してください。

良いパスワードの例としては以下のようなものが考えられます。

- 英語の単語を組み合わせたフレーズ (たとえば、`beammeup`)
- フレーズ内の各単語の頭文字だけを集めた、意味のない文字列。たとえば、`SomeWhere Over The RainBow` から取った `swotrb`
- 文字を数字や記号に代えた単語。たとえば、`snoopy` から文字を代えた `sn00py`

次のものは、パスワードに不適当です。

- 自分の名前そのもの、逆読み、飛ばし読みのもの
- 家族やペットの名前
- 免許証番号
- 電話番号
- 社会保健番号
- 従業員番号
- 趣味や興味に関連した名前
- 季節に関係のある名前 (たとえば 12 月に `Santa` を使うなど)
- 辞書にある単語

パスワードの有効期限を設定する

NIS+ または `/etc` 内のファイルを使用してユーザーアカウント情報を格納する場合は、ユーザーのパスワードにパスワード有効期限を設定できます。Solaris 9 12/02 リリース以降は、LDAP ディレクトリサービスでもパスワードの有効期限がサポートされています。

パスワード有効期限の設定によって、ユーザーに定期的なパスワード変更を強制したり、あるパスワードを保持するのに必要な最低日数以前にパスワードを変更するのを防止したりできます。不正ユーザーが、古くて使用されていないアカウントを使用して、発覚せずにシステムのアクセス権を得るような場合を防止するために、アカウントが無効になる日付を設定することができます。パスワードの有効期限属性を設定するには、`passwd` コマンドまたは Solaris 管理コンソールのユーザーツールを使用します。

ホームディレクトリ

ホームディレクトリは、ユーザーが独自のファイルを格納するのに割り当てられるファイルシステムの一部分です。ホームディレクトリに割り当てる大きさは、ユーザーが作成するファイルの種類、サイズ、および数によって異なります。

ホームディレクトリは、ユーザーのローカルシステムまたはリモートファイルサーバーのどちらにでも配置できます。どちらの場合も、慣例により、ホームディレクトリは `/export/home/username` として作成します。大規模なサイトでは、ホームディレクトリをサーバーに格納してください。ホームディレクトリのバックアップおよび復元を簡単にするには、`/export/homen` ディレクトリごとに別々のファイルシステムを使用してください。たとえば、`/export/home1` と `/export/home2` を使用します。

ホームディレクトリが配置される場所に関係なく、ユーザーは通常 `/home/username` という名前のマウントポイントを介してホームディレクトリにアクセスします。Autofs を使用してホームディレクトリがマウントされていると、どのシステムでも `/home` マウントポイントの下にディレクトリを作成することは許可されません。Autofs が使用されていると、システムはマウントされている `/home` を特別なものと認識します。ホームディレクトリを自動マウントする方法の詳細については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「autofs 管理作業の概要」を参照してください。

ネットワーク上の任意の位置からホームディレクトリを使用するには、`/export/home/username` ではなく、常に `$HOME` という環境変数の値によって参照するようにしてください。前者はマシンに固有の指定です。さらに、ユーザーのホームディレクトリで作成されるシンボリックリンクはすべて相対パス (たとえば `../.././x/y/x`) を使用する必要があります。こうすることによって、そのリンクはどのシステムにホームディレクトリがマウントされても有効になります。

ユーザーの作業環境

ファイルを作成して格納するホームディレクトリのほかに、ユーザーには仕事をするために必要なツールとリソースにアクセスできる環境が必要です。ユーザーがシステムにログインするとき、C、Korn、Bourne シェルなどユーザーの起動シェルで定義される初期設定ファイルによって、ユーザーの作業環境が決定されます。

ユーザーの作業環境を管理するのに便利な方法として、カスタマイズしたユーザー初期設定ファイル (.login、.cshrc、.profile など) をユーザーのホームディレクトリに置くという方法があります。ユーザー初期設定ファイルをユーザー用にカスタマイズする方法については、93 ページの「ユーザーの作業環境のカスタマイズ」を参照してください。ユーザー初期設定ファイルをカスタマイズした後、新しいユーザーアカウントを作成するときにそれらをユーザーのホームディレクトリに追加できます。

1 回だけ行う作業としてお勧めするのは、「スケルトン」ディレクトリと呼ばれる別々のディレクトリをサーバーに設定することです。ユーザーのホームディレクトリが格納されるサーバーと同じサーバーを使用できます。スケルトンディレクトリによって、タイプの異なるユーザーに合わせてカスタマイズしたユーザー初期設定ファイルを格納できます。

注 - システム初期設定ファイル (/etc/profile、/etc/.login など) を使用してユーザーの作業環境を管理しないでください。これらのファイルはローカルシステムに存在するため、一元管理されません。たとえば、Autofs を使用してネットワーク上の任意のシステムからユーザーのホームディレクトリをマウントした場合、ユーザーがシステム間を移動しても環境が変わらないよう保証するには、各システムでシステム初期設定ファイルを修正しなければなりません。

また、役割によるアクセス制御でユーザーアカウントをカスタマイズする方法もあります。詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

グループを管理するガイドライン

「グループ」とは、ファイルやその他のシステム資源を共有できるユーザーの集合のことです。たとえば、同じプロジェクトで作業するユーザーはグループを形成することになります。グループは、従来の UNIX グループのことです。

各グループには、名前、グループ識別 (GID) 番号、およびそのグループに属しているユーザー名のリストが必要です。システムは GID によって内部的にグループを識別します。ユーザーは次の 2 つの種類のグループに所属できます。

- 一次グループ - オペレーティングシステムが、ユーザーによって作成されたファイルに割り当てるグループです。各ユーザーは、1つの一次グループに所属していません。
- 二次グループ - ユーザーが所属する1つまたは複数のグループです。ユーザーは、最大15個の二次グループに所属できます。

ユーザーの二次グループは、場合によっては重要でないことがあります。たとえば、ファイルの所有権は、一次グループだけが関係し、二次グループは関係しません。ただし、アプリケーションによってはユーザーの二次グループが関係することがあります。たとえば、ユーザーは、Admintool ソフトウェアを使用するとき `sysadmin` グループ (グループ 14) のメンバーでなければなりません、グループ 14 がそのユーザーの現在の一次グループであるかどうかは問題にはなりません。

`groups` コマンドを使って、ユーザーが所属しているグループを表示できます。ユーザーは一度に1つの一次グループにしか所属できません。ただし、`newgrp` コマンドを使用して、ユーザーがメンバーとなっている他のグループに一時的に一次グループを変更することはできます。

ユーザーアカウントを追加するとき、ユーザーに一次グループを割り当てるか、デフォルトの `staff` (グループ10) を使用しなければなりません。一次グループは、すでに存在しているものでなければなりません。存在しない場合は、GID 番号でグループを指定します。ユーザー名は、一次グループに追加されません。ユーザー名が追加されると、リストが長くなりすぎるからです。ユーザーを新しい二次グループに割り当てる前に、そのグループを作成し、それに GID 番号を割り当てなければなりません。

グループは、システムにとってローカルになるか、またはネームサービスを介して管理することができます。グループ管理を単純にするために、NIS のようなネームサービス、または LDAP のようなディレクトリサービスを使用してグループメンバーを集中管理してください。

ユーザーアカウントとグループを管理するツール

次の表に、ユーザーとグループを管理するための推奨ツールを示します。これらのツールはすべて、Solaris 管理コンソールツール群に含まれています。Solaris 管理コンソールの起動および使用方法については、第2章を参照してください。

表 4-4 ユーザーとグループを管理するためのツール

Solaris 管理ツール	用途	使用情報
ユーザー	ユーザーを管理する	Solaris 管理コンソールオンラインヘルプ
ユーザーテンプレート	学生、技術者、教師のように特定のユーザーの属性セットを作成する	Solaris 管理コンソールオンラインヘルプ
権限	RBAC 権限を管理する	Solaris 管理コンソールオンラインヘルプ
管理役割	RBAC 管理の役割を管理する	Solaris 管理コンソールオンラインヘルプ
グループ	グループ情報を管理する	Solaris 管理コンソールオンラインヘルプ
プロジェクト	プロジェクト情報を管理する	Solaris 管理コンソールオンラインヘルプ
メーリングリスト	メーリングリストを管理する	Solaris 管理コンソールオンラインヘルプ

Solaris 管理コンソールを使わずにユーザーアカウントおよびグループを管理する場合に使用する Solaris 管理コマンドについては、表 1-6 を参照してください。これらのコマンドは、認証およびネームサービスサポートを含め、Solaris 管理ツールと同じ機能を提供します。

Solaris ユーザー管理ツールで実行できる作業

Solaris ユーザー管理ツールを使用すると、ローカルシステムまたはネームサービス環境のユーザーアカウントを管理できます。

次の表で、ユーザーツールのユーザーアカウント機能を使って実行可能な作業について説明します。

表 4-5 ユーザーアカウント管理作業

作業	説明	内容説明
ユーザーの追加	ユーザーをローカルシステムまたはネームサービスに追加できる	73 ページの「ユーザーアカウントとグループとは」および 74 ページの「ユーザーアカウント管理のガイドライン」
ユーザーテンプレートの作成	ユーザー、契約者、技術者など、同じグループのユーザーを作成するために、定義済みのユーザー属性のテンプレートを作成できる	同上
ユーザーテンプレートを使用したユーザーの追加	テンプレートを使って、定義済みのユーザー属性を使用してユーザーを追加できる	同上
ユーザーテンプレートの複製	定義済みのユーザー属性を少しだけ変更して使用したい場合は、ユーザーテンプレートを複製する。次に、必要な属性のみを変更する	同上
ユーザープロパティの設定	ユーザーを追加する前にユーザープロパティを使用し、ユーザーの追加時にユーザーテンプレートを使用するかどうか、ユーザー削除時に、デフォルトでホームディレクトリやメールボックスを削除するかどうかなどを設定できる	同上
複数ユーザーの追加	ユーザー名を入力したテキストファイルを指定、または自動的に一連のユーザー名を生成することにより、ローカルシステムまたはネームサービスに複数のユーザーを追加できる	同上
ユーザープロパティの表示および変更	ログインシェル、パスワード、またはパスワードオプションのようなユーザープロパティを表示または変更できる	同上
ユーザーへの権限割り当て	特定の管理作業の実行を許可する権限をユーザーに割り当てること	同上

表 4-5 ユーザーアカウント管理作業 (続き)

作業	説明	内容説明
ユーザーの削除	ユーザーをローカルシステムまたはネームサービスから削除することができる。またオプションでユーザーのホームディレクトリまたはメールを削除するかどうかを指定できる。ユーザーは、グループまたは役割からも削除される	同上

表 4-6 ユーザー権限の管理作業

作業	説明	内容説明
権限を付与する	管理者だけが実行できた特定のコマンドまたはアプリケーションの実行権限をユーザーに付与することができる	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の権利プロファイル」
既存の権限のプロパティの表示および変更	既存の権限を表示または変更できる	同上
承認の追加	承認、つまり役割またはユーザーに個別に付与できる権限を追加できる	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の承認」
承認の表示および変更	既存の承認を表示または変更できる	同上

表 4-7 ユーザーの役割の管理作業

作業	説明	内容説明
管理の役割の追加	RBAC の承認役割を追加できる	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の役割」
管理の役割への権限の割り当て	作業を実行できるように、役割に特定の権限を割り当てることができる	同上
管理の役割の変更	役割に権限を追加したり削除したりすることができる	同上

表 4-8 グループ管理作業

作業	説明	内容説明
グループの追加	ユーザーを追加する前にグループ名を使用できるように、ローカルシステムまたはネームサービスにグループを追加する	80 ページの「グループを管理するガイドライン」
グループへのユーザーの追加	グループが所有するファイルにユーザーがアクセスする場合、ユーザーをグループに追加する	同上
グループからのユーザーの削除	ユーザーがグループファイルにアクセスする必要がなくなった場合は、グループからユーザーを削除できる	同上

表 4-9 プロジェクト管理作業

作業	説明	内容説明
プロジェクトの作成および複製	新しいプロジェクトを作成できるか、または新しいプロジェクトで必要な属性が酷似している既存のプロジェクトがある場合は、そのプロジェクトを複製できる	Solaris 管理コンソールのオンラインヘルプ
プロジェクト属性の変更および表示	既存のプロジェクト属性を表示または変更できる	Solaris 管理コンソールのオンラインヘルプ
プロジェクトの削除	不要になった場合は、プロジェクトを削除できる	Solaris 管理コンソールのオンラインヘルプ

表 4-10 メーリングリスト管理作業

作業	説明	内容説明
メーリングリストの作成	メーリングリスト (電子メールのメッセージの宛先のリスト) を作成できる	Solaris 管理コンソールのオンラインヘルプ
メーリングリスト名の変更	メーリングリストの作成後、その内容を変更できる	Solaris 管理コンソールのオンラインヘルプ
メーリングリストの削除	不要になった場合は、メーリングリストを削除できる	Solaris 管理コンソールのオンラインヘルプ

Solaris 管理コンソールによるホームディレクトリの管理

Solaris 管理コンソールツールを使用してユーザーのホームディレクトリを管理するときは、次のことに注意してください。

- ユーザーツールの「ユーザーを追加 (Add User)」ウィザードを使用してユーザーアカウントを追加し、ユーザーのホームディレクトリを `/export/home/username` として指定すると、ホームディレクトリが自動マウントされるように自動的に設定されて、次のエントリが `passwd` ファイルに追加されます。

```
/home/username
```

- ユーザーツールを使用して、ホームディレクトリを自動マウントしないユーザーアカウントを設定するには、この機能を無効にするユーザーアカウントのテンプレートを設定するほかに方法はありません。その後、このテンプレートを使ってユーザーを追加します。「ユーザーを追加 (Add User)」ウィザードでこの機能を無効にすることはできません。
- `-x autohome=N` オプションを指定して `smuser add` コマンドを使用すると、ユーザーのホームディレクトリを自動マウントしないでユーザーを追加できます。ただし、`smuser delete` コマンドには、ユーザーを追加した後でホームディレクトリを削除するオプションはありません。その場合は、ユーザーツールを使用して、ユーザーとユーザーのホームディレクトリを削除する必要があります。

ユーザーアカウントの変更

既存のものと重複するユーザー名や UID 番号を定義しないかぎり、ユーザーアカウントのログイン名や UID 番号を変更する必要はありません。2つのユーザーアカウントが、同じユーザー名または UID 番号を持つ場合、次の手順に従ってください。

- 2つのユーザーアカウントが同じ UID 番号を持つ場合、ユーザーツールを使用して、どちらか一方のアカウントを削除し、もう一度、異なる UID 番号で追加します。ユーザーツールを使用して、既存のユーザーアカウントの UID 番号を変更することはできません。
- 2つのユーザーアカウントが同じユーザー名を持つ場合、ユーザーツールを使用して、どちらか一方のアカウントを修正し、ユーザー名を変更します。

ユーザーツールを使用してユーザー名を変更する場合でも、ユーザーのホームディレクトリが存在すれば、ホームディレクトリの所有権は変更されます。

ユーザーアカウントの中で変更できる情報に、ユーザーのグループメンバーシップがあります。ユーザーツールの「アクション (Action)」メニューの「プロパティ (Properties)」を選択すると、ユーザーの二次グループを追加したり、削除したりできます。また、グループツールを使ってグループのメンバーリストを直接修正したりすることもできます。

ユーザーアカウントの次の部分も変更できます。

- 説明 (コメント)
- ログインシェル
- パスワードおよびパスワードオプション
- ホームディレクトリおよびホームディレクトリアクセス
- 権限および役割

ユーザーアカウントの削除

ユーザーツールでユーザーアカウントを削除すると、`passwd` ファイル、`group` ファイル内のエントリが自動的に削除されます。さらに、ユーザーのホームディレクトリおよびメールディレクトリにあるファイルを削除します。

カスタマイズしたユーザー初期設定ファイルの追加

ユーザーツールを使って、カスタマイズしたユーザー初期設定ファイルを作成することはできませんが、指定された「スケルトン」ディレクトリ内のユーザー初期設定ファイルでユーザーのホームディレクトリを生成することができます。このためには、ユーザーテンプレートツールを使ってユーザーテンプレートを作成し、コピーするユーザー初期設定ファイルを保存するスケルトンディレクトリを指定します。

`/etc/skel` ディレクトリにあるユーザー初期設定テンプレートをカスタマイズし、ユーザーのホームディレクトリへコピーできます。

パスワードの管理

ユーザーツールを使って、次のようなパスワード管理ができます。

- ユーザーアカウントに通常のパスワードを指定する
- ユーザーが最初のログイン時にパスワードを作成できるようにする
- ユーザーアカウントを無効にするかロックする
- 有効期限とパスワード有効期限情報を指定する

注 - パスワード有効期限は、NIS ネームサービスではサポートされません。

ユーザーアカウントを無効にする

一時的にまたは永続的に、ログインアカウントを無効にしなければならないことがあります。ユーザーアカウントを無効にしたりロックしたりすると、無効なパスワード *LK* がユーザーアカウントに割り当てられ、それ以後ログインできなくなります。

最も簡単にユーザーアカウントを無効にする方法は、ユーザーツールを使用してアカウントのパスワードをロックすることです。

「ユーザープロパティ (User Properties)」画面の「有効期限 (Expiration Date)」フィールドに有効期限を入力して、ユーザーアカウントの有効期間に制限を設けることもできます。

パスワード有効期限を設定するかパスワードを変更することによって、ユーザーアカウントを無効にすることもできます。

ユーザーアカウントとグループ情報の格納場所

ユーザーアカウントとグループ情報は、サイトの方針に応じて、ネームサービスまたはローカルシステムの /etc ファイルのどちらかに格納できます。情報は、NIS+ ネームサービスではテーブルに格納され、NIS ネームサービスではマップに格納され、LDAP ディレクトリサービスではインデックス付きのデータベースファイルに格納されます。

注 - 混乱を避けるために、ユーザーアカウントとグループ情報の位置は、「データベース」、「テーブル」、「マップ」という3種類の呼び方ではなく、単に「ファイル」と呼びます。

ほとんどのユーザーアカウント情報は、passwd ファイルに格納されます。ただし、パスワード暗号とパスワード有効期限は、NIS か NIS+ を使用するときは passwd ファイルに、/etc ファイルを使用するときは /etc/shadow ファイルに格納されます。NIS を使用するとき、パスワード有効期限は使用できません。

グループ情報は group ファイルに格納されます。

passwd ファイルのフィールド

passwd ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

```
username:password:uid:gid:comment:home-directory:login-shell
```

たとえば、次のようになります。

```
kryten:x:101:100:Kryten Series 4000 Mechanoid:/export/home/kryten:/bin/csh
```


次の表では、passwd ファイルの各フィールドについて説明します。

表 4-11 passwd ファイルのフィールド

フィールド名	説明
<i>username</i>	ユーザー (またはログイン) 名。ユーザー名は固有で、1～8 文字の英字 (A-Z, a-z) と数字 (0-9) を使用する。最初の文字は英字で、少なくとも 1 文字は小文字を使用する
<i>password</i>	暗号化されたパスワードの代わりに x (パスワードフィールドはもう使用されない)。暗号化されたパスワードは shadow ファイルに格納される
<i>uid</i>	ユーザーをシステムに識別させるユーザー識別番号 (UID)。一般ユーザーの UID 番号は 100 から 60000 までの範囲とする。UID 番号はすべて固有でなければならない
<i>gid</i>	ユーザーの一次グループのグループ識別番号 (GID)。GID 番号は 0 から 60002 までの範囲の整数で指定する。60001 と 60002 はそれぞれ nobody と noaccess に割り当てられる。65534 は nobody4 に割り当てられる
<i>comment</i>	通常はユーザーのフルネーム。このフィールドはコメントとしての情報専用。このフィールドは、もともとは、Bell 研究所の UNIX システムから GECOS (General Electric Computer Operating System) を実行するメインフレームにバッチジョブを依頼する場合、必要なログイン情報を保持するために使われていたため、GECOS フィールドと呼ばれることもある
<i>home-directory</i>	ユーザーのホームディレクトリのパス名
<i>login-shell</i>	ユーザーのデフォルトログインシェル。これは /bin/sh、/bin/csh、/bin/ksh のどれかになる。表 4-18 のシェル機能の説明を参照

デフォルトの passwd ファイル

Solaris のデフォルトの passwd ファイルには、標準のデーモン用のエントリが入っています。デーモンとは、通常ブート時に起動され、システム全体で有効な操作 (印刷、ネットワークの管理、ポートの監視など) を実行するプロセスです。

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1:/:
bin:x:2:2:/:usr/bin:
sys:x:3:3:/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
```

```
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

表 4-12 デフォルトの passwd ファイルのエントリ

ユーザー名	ユーザー ID	説明
root	0	スーパーユーザーのアカウント
daemon	1	ルーチンシステムタスクに関連するシステム包括デーモン
bin	2	ルーチンシステムタスクを実行するシステムバイナリの実行に関連する管理デーモン
sys	3	システムのログの記録や一時ディレクトリのファイルの更新に関連する管理デーモン
adm	4	システムのログの記録に関連する管理デーモン
lp	71	ラインプリンタのデーモン
uucp	5	uucp 関数に関連するデーモン
nuucp	6	uucp 関数に関連するデーモン
smmsp	25	Sendmail メッセージ送信プログラムデーモン
listen	37	ネットワーク監視デーモン
nobody	60001	特別なアクセス権を必要としない、あるいは持つべきではないユーザーまたはソフトウェアプロセスに割り当てられる
noaccess	60002	あるアプリケーションを経由するが実際にログインせずに、システムにアクセスする必要があるユーザーまたはプロセスに割り当てられる
nobody4	65534	SunOS 4.0 または 4.1 の nobody ユーザーアカウント

shadow ファイルのフィールド

shadow ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

```
username:password:lastchg:min:max:warn:inactive:expire
```

たとえば、次のようになります。

```
rimmer:86Kg/MNT/dGu.:8882:0::5:20:8978
```

次の表では、shadow ファイルの各フィールドについて説明します。

表 4-13 shadow ファイルのフィールド

フィールド名	説明
<i>username</i>	ユーザー (またはログイン) 名
<i>password</i>	次のエントリのいずれかになる。13 文字の暗号化されたユーザーパスワード。アクセス不可能なアカウントを示す *LK*。アカウントのパスワードがないことを示す NP
<i>lastchg</i>	1970 年 1 月 1 日から最後にパスワードを変更した日付までの日数
<i>min</i>	パスワードの変更から次の変更までに必要な最短日数
<i>max</i>	ユーザーが新しいパスワードの指定をもとめられるまで、パスワードを変更しないで使い続けることができる最長日数
<i>inactive</i>	アカウントを使用 (ログイン) しなくてもよい最長日数
<i>expire</i>	ユーザーアカウントの有効期限が切れる日付。この日付が過ぎると、ユーザーはシステムにログインできない

group ファイルのフィールド

group ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

```
group-name : group-password : gid : user-list
```

たとえば、次のようになります。

```
bin : 2 : root, bin, daemon
```

次の表では、group ファイルの各フィールドについて説明します。

表 4-14 group ファイルのフィールド

フィールド名	説明
<i>group-name</i>	グループに付けられた名前。たとえば、大学の化学部のメンバーであれば chem などと指定する。グループ名に許される最大文字数は 8 文字
<i>group-password</i>	通常は空のままか、アスタリスクを指定する。group-password フィールドは初期バージョンの UNIX のなごり。グループにパスワードがある場合、newgrp コマンドはユーザーにグループパスワードを入力するよう求める。ただし、グループパスワードを設定するためのユーティリティはない

表 4-14 group ファイルのフィールド (続き)

フィールド名	説明
<i>gid</i>	グループの GID 番号。ローカルシステムで固有にする必要があり、組織全体を通じても固有であることが望ましい。GID 番号は 0 から 60002 までの範囲の整数で指定する。ただし、100 未満の番号はシステムのデフォルトグループアカウント用に予約されている。したがって、ユーザー定義グループの範囲は 100 から 60000。60001 と 60002 はそれぞれ <i>nobody</i> と <i>noaccess</i> に割り当てられている
<i>user-list</i>	コンマで区切られたユーザー名のリスト。ユーザーの二次グループメンバーシップを表す。各ユーザーは最大 15 個の二次グループに所属できる

デフォルトの group ファイル

Solaris のデフォルトの group ファイルには、システム全体に有効な操作 (印刷、ネットワーク管理、電子メールなど) をサポートする次のようなシステムグループが記述されています。これらのグループの多くは、passwd ファイルのエントリに対応しています。

```

root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
smmsp::25:smmsp
sysadmin::14:root
nobody::60001:
noaccess::60002:
nogroup::65534:

```

表 4-15 デフォルトの group ファイルのエントリ

グループ名	グループ ID	説明
root	0	スーパーユーザーのグループ
other	1	オプションのグループ
bin	2	システムバイナリの実行に関連する管理グループ

表 4-15 デフォルトの group ファイルのエントリ (続き)

グループ名	グループ ID	説明
sys	3	システムのログの記録や一時ディレクトリに関連する管理グループ
adm	4	システムのログの記録に関連する管理グループ
uucp	5	uucp 関数に関連するグループ
mail	6	電子メールのグループ
tty	7	tty デバイスに関連するグループ
lp	8	ラインプリンタのグループ
nuucp	9	uucp 関数に関連するグループ
staff	10	一般的な管理グループ
daemon	12	ルーチンシステムタスクに関連するグループ
sysadmin	14	Admintool と Solstice AdminSuite ツールに関連する管理グループ
smmsp	25	Sendmail メッセージ送信プログラムデーモン
nobody	60001	特別なアクセス権を必要としない、あるいは持つべきではないユーザーまたはソフトウェアプロセスに割り当てられるグループ
noaccess	60002	あるアプリケーションを経由するが実際にログインをせずに、システムにアクセスする必要があるユーザーまたはプロセスに割り当てられるグループ
nogroup	65534	既知のグループのメンバーでないユーザーに割り当てられるグループ

ユーザーの作業環境のカスタマイズ

ユーザーのホームディレクトリの設定には、ユーザーのログインシェルにユーザー初期設定ファイルを提供することも含まれます。ユーザー初期設定ファイルは、ユーザーがシステムにログインしたあとにユーザーのために作業環境を設定するシェルスクリプトです。基本的にシェルスクリプトで行える処理はどれもユーザー初期設定ファイルで実行できます。主に、ユーザーの検索パス、環境変数、ウィンドウ機能の環境など、ユーザーの作業環境を定義します。次の表に示すように、各ログインシェルには、1つまたは複数の、固有のユーザー初期設定ファイルがあります。

表 4-16 Bourne、C、Korn シェルのユーザー初期設定ファイル

シェル	初期設定ファイル	目的
Bourne	\$HOME/.profile	ログイン時のユーザー環境の定義
C	\$HOME/.cshrc	すべての C シェルに対するユーザー環境の定義で、ログインシェルのあとに起動される
	\$HOME/.login	ログイン時のユーザー環境の定義
Korn	\$HOME/.profile	ログイン時のユーザー環境の定義
	\$HOME/\$ENV	ログイン時のユーザー環境の定義で、Korn シェルの ENV 環境変数によって指定される

Solaris 環境には、次の表に示すように、各システムの /etc/skel ディレクトリに、各シェル用のデフォルトのユーザー初期設定ファイルが提供されています。

表 4-17 デフォルトのユーザー初期設定ファイル

シェル	デフォルトファイル
C	/etc/skel/local.login
	/etc/skel/local.cshrc
Bourne または Korn	/etc/skel/local.profile

これらのファイルを変更して、すべてのユーザーに共通な作業環境を提供する標準のファイルセットを作成できます。または、異なるタイプのユーザーに作業環境を提供することもできます。異なるタイプのユーザーにユーザー初期設定ファイルを作成する手順については、107 ページの「ユーザー初期設定ファイルをカスタマイズする方法」を参照してください。

ユーザーツールで新しいユーザーアカウントを作成して、ホームディレクトリを作成するオプションを選択すると、選択したログインシェルに合わせて次のファイルが作成されます。

シェル	作成されるファイル
C	/etc/skel/local.cshrc ファイルと /etc/skel/local.login ファイルがユーザーのホームディレクトリにコピーされ、それぞれ、.cshrc と .login という名前に変更される
Bourne と Korn	/etc/skel/local.profile ファイルがユーザーのホームディレクトリにコピーされ、.profile という名前に変更される

useradd コマンドで新しいユーザーアカウント追加するために、`-k` オプションと `-m` オプションで `/etc/skel` ディレクトリを指定した場合、3つの `/etc/skel/local*` ファイルと `/etc/skel/.profile` ファイルがすべてユーザーのホームディレクトリにコピーされます。この時点で、これらのファイルの名前をユーザーのログインシェルに合わせて変更する必要があります。

サイト初期設定ファイルの使用方法

ユーザー初期設定ファイルは管理者とユーザーの両方によってカスタマイズできます。この重要な機能は、サイト初期設定ファイルと呼ばれる、グローバルに配布されるユーザー初期設定ファイルによって実現します。サイト初期設定ファイルを使用して、ユーザーの作業環境に新しい機能を絶えず導入でき、しかもユーザーはユーザー初期設定ファイルをカスタマイズすることもできます。

ユーザー初期設定ファイルでサイト初期設定ファイルを参照するとき、サイト初期設定ファイルに対して行なったすべての更新は、ユーザーがシステムにログインするときかユーザーが新しいシェルを起動するとき自動的に反映されます。サイト初期設定ファイルは、ユーザーを追加したときにはなかったサイト全体の変更をユーザーの作業環境に配布するよう設計されています。

ユーザー初期設定ファイルでできるカスタマイズは、サイト初期設定ファイルでも行えます。これらのファイルは通常はサーバー、またはサーバーのグループにあり、ユーザー初期設定ファイルの最初の行に現れます。また、各サイト初期設定ファイルは、それを参照するユーザー初期設定ファイルと同じ型のシェルスクリプトでなければなりません。

C シェルのユーザー初期設定ファイルでサイト初期設定ファイルを参照するには、ユーザー初期設定ファイルの最初に次のような行を挿入してください。

```
source /net/machine-name/export/site-files/site-init-file
```

Bourne または Korn シェルのユーザー初期設定ファイルでサイト初期設定ファイルを参照するには、ユーザー初期設定ファイルの最初に次のような行を入れてください。

```
./net/machine-name/export/site-files/site-init-file
```

ローカルシステムへの参照を避ける

ユーザー初期設定ファイルに、ローカルシステムへの固有の参照は追加しないでください。初期設定ファイルの設定は、ユーザーがどのシステムにログインしても有効になる必要があります。たとえば、次のようになります。

- ユーザーのホームディレクトリをネットワーク上の任意の位置で利用できるようにするには、常に環境変数の値 `$HOME` を使用してホームディレクトリを参照してください。たとえば、`/export/home/username/bin` ではなく `$HOME/bin` を使用してください。`$HOME` は、ユーザーが別のシステムにログインする場合でも有効

で、その場合ホームディレクトリは自動マウントされます。

- ローカルディスクのファイルにアクセスするには、`/net/system-name/directory-name` などのグローバルパス名を使用してください。システムが `Autofs` を実行していれば、`/net/system-name` で参照されるディレクトリはすべてユーザーがログインする任意のシステムに自動的にマウントできます。

シェル機能

次の表に各シェルの基本的な機能を示します。ユーザー初期設定ファイルを作成するのにどのシェルがどんな機能を提供するか参考にしてください。

表 4-18 Bourne、C、Korn シェルの基本機能

機能	Bourne	C	Korn
UNIX で標準シェルとして知られる	O	X	X
Bourne シェルと互換性がある構文	-	X	O
ジョブ制御	O	O	O
履歴リスト	X	O	O
コマンド行の編集	X	O	O
別名 (エイリアス)	X	O	O
ログインディレクトリの 1 文字省略形	X	O	O
ファイルの上書き保護 (noclobber)	X	O	O
CTRL + D 無視 (ignoreeof)	X	O	O
拡張 cd	X	O	O
.profile とは別の初期設定ファイル	X	O	O
ログアウトファイル	X	O	X

シェル環境

シェルは、`login` プログラム、システム初期設定ファイル、ユーザー初期設定ファイルによって定義される変数を含む環境を管理します。また、一部の変数はデフォルトで定義されます。シェルには次の 2 種類の変数があります。

- 環境変数 - シェルによって生成されるすべてのプロセスにエクスポートされる変数。環境変数の設定値は `env` コマンドで表示できます。PATH などを含む環境変数の一部が、シェルそのものの動作に影響を与えます。

- シェル (ローカル) 変数 – 現在使用中のシェルだけに関係する変数。C シェルの場合は、シェル変数は環境変数と特別に対応しています。これらのシェル変数は `user`、`term`、`home`、`path` です。シェル変数は、対応する環境変数の値によって初期設定されます。

C シェルでは、小文字を使って `set` コマンドでシェル変数を設定し、大文字を使って `setenv` コマンドで環境変数を設定します。シェル変数を設定すると、対応する環境変数が設定され、その逆もあります。たとえば、`path` シェル変数を新しいパスで更新すると、シェルは `PATH` 環境変数も新しいパスで更新します。

Bourne、Korn の両シェルでは、何らかの値に等しい大文字の変数名を使ってシェル変数と環境変数を設定できます。また、`export` コマンドを使って、その後に実行されるコマンドの変数をアクティブにする必要があります。

すべてのシェルで、シェル変数と環境変数は一般的に大文字の名前で参照します。

ユーザー初期設定ファイルで、ユーザーのシェル環境を、あらかじめ定義された変数の値を変更するか、変数を追加することによってカスタマイズできます。次の表に、ユーザー初期設定ファイルで環境変数を設定する方法を示します。

表 4-19 ユーザー初期設定ファイルでの環境変数の設定方法

環境変数を設定したいシェルタイプ	ユーザー初期設定ファイルに追加する行
C シェル	<code>setenv VARIABLE value</code> 例: <code>setenv MAIL /var/mail/ripley</code>
Bourne または Korn シェル	<code>VARIABLE =value; export VARIABLE</code> 例: <code>MAIL=/var/mail/ripley;export MAIL</code>

次の表では、ユーザー初期設定ファイルでカスタマイズできる環境変数とシェル変数について説明します。各シェルで使用される変数の詳細については、`sh(1)`、`ksh(1)`、`csch(1)` の各マニュアルページを参照してください。

表 4-20 シェル変数と環境変数の説明

変数	説明
CDPATH (C シェルでは cdpath)	cd コマンドで使用する変数を設定する。cd コマンドの対象ディレクトリを相対パス名で指定すると、cd コマンドは対象ディレクトリをまず現在のディレクトリ (.) 内で検索する。対象ディレクトリが見つからなかった場合は、CDPATH 変数のリストの順で検索され、見つかると、ディレクトリの変更が行われる。CDPATH で対象ディレクトリが見つからなかった場合は、現在の作業ディレクトリは変更されない。たとえば、CDPATH 変数を /home/jean に設定し、その下に bin と rje の 2 つのディレクトリがある場合、/home/jean/bin ディレクトリの中で cd rje と入力すると、絶対パスを指定しなくても、ディレクトリを /home/jean/rje に変更することになる
history	C シェルの履歴を設定する
HOME (C シェルでは home)	ユーザーのホームディレクトリへのパスを設定する
LANG	ロケールを設定する
LOGNAME	現在ログインしているユーザーの名前を設定する。LOGNAME のデフォルト値は、passwd ファイルに指定されているユーザー名にログインプログラムによって自動的に設定される。この変数は参照用のみ使用し、設定を変更してはならない
LPDEST	ユーザーのデフォルトプリンタを設定する
MAIL	ユーザーのメールボックスへのパスを設定する
MANPATH	アクセスできるマニュアルページの階層を設定する
PATH (C シェルでは path)	ユーザーがコマンドを入力したときに実行するプログラムについて、シェルが検索するディレクトリを順番に指定する。ディレクトリが検索パス上にない場合は、ユーザーはコマンドの絶対パス名を入力しなければならない デフォルトの PATH は、ログインプロセスで .profile (Bourne または Korn シェル) または .cshrc (C シェル) の指定どおりに自動的に定義され、設定される 検索パスの順序が重要となる。同じコマンドが異なる場所にそれぞれ存在するときは、最初に見つかったコマンドが使用される。たとえば、PATH が Bourne および Korn シェル構文で PATH=/bin:/usr/bin:/usr/sbin:\$HOME/bin のように定義されていて、sample というファイルが /usr/bin と /home/jean/bin の両方にあるものとする。ユーザーが sample コマンドを、その絶対パスを指定しないで入力した場合は、/usr/bin で見つかったバージョンが使用される
prompt	C シェルのシェルプロンプトを設定する
PS1	Bourne または Korn シェルのシェルプロンプトを設定する

表 4-20 シェル変数と環境変数の説明 (続き)

変数	説明
SHELL (C シェルではshell)	make、vi、その他のツールが使うデフォルトシェルを設定する
TERMINFO	terminfo ファイルに追加した、サポートされていない端末のパス名を指定する。/etc/profile または /etc/.login で TERMINFO 変数を使用する TERMINFO 環境変数を設定すると、システムはまずユーザーが定義した TERMINFO パスを調べる。ユーザーが定義した TERMINFO ディレクトリ内に端末の定義が見つからなかった場合は、システムはデフォルトディレクトリ /usr/share/lib/terminfo で定義を探す。どちらにも見つからなかった場合、端末は dumb として定義される
TERM (C シェルではterm)	端末を設定する。この変数は /etc/profile または /etc/.login で再設定する必要がある。ユーザーがエディタを起動すると、システムはこの環境変数の定義と同じ名前のファイルを探す。システムは、TERMINFO が参照するディレクトリ内を探して端末の特性を知る
TZ	時間帯を設定する。これは、たとえば ls -l コマンドで日付を表示する場合に使われる。TZ をユーザーの環境に設定しないと、システムの設定が使用される。設定する場合、グリニッジ標準時が使用される

PATH 変数

ユーザーが絶対パス名でコマンドを入力すると、シェルはそのパス名を使ってコマンドを探します。ユーザーがコマンド名しか指定しないと、シェルは PATH 変数で指定されているディレクトリの順でコマンドを探します。コマンドがいずれかのディレクトリで見つければ、シェルはコマンドを実行します。

デフォルトのパスがシステムで設定されますが、大部分のユーザーはそれを変更して他のコマンドディレクトリを追加します。環境の設定や、正しいバージョンのコマンドまたはツールへのアクセスに関連して発生するユーザーの問題の多くは、パス定義の誤りが原因です。

パスの設定のガイドライン

次に、効率的な PATH 変数を設定するためのガイドラインをいくつか示します。

- セキュリティが特に問題とならないときは、現在の作業ディレクトリ (.) をパスの最初に指定します。しかし、現在の作業ディレクトリをパスに入れると、セキュリティ上の問題となることがあり、特にスーパーユーザーにとって問題となります。
- 検索パスはできるだけ短くしておきます。シェルはパスで各ディレクトリを探します。コマンドが見つからないと、検索に時間がかかり、システムのパフォーマンスが低下します。
- 検索パスは左から右に読まれるため、通常使用するコマンドをパスの初めの方に指定するようにしてください。

- パスでディレクトリが重複していないか確認してください。
- 可能であれば、大きなディレクトリの検索は避けてください。大きなディレクトリはパスの終わりに指定します。
- NFS™ サーバーが応答しないときに「ハング」の可能性を少なくしたり、不要なネットワークトラフィックを削減するよう、NFS がマウントするディレクトリより前にローカルディレクトリを指定します。

例 — ユーザーのデフォルトパスの設定

次の例は、ユーザーのデフォルトパスがホームディレクトリと他の NFS マウントディレクトリを含むように設定する方法を示します。現在の作業ディレクトリはパスの最初に指定されます。C シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
set path=(. /usr/bin $HOME/bin /net/glrr/files1/bin)
```

Bourne または Korn シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
PATH=./usr/bin:/$HOME/bin:/net/glrr/files1/bin
export PATH
```

ロケール変数

LANG および LC 環境変数は、時間帯と照合順序、および日付、時間、通貨、番号の書式など、ロケール固有の変換と表記をシェルに指定します。さらに、ユーザー初期設定ファイルで stty コマンドを使って、システムが複数バイト文字をサポートするかどうかを設定できます。

LANG は、ロケールのすべての変換と表記を設定します。特に必要な場合、これとは別に LC_COLLATE、LC_CTYPE、LC_MESSAGES、LC_NUMERIC、LC_MONETARY、LC_TIME の各 LC 変数によってその他の設定を行えます。

次の表は、LANG と LC 環境変数の値の一部を示します。

表 4-21 LANG と LC 変数の値

値	ロケール
de	German
fr	French
iso_8859_1	English および European
it	Italian

表 4-21 LANG と LC 変数の値 (続き)

値	ロケール
japanese	Japanese
korean	Korean
sv	Swedish
tchinese	Taiwanese

例 — LANG 変数によるロケールの設定

次の例は、LANG 環境変数を使ってロケールを設定する方法を示しています。C シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
setenv LANG de
```

Bourne または Korn シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
LANG=DE; export LANG
```

デフォルトのファイルアクセス権 (umask)

ファイルまたはディレクトリを作成したときに設定されるデフォルトのファイルアクセス権は、ユーザーマスクによって制御されます。ユーザーマスクは、初期設定ファイルで umask コマンドによって設定されます。現在のユーザーマスクの値は、umask と入力して Return キーを押すと表示できます。

ユーザーマスクは、次の 8 進値で構成されます。

- 最初の桁でそのユーザーのアクセス権を設定する
- 2 桁目でグループのアクセス権を設定する
- 3 桁目で「その他」(ワールドとも呼ばれる)のアクセス権を設定する

最初の桁がゼロの場合、その桁は表示されません。たとえば、umask を 022 に設定すると、22 が表示されます。

設定する umask の値は、与えたいアクセス権の値を 666 (ファイルの場合) または 777 (ディレクトリの場合) から引きます。引いた残りが umask に使用する値です。たとえば、ファイルのデフォルトモードを 644 (rw-r--r--) に変更したいとします。このとき 666 と 644 の差 022 が umask コマンドの引数として使用する値です。

また、次の表から umask 値を決めることもできます。この表は、umask の各 8 進値から作成される、ファイルとディレクトリのアクセス権を示します。

表 4-22 umask 値のアクセス権

umask 8 進値	ファイルアクセス権	ディレクトリアクセス権
0	rw-	rwx
1	rw-	rw-
2	r--	r-x
3	r--	r--
4	-w-	-wx
5	-w-	-w-
6	--x	--x
7	--- (なし)	--- (なし)

次の例は、デフォルトのファイルアクセス権を rw-rw-rw- に設定します。

```
umask 000
```

ユーザー初期設定ファイルとサイト初期設定ファイルの例

ここでは、ユーザー自身の初期設定ファイルをカスタマイズする場合に使用する、ユーザー初期設定ファイルとサイト初期設定ファイルの例を示します。例の中のシステム名やパス名は、実際のサイトに合わせて置き換えてください。

例 — .profile ファイル

```

1 PATH=$PATH:$HOME/bin:/usr/local/bin:/usr/ccs/bin:.
2 MAIL=/var/mail/$LOGNAME
3 NNTPSERVER=server1
4 MANPATH=/usr/share/man:/usr/local/man
5 PRINTER=printer1
6 umask 022
7 export PATH MAIL NNTPSERVER MANPATH PRINTER

```

1. ユーザーのシェル検索パスを設定する
2. ユーザーのメールファイルへの検索パスを設定する
3. ユーザーの Usenet ニュースサーバーを設定する
4. マニュアルページへのユーザーの検索パスを設定する
5. ユーザーのデフォルトプリンタを設定する
6. ユーザーのデフォルトのファイル作成アクセス権を設定する
7. 指定された環境変数をエクスポートする

例 — .cshrc ファイル

```
1 set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)
2 setenv MAIL /var/mail/$LOGNAME
3 setenv NNTPSERVER server1
4 setenv PRINTER printer1
5 alias h history
6 umask 022
7 source /net/server2/site-init-files/site.login
```

1. ユーザーのシェル検索パスを設定する
2. ユーザーのメールファイルへの検索パスを設定する
3. ユーザーの Usenet ニュースサーバーを設定する
4. ユーザーのデフォルトプリンタを設定する
5. history コマンドの別名を作成する。これにより、h と入力するだけで history コマンドを実行できる
6. ユーザーのデフォルトのファイル作成アクセス権を設定する
7. 指定されたサイト初期設定ファイルを読み込む

例 — サイト初期設定ファイル

次のサイト初期設定ファイルの例では、ユーザーは特定のバージョンのアプリケーションを選択できます。

```
# @(#)site.login
main:
echo "Application Environment Selection"
echo ""
echo "1. Application, Version 1"
echo "2. Application, Version 2"
echo ""
echo -n "Type 1 or 2 and press Return to set your
application environment: "

set choice = $<

if ( $choice !~ [1-2] ) then
goto main
endif

switch ($choice)
case "1":
setenv APPHOME /opt/app-v.1
breaksw

case "2":
setenv APPHOME /opt/app-v.2
endsw
```

次のようにして、このサイト初期設定ファイルをユーザーの `.cshrc` ファイル (C シェルユーザーのみ使用可能) で参照させることができます。

```
source /net/server2/site-init-files/site.login
```

この行では、サイト初期設定ファイルは `site.login` という名前で、`server2` という名前のサーバー上にあります。また、自動マウンタがユーザーのシステムで実行されていることを前提としています。

第 5 章

ユーザーアカウントとグループの管理 (手順)

この章では、Solaris 管理コンソールを使ってユーザーアカウントとグループを設定し、管理する方法について説明します。

Solaris 管理コンソールを使ったユーザーアカウントとグループの設定および管理に関連する手順については、105 ページの「ユーザーアカウントの設定 (作業マップ)」および 115 ページの「ユーザーアカウントの管理 (作業マップ)」を参照してください。

ユーザーアカウントとグループ管理の概要については、第 4 章を参照してください。

ユーザーアカウントの設定 (作業マップ)

作業	説明	参照先
(省略可能) ユーザー情報の収集	標準の書式を使ってユーザー情報を収集すると、情報を整理しやすくなる	106 ページの「ユーザー情報を収集する方法」
(省略可能) ユーザー初期設定ファイルのカスタマイズ	新規ユーザーに一貫した環境を提供できるようにユーザー初期設定ファイル (.cshrc、.profile、.login) を設定する	107 ページの「ユーザー初期設定ファイルのカスタマイズする方法」
(省略可能) グループの追加	次のツールを使ってグループを追加できる	

作業	説明	参照先
	Solaris 管理コンソールのグループツール	109 ページの「Solaris 管理コンソールのグループツールを使ってグループを追加する方法」
	Solaris コマンド行インタフェースツール	111 ページの「CLI ツールを使ってグループやユーザーを追加する方法」
ユーザーの追加	次のツールを使ってユーザーを追加できる	
	Solaris 管理コンソールのユーザーツール	110 ページの「Solaris 管理コンソールのユーザーツールを使ってユーザーを追加する方法」
	Solaris コマンド行インタフェースツール	111 ページの「CLI ツールを使ってグループやユーザーを追加する方法」
(省略可能) ユーザーテンプレートの設定	類似したユーザープロパティをすべて手動で追加する必要がないようにユーザーテンプレートを作成できる	Solaris 管理コンソールオンラインヘルプ
(省略可能) ユーザーへの権限または役割の追加	特定のコマンドまたは操作を実行できるようにユーザーに権限または役割を追加する	Solaris 管理コンソールオンラインヘルプ
ユーザーのホームディレクトリの共有	ユーザーのホームディレクトリを共有することによって、ユーザーのシステムからそのディレクトリをリモートでマウントできる	112 ページの「ユーザーのホームディレクトリを共有する方法」
ユーザーのホームディレクトリのマウント	ユーザーのホームディレクトリをユーザーのシステムにマウントする	113 ページの「ユーザーのホームディレクトリをマウントする方法」

ユーザー情報を収集する方法

ユーザーアカウントを追加する前に、ユーザーに関する情報を次のような書式で記録しておくことができます。

項目	説明
ユーザー名	

役割名	
プロファイルまたは承認	
ユーザー名	
UID	
一次グループ	
二次グループ	
コメント	
デフォルトシェル	
パスワードの状態と有効期限	
ホームディレクトリのサーバー名	
ホームディレクトリのパス名	
マウント方法	
ホームディレクトリのアクセス権	
メールサーバー	
所属部署	
部署管理者	
管理者	
従業員名	
役職	
資格	
従業員番号	
開始日	
メール別名への追加	
デスクトップシステム名	

▼ ユーザー初期設定ファイルをカスタマイズする方法

1. ユーザーのホームディレクトリを作成し、共有するシステムでスーパーユーザーになるか、同等の役割を引き受けます。
2. 各タイプのユーザー用にスケルトンディレクトリを作成します。

```
# mkdir /shared-dir/skel/user-type
```

shared-directory ネットワーク上の別のシステムで利用できるディレクトリ
の名称

user-type ユーザーのタイプに応じて初期設定ファイルを格納する
ディレクトリの名称

3. デフォルトのユーザー初期設定ファイルを、異なるタイプのユーザー用に作成したディレクトリにコピーします。

```
# cp /etc/skel/local.cshrc /shared-dir/skel/user-type/.cshrc
# cp /etc/skel/local.login /shared-dir/skel/user-type/.login
# cp /etc/skel/local.profile /shared-dir/skel/user-type/.profile
```

注 - アカウントにプロファイルが割り当てられている場合、プロファイルに割り当てられた (セキュリティ属性付きの) コマンドを使用するために、ユーザーは特別なバージョンのシェルスであるプロファイルシェルを起動する必要があります。シェルの種類に合わせて、`pfsh` (Bourne シェル)、`pfcsh` (C シェル)、および `pfksh` (Korn シェル) の 3 つのプロファイルシェルがあります。

4. 各ユーザータイプ用にユーザー初期設定ファイルを編集し、必要に応じてカスタマイズします。

ユーザー初期設定ファイルをカスタマイズする方法については、93 ページの「ユーザーの作業環境のカスタマイズ」を参照してください。

5. ユーザー初期設定ファイルのアクセス権を設定します。

```
# chmod 744 /shared-dir/skel/user-type/.*
```

6. ユーザー初期設定ファイルのアクセス権が正しいことを確認します。

```
# ls -la /shared-dir/skel/*
```

例 — ユーザー初期設定ファイルをカスタマイズする

次の例では、特定のタイプのユーザー向けの、`/export/skel/enduser` ディレクトリにある C シェルユーザー初期設定ファイルをカスタマイズします。`.cshrc` ファイルの例については、103 ページの「例 — `.cshrc` ファイル」を参照してください。

```
# mkdir /export/skel/enduser
# cp /etc/skel/local.cshrc /export/skel/enduser/.cshrc
```

(`.cshrc` ファイルの編集)

```
# chmod 744 /export/skel/enduser/.*
```

▼ Solaris 管理コンソールのグループツールを使ってグループを追加する方法

システムにグループを追加するには、次の手順を実行します。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris** 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」または 64 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」を参照してください。

3. ナビゲーション区画の「管理ツール (**Management Tools**)」アイコンの下にある「このコンピュータ (**This Computer**)」アイコンをダブルクリックします。カテゴリのリストが表示されます。
4. (省略可能) ネームサービス環境に適したツールボックスを選択します。
5. 「**System Configuration**」アイコンをダブルクリックします。
6. 「ユーザー (**Users**)」アイコンをクリックします。
7. スーパーユーザーのパスワードまたは役割のパスワードを入力します。
8. 「グループ (**Groups**)」アイコンをクリックします。コンテキストヘルプを使用して、グループをシステムに追加します。

例 — Solaris 管理コンソールのグループツールを使ってグループを追加する

次の例では、グループ `mechanoids` (グループ ID 101) をシステム `starbug` に追加します。この例では、起動ツールが起動済みで、ユーザーツールが開いていることを前提とします。

グループを追加するときに既存のユーザーをグループに追加できます。また、グループだけを追加し、その後でユーザーを追加するときに既存のユーザーをそのグループに追加することもできます。

- 「アクション (Action)」メニューから「グループを追加 (Add Group)」を選択します。
- 「グループ ID (Group Identification)」の下にある「グループ (Group Name)」プロンプトで、グループ名 `mechanoids` を指定します。
- 「グループ ID 番号 (Group ID number)」プロンプトで、グループ番号 101 を指定します。
- 「了解 (OK)」をクリックします。

▼ Solaris 管理コンソールのユーザーツールを使ってユーザーを追加する方法

ユーザーをシステムに追加するには、次の手順を実行します。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris** 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」または 64 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」を参照してください。

3. ナビゲーション区画の「管理ツール (**Management Tools**)」アイコンの下にある「このコンピュータ (**This Computer**)」アイコンをダブルクリックします。カテゴリのリストが表示されます。
4. (省略可能) ネームサービス環境に適したツールボックスを選択します。
5. 「**System Configuration**」アイコンをダブルクリックします。
6. 「ユーザー (**Users**)」アイコンをクリックします。
7. スーパーユーザーのパスワードまたは役割のパスワードを入力します。
8. 「ユーザーアカウント (**User Accounts**)」アイコンをクリックします。コンテキストヘルプを使用して、ユーザーをシステムに追加します。

例 — Solaris 管理コンソールのグループツールを使ってユーザーを追加する

次の例では、ユーザー kryten (ユーザー ID 1001) をシステム starbug に追加します。この例では、起動ツールが起動済みで、ユーザーツールが開いていることを前提とします。

次の手順に進むときは「次へ (Next)」をクリックしてください。

- 「アクション (Action)」メニューから「ユーザーを追加 (Add User)」→「ウィザードを使用 (With Wizard)」を選択します。
- 手順 1 - 「ユーザー名 (User Name)」プロンプトで、ユーザー名またはログイン名 kryten を指定します。
- (省略可能) 「正式名 (Full Name)」プロンプトで、ユーザーのフルネーム kryten series 3000 を指定します。
- (省略可能) 「備考欄 (Description)」プロンプトで、このユーザーの詳細な説明を入力します。

- 手順 2 – 「ユーザー ID 番号 (User ID Number)」プロンプトで、ユーザー ID 1001 を入力します。
- 手順 3 – 「ユーザーは最初のログイン時にこのパスワードを使用する必要があります (User Must Use This Password At First Login)」オプションを選択します。
「パスワード (Password)」プロンプトでユーザーのパスワードを入力し、「パスワードを確認 (Confirm Password)」プロンプトでそのパスワードを確認します。
- 手順 4 – ユーザーの一次グループ `mechanoids` を選択します。
- 手順 5 – 「サーバー (Server)」と「パス (Path)」の各プロンプトでデフォルトを受け入れることにより、ユーザーのホームディレクトリを作成します。
- 手順 6 – メールサーバーを指定します。
- 手順 7 – 入力した情報を確認し、必要があれば前の画面に戻って情報を訂正します。訂正する必要がない場合は、「完了 (Finish)」をクリックします。

CLI ツールを使ってグループやユーザーを追加する方法

この節では、CLI ツールを使ってユーザーやグループを追加する例を示します。

例 — `groupadd` および `useradd` コマンドを使ってグループやユーザーを追加する

次の例では、`groupadd` および `useradd` の各コマンドを使って、グループ `scutters` やユーザー `scutter1` をローカルシステムのファイルに追加します。なお、これらのコマンドは、ネームサービス環境でユーザーを管理する場合には使用できません。

```
# groupadd -g 102 scutters
# useradd -u 1003 -g 102 -d /export/home/scutter1 -s /bin/csh
-c "Scutter 1" -m -k /etc/skel scutter1
64 blocks
```

詳細については、`groupadd(1M)` および `useradd(1M)` のマニュアルページを参照してください。

例 — `smgroup` および `smuser` コマンドを使ってグループやユーザーを追加する

次の例では、`smgroup` および `smuser` の各コマンドを使って、グループ `gelfs` やユーザー `camille` をホスト `starbug` の NIS ドメイン `solar.com` に追加します。

```
# /usr/sadm/bin/smgroup add -D nis:/starbug/solar.com -- -g 103 -n gelfs
# /usr/sadm/bin/smuser add -D nis:/starbug/solar.com -- -u 1004 -n camille
-c "Camille G." -d /export/home/camille -s /bin/csh -g gelfs
```

詳細については、smgroup(1M) および smuser(1M) のマニュアルページを参照してください。

▼ ユーザーのホームディレクトリを共有する方法

1. ユーザーのホームディレクトリを含むシステムでスーパーユーザーになるか、同等の役割を引き受けます。

2. 次のように入力して、**mountd** デーモンが動作していることを確認します。

```
# ps -ef | grep mountd
```

```
root 176 1 0 May 02 ? 0:19 /usr/lib/nfs/mountd
```

mountd デーモンが動作している場合には、`/usr/lib/nfs/mountd` と表示されます。

3. **mountd** デーモンが動作していない場合は、**mountd** デーモンを起動します。

```
# /etc/init.d/nfs.server start
```

4. システム上で共有されているファイルシステムをリスト表示します。

```
# share
```

5. ユーザーのホームディレクトリを含むファイルシステムがすでに共有されているかどうかによって、次のいずれかの手順を選択します。

- a. ユーザーのホームディレクトリがすでに共有されている場合、次の確認手順へ進みます。

- b. ユーザーのホームディレクトリが共有されていない場合、手順 6 に進みます。

6. `/etc/dfs/dfstab` ファイルを編集して、次の行を追加します。

```
share -F nfs /file-system
```

`/file-system` は、共有するユーザーのホームディレクトリを含むファイルシステムです。規約上、このファイルシステムは `/export/home` になります。

7. `/etc/dfs/dfstab` ファイルで指定されたファイルシステムを共有します。

```
# shareall -F nfs
```

このコマンドは、`/etc/dfs/dfstab` ファイルにある `share` コマンドをすべて実行するので、システムをリブートする必要はありません。

8. ユーザーのホームディレクトリが共有されていることを確認します。

```
# share
```


次に進む手順

ユーザーのホームディレクトリがユーザーのシステム上にない場合、それが配置されているシステムから、ユーザーのホームディレクトリをマウントしなければなりません。詳細な手順については、113 ページの「ユーザーのホームディレクトリをマウントする方法」を参照してください。

例 — ユーザーのホームディレクトリを共有する

```
# ps -ef | grep mountd
# /etc/init.d/nfs.server start
# share
# vi /etc/dfs/dfstab

(share -F nfs /export/home 行を追加)
# shareall -F nfs
# share
-                /usr/dist                ro    ""
-                /export/home/user-name    rw    ""
```

▼ ユーザーのホームディレクトリをマウントする方法

ホームディレクトリを自動マウントする場合は、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「autofs 管理作業の概要」を参照してください。

1. ユーザーのホームディレクトリが共有されていることを確認します。
詳細については、112 ページの「ユーザーのホームディレクトリを共有する方法」を参照してください。
2. ユーザーのシステムにスーパーユーザーとしてログインします。
3. **/etc/vfstab** ファイルを編集して、次のようなユーザーのホームディレクトリ用のエントリを作成します。

```
system-name:/export/home/user-name - /export/home/user-name nfs - yes rw
```

<i>system-name</i>	ホームディレクトリが配置されているシステムの名前
<i>/export/home/user-name</i>	共有されるユーザーのホームディレクトリの名前。規約上、 <i>/export/home/user-name</i> にユーザーのホームディレクトリが含まれる。ただし、別のファイルシステムでも構わない
-	エントリに必要な可変部分

`/export/home/user-name` ユーザーのホームディレクトリがマウントされるディレクトリの名前

エントリを `/etc/vfstab` ファイルに追加する方法については、第 40 章を参照してください。

4. ユーザーのホームディレクトリのマウント先を作成します。

```
# mkdir -p /export/home/user-name
```

5. ユーザーのホームディレクトリをマウントします。

```
# mountall
```

現在の `vfstab` ファイルにあるすべてのエントリ (このファイルの「mount at boot」フィールドが `yes` に設定されている) がマウントされます。

6. ホームディレクトリがマウントされているかどうかを確認します。

```
# mount | grep user-name
```

例 — ユーザーのホームディレクトリをマウントする

```
# vi /etc/vfstab
```

```
( venus:/export/home/ripley - /export/home/ripley
nfs - yes rw 行を追加)
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/proc on /proc read/write/setuid/dev=38c0000 on Sun Feb  2 18:20:07 2003
/etc/mnttab on mnttab read/write/setuid/dev=3980000 on Sun Feb  2 18:20:07 2003
/dev/fd on fd read/write/setuid/dev=39c0000 on Sun Feb  2 18:20:10 2003
/var/run on swap read/write/setuid/xattr/dev=1 on Sun Feb  2 18:20:11 2003
/tmp on swap read/write/setuid/xattr/dev=2 on Sun Feb  2 18:20:15 2003
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/onerror=...
/export/home/ripley on venus:/export/home/ripley remote/read/write/setuid/xattr/dev=...
```

ユーザーアカウントの管理 (作業マップ)

作業	説明	参照先
グループの変更	グループツールを使用して、グループ名またはグループ内のユーザーを変更する	Solaris 管理コンソールオンラインヘルプ
グループの削除	不要な場合には、グループを削除できる	Solaris 管理コンソールオンラインヘルプ
ユーザーアカウントの変更	「ユーザーアカウントを無効にする」 将来必要な場合、ユーザーアカウントを一時的に無効にすることができる 「ユーザーのパスワードを変更する」 ユーザーが忘れた場合、ユーザーのパスワードを変更する必要がある 「パスワード有効期限を変更する」 ユーザーアカウントツールの「パスワード (Password)」オプションメニューを使えば、定期的にユーザーにパスワードを変更させることができる	Solaris 管理コンソールオンラインヘルプ Solaris 管理コンソールオンラインヘルプ Solaris 管理コンソールオンラインヘルプ
ユーザーアカウントの削除	不要になった場合には、ユーザーアカウントを削除できる	Solaris 管理コンソールオンラインヘルプ

Solaris ユーザー登録

Solaris ユーザー登録は、新しい Solaris リリース情報、アップグレード情報、その他の販売情報を入手できるようにするためのツールです。このグラフィカルユーザーインターフェイス (GUI) は、ユーザーが最初にユーザーのデスクトップにログインしたと

きに、自動的に起動されます。このユーザー登録は、今すぐに登録することも、後で登録することも、あるいは登録しないことも可能です。ユーザー登録によって、Solaris のバージョン、ユーザー情報、プラットフォーム、ハードウェア、およびロケールの情報が Sun に提供されます。

Solaris Solve へのアクセス

Solaris ユーザー登録が完了すると、Solaris SolveSM にアクセスできるようになります。Solaris Solve は、Solaris の貴重な製品情報やソリューションが 1 箇所にまとめて提供されている会員制の Web サイトです。Solaris Solve を使用すると、Solaris の最新リリースに関する最新情報をすばやく簡単に入手できます。また、Sun のその他の契約やサービスについても知るすることができます。

Solaris ユーザー登録を完了して、Solaris Solve にアクセスする基本的な手順は次のとおりです。

1. Solaris ユーザー登録電子プロフィールに入力します。
2. 上記プロフィールを電子メールで送信するか、印刷したものを FAX か郵便で送ります。
3. Solaris Solve サイトにアクセスするためのログイン ID とパスワードを作成します。

Solaris Solve サイトにすぐにアクセスしない場合でも、Solaris ユーザーの登録時に自分の Solaris Solve ログイン ID とパスワードを作成しておくことをお勧めします。Solaris Solve ログイン ID とパスワードは、6～8 文字の英数字で作成します。空白とコロンは使用できません。

4. Solaris Solve サイトにアクセスします。

注 - システム管理者またはスーパーユーザーとしてログインすると、Solaris ユーザー登録は起動されません。

登録するように選択した場合、完成したフォームのコピーが \$HOME/.solregis/uprops に格納されます。登録しないように選択した場合でも、次のいずれかの方法でユーザー登録を起動して、後から登録することができます。

- コマンド行プロンプトで /usr/dt/bin/solregis と入力する
- アプリケーションマネージャのデスクトップツールフォルダにある、「Solaris ユーザー登録」アイコンをクリックする (CDE デスクトップの場合)

詳細については、solregis(1) のマニュアルページを参照してください。

Solaris ユーザー登録の問題の障害追跡

この節では、Solaris ユーザー登録に関する問題の障害追跡の方法について説明します。

次の表に、登録時に発生する可能性のある問題とその対処方法について説明します。

表 5-1 登録時の問題と対処方法

問題	対処方法
登録フォームが初期化されない。Web ページウィンドウに、システム管理者に登録設定を妨げている問題を解決してもらうように促すメッセージが表示される	登録ファイルが失われていないかチェックする
フォームを電子メールで送信できない。ダイアログボックスに、システム管理者に問題を解決してもらうように促すメッセージが表示される	電子メールが正しく設定されているかどうかをチェックする。完成した登録フォームを電子メールで送信するには CDE が必要なため、CDE がユーザーのシステム上にあるかを確認する。別の方法としてフォームを印刷して FAX か郵便で送信することもできる
フォームを印刷できない。ダイアログボックスに、システム管理者に問題を解決してもらうように促すメッセージが表示される	プリンタが正しく構成されているかどうかをチェックする。別の方法として電子メールでフォームを送信することもできる
フォームを保存できない。ダイアログボックスに、登録は成功したが、将来登録を更新するときに登録情報を呼び出せないことが示される。	ユーザーのホームディレクトリをチェックする。必要な処置は、システムの構成によって異なる
Solaris Solve のログイン ID とパスワードを忘れてしまった	SolarisSolve@sun.com に問題を説明するメールを英語で送るか、117 ページの「Solaris ユーザー登録をやり直す方法」を参照
登録プロセスをやり直したい	117 ページの「Solaris ユーザー登録をやり直す方法」

▼ Solaris ユーザー登録をやり直す方法

Solaris ユーザー登録を最初からやり直すには、次の手順に従ってください。

1. `$HOME/.solregis` ディレクトリに移動します。

```
% cd $HOME/.solregis
```

2. `uprops` ファイルを削除します。

```
% rm uprops
```

3. 登録プロセスを再起動します。

```
% /usr/dt/bin/solregis &
```

▼ ユーザー登録を無効にする方法

Solaris のインストール前またはインストール後に、ユーザー登録を無効にすることができます。Solaris ユーザー登録を無効にする前に、組織のシステム管理者をユーザー登録しておくことをお勧めします。

1. **Solaris** リリースをインストールする前にユーザー登録を無効にするには、次のいずれかを選択します。
 - SUNWsregu パッケージを選択解除する (対話式インストール)
 - SUNWsregu パッケージをインストールしないように、カスタム JumpStart プロファイルを変更する
 - 1つまたは複数のシステム上の /etc/default ディレクトリに、solregis という名前のファイルを作成するような finish スクリプトを作成して実行する。solregis は次の行を持つ必要があります。

```
DISABLE=1
```

詳細については、『Solaris 9 12/03 インストールガイド』または solregis(1) のマニュアルページを参照してください。
2. **Solaris** リリースをインストールした後にユーザー登録を無効にするには、次のいずれかを選択します。
 - SUNWsregu パッケージを削除する
 - /etc/default ディレクトリに上記の solregis ファイルを追加する

第 6 章

サーバーとクライアントサポートの管理

以下の各章では、サーバーおよびクライアントサポートの管理方法について説明します。

第 7 章

この章ではネットワークでサーバーとクライアントサポートを管理する場合について、管理者を対象に概要を説明します。また、サポートを追加可能なさまざまなシステムタイプおよび環境に適したシステムタイプを選択するためのガイドラインについて説明します。

第 8 章

`smosservice` コマンドおよび `smdiskless` コマンドを使用して、ディスクレスクライアントサポートを管理する手順について説明します。

第 7 章

サーバーとクライアントサポートの管理 (概要)

この章では、ネットワーク環境でのサーバーとクライアントの管理について説明し、Solaris 環境でサポートされる各システム構成(「システムタイプ」と呼びます)に関する情報を紹介します。また、目的に合った最適なシステムを選択するためのガイドラインも示します。

この章の内容は次のとおりです。

- 121 ページの「サーバーおよびクライアント管理の新機能」
- 122 ページの「サーバーとクライアントタスクの操作手順」
- 123 ページの「サーバー、クライアント、およびアプライアンスとは」
- 124 ページの「クライアントサポートとは」
- 124 ページの「システムタイプの概要」
- 128 ページの「ディスクレスクライアント管理の概要」

ディスクレスクライアントサポートの管理手順については、第 8 章を参照してください。

サーバーおよびクライアント管理の新機能

この節では、Solaris 9 リリースのサーバーとクライアントの新しい管理機能について説明します。

ディスクレスクライアントサポート

この Solaris 9 リリースでは、smosservice および smdiskless コマンドを使ってディスクレスクライアントを管理できます。ディスクレスクライアントとは、ディスクが搭載されておらず、すべてのサービスがサーバーに依存しているシステムのことです。

これらのコマンドは、Solaris 管理コンソールツール群の一部です。しかし、Solaris 管理コンソールでは、ディスクレスクライアントを管理できません。ディスクレスクライアントの管理に使用できるのは、smosservice コマンドおよび smdiskless コマンドに限られます。

ディスクレスクライアントの管理方法の詳細については、128 ページの「ディスクレスクライアント管理の概要」および第 8 章を参照してください。

サーバーとクライアントタスクの操作手順

サーバーとクライアントサポートの設定手順については、次の表を参照してください。

サーバー/クライアントサービス	参照先
インストールまたは JumpStart クライアント	『Solaris 9 12/03 インストールガイド』
Solaris 9 でのディスクレスクライアントシステム	128 ページの「ディスクレスクライアント管理の概要」および第 8 章
Solaris の以前のリリースのディスクレスクライアントシステムおよび Solstice AutoClient システム	『Solstice AdminSuite 2.3 管理者ガイド』
Solaris 8 または Solaris 9 での AutoClient 3.0.1 システム	サービスプロバイダにお問い合わせください。

サーバー、クライアント、およびアプライアンスとは

通常、ネットワーク上のシステムは、次のいずれかに該当します。

システムタイプ	説明
サーバー	ネットワーク上のほかのシステムにサービスを提供するシステム。ファイルサーバー、ブートサーバー、Web サーバー、データベースサーバー、ライセンスサーバー、印刷サーバー、インストールサーバー、さらに、特定のアプリケーション用のサーバーなどもあります。この章では、サーバーとは、ネットワーク上の他のシステムにブートサービスとファイルシステムを提供するシステムのことを意味します。
クライアント	サーバーから提供されるリモートサービスを利用するシステム。クライアントによってはディスク容量に制限があったり、まったくディスクを持たず、サーバーから提供されるファイルシステムに依存するものもあります。ディスクレスシステム、AutoClient システム、およびアプライアンスシステムは、このタイプのクライアントの一例です。 またサーバーが提供するリモートサービス (インストールソフトウェアなど) を利用しながらも、サーバーに依存しなくても機能するクライアントもあります。この種類のクライアントの例として、ルート (/)、/usr、/export/home ファイルシステムとスワップ空間をそのハードディスクに含むスタンドアロンシステムがあります。
Sun Cobalt サーバーアプライアンス	Sun Cobalt サーバーアプライアンスは、あらかじめ設定されたインターネットサービスの統合セットを提供します。サーバーアプライアンスのユーザーが必要なのは、Web ブラウザと IP アドレスだけです。サーバーは、一元管理されるので、アプライアンスのユーザーはクライアントを管理する必要がありません。詳細については、 http://www.sun.com/hardware/serverappliances を参照してください。
アプライアンス	Sun Ray アプライアンスのようなネットワークアプライアンスを使うと、アプリケーションや Solaris 環境にアクセスできます。アプライアンスを使うと、サーバー管理を一元化できるので、クライアント管理またはアップグレードを行う必要がなくなります。Sun Ray アプライアンスでは、「ホットデスク」も提供されます。この機能を使うと、サーバーグループのどのアプライアンスからでもただちに、作業中のコンピュータセッション (正確に言うと作業を中断したところ) にアクセスできます。詳細については、 http://www.sun.com/products/sunray を参照してください。

クライアントサポートとは

クライアントに対するサポートとは、クライアントの機能を促進するソフトウェアおよびサービスを提供することです。サポートには、次の項目が含まれます。

- システムをネットワークに認識させる (ホスト名と Ethernet アドレス情報)。
- システムをリモートからインストールおよびブートできるインストールサービスを提供する。
- ディスク容量に限りがある、またはディスク容量がないシステムにオペレーティングシステム (OS) サービスおよびアプリケーションサービスを提供する。

システムタイプの概要

システムタイプは、ルート (/) と /usr ファイルシステム (スワップ領域を含む) にアクセスする方法によって決まる場合があります。たとえば、スタンドアロンとサーバーシステムでは、これらのファイルシステムをローカルディスクからマウントしていますが、その他のクライアントでは、これらのファイルシステムをリモートからマウントし、サーバーから提供されるサービスに依存しています。次の表にそれぞれのシステムタイプの特徴を示します。

表 7-1 一般的なシステムタイプの特徴

システムタイプ	ローカルファイルシステム	ローカルスワップ領域	リモートファイルシステム	ネットワーク利用度	相対パフォーマンス
サーバー	ルート (/) /usr /home /opt /export/home /export/root	あり	なし	高	高
スタンドアロンシステム	ルート (/) /usr /export/home	あり	なし	低	高

表 7-1 一般的なシステムタイプの特徴 (続き)

システムタイプ	ローカルファイルシステム	ローカルスワップ領域	リモートファイルシステム	ネットワーク利用度	相対パフォーマンス
ディスクレスクライアント	なし	なし	ルート (/) スワップ /usr /home	高	低
AutoClient システム	キャッシュルート (/) キャッシュされた /usr	あり	/var	低	高
アプライアンス	なし	なし	なし	高	高

サーバー

サーバーシステム上には、次のファイルシステムがあります。

- ルート (/) と /usr ファイルシステム、およびスワップ領域
- /export と /export/home ファイルシステム。クライアントシステムをサポートし、ユーザーにホームディレクトリを提供する
- アプリケーションソフトウェアを格納する /opt ディレクトリまたはファイルシステム

サーバー上には、他のシステムをサポートするために次のソフトウェアも格納できます。

- サーバーとは異なるプラットフォームで、別のリリースまたはクライアントを稼働させるディスクレスシステムまたは AutoClient システム用のオペレーティングシステム (OS) サービス
- ネットワークに接続されたシステムがリモートインストールを実行するのに必要な Solaris CD のイメージソフトウェアとブート用ソフトウェア
- ネットワークに接続されたシステムがカスタム JumpStart™ インストールを行うのに必要な JumpStart ディレクトリ

スタンドアロンシステム

「ネットワークに接続されたスタンドアロンシステム」は、ネットワーク上の他のシステムと情報を共有できますが、ネットワークから切り離されても機能できます。

スタンドアロンシステムは、ルート (/)、/usr、/export/home の各ファイルシステムとスワップ空間を含むハードディスクを自ら持つため、独立して動作できます。つまり、スタンドアロンシステムは、オペレーティングシステムのソフトウェア、実行可能ファイル、仮想メモリー空間、ユーザーが作成したファイルにローカルにアクセスできます。

注-スタンドアロンシステムに必要なファイルシステムを保持するには、十分なディスク領域が必要です。

「ネットワークに接続されないスタンドアロンシステム」は、ネットワークに接続されていない点を除き、ネットワークに接続されたスタンドアロンシステムと同じです。

ディスクレスクライアント

「ディスクレスクライアント」とは、ディスクが搭載されておらず、必要なすべてのソフトウェアおよび記憶装置をサーバーに依存しているシステムのことです。ディスクレスクライアントには、サーバーからリモートで、ルート (/)、/usr、および /home ファイルシステムがマウントされます。

ディスクレスクライアントでは、ネットワークを介してオペレーティングシステムソフトウェアおよび仮想メモリー領域に継続的にアクセスする必要があるため、かなりのネットワークトラフィックが発生します。ディスクレスクライアントは、ネットワークから切り離されたり、そのサーバーが正しく機能しない場合は機能できません。

ディスクレスクライアントの概要については、128 ページの「ディスクレスクライアント管理の概要」を参照してください。

AutoClient システム

AutoClient システムは、インストールおよび管理方法の点では、ディスクレスクライアントとほとんど同じです。AutoClient システムには、次の特徴があります。

- サーバーから個々のルート (/) ファイルシステムおよび /usr ファイルシステムをスワップおよびキャッシュするため最低 100M バイトのローカルディスクが必要
- サーバーが使用できない場合、サーバーのキャッシュにアクセスできるように設定可能
- ほかのファイルシステムおよびソフトウェアアプリケーションへのアクセスは、サーバーに依存
- 永続的なデータがない、FRU (現場交換可能ユニット) として使用

アプライアンス

アプライアンス (たとえば、Sun Ray アプライアンス) は、管理を必要としない X ディスプレイデバイスです。このデバイスには、CPU、ファン、ディスクがなく、メモリーもわずかししか搭載されていません。アプライアンスは Sun ディスプレイモニターに接続されていますが、アプライアンスユーザーのデスクトップセッションは、サーバーで実行され、その結果がユーザーのモニタに表示されます。ユーザーの X 環境は自動的に設定されます。この環境には次の特徴があります。

- ほかのファイルシステムおよびソフトウェアアプリケーションへのアクセスは、サーバーに依存
- ソフトウェアの一元管理およびリソース共有機能を提供
- 永続的なデータがない、FRU (現場交換可能ユニット) として使用

システムタイプ選択のガイドライン

次の特徴に基づいてそれぞれのシステムタイプを比較することにより、使用中の環境にどのシステムタイプが適切かを判断することができます。

- 一元管理
 - システムを FRU (現場交換可能ユニット) として扱えるか。これは、時間がかかるバックアップや復旧操作を必要とせずに、またシステムデータを失わずに、障害が発生したシステムを直ちに新しいシステムと交換できることを意味する
 - システムをバックアップする必要があるか。数多くのデスクトップシステムのバックアップを実行するには、時間とリソースの点で多大の費用コストがかかる場合がある
 - システムのデータは、中央サーバーから変更できるか
 - クライアントシステムのハードウェアを操作せず、短時間で簡単に、システムを中央サーバーからインストールできるか
- 性能
 - この構成は、デスクトップで使用しても性能が低下しないか
 - ネットワークにシステムを追加すると、既存のネットワーク上のシステムの性能に影響を与えるか
- ディスク使用率
 - この構成を効果的に導入するには、どれくらいのディスク容量が必要か

次の表では、各システムタイプの順位をカテゴリ別に表示しています。1 は、最も効果があることを意味します。4 は、最も効果が低いことを意味します。

表 7-2 システムタイプの比較

システムタイプ	一元管理	性能	ディスク使用率
スタンドアロンシステム	4	1	4
ディスクレスクライアント	1	4	1
AutoClient システム	1	2	2
アプライアンス	1	1	1

ディスクレスクライアント管理の概要

次の節および第 8 章では、Solaris 9 リリースでのディスクレスクライアントサポートの管理方法について説明します。

「ディスクレスクライアント」とは、オペレーティングシステム、ソフトウェア、および記憶装置を「OS サーバー」に依存しているシステムのことです。ディスクレスクライアントは、そのルート (/)、/usr、およびその他のファイルシステムを OS サーバーからマウントします。ディスクレスクライアントは独自の CPU と物理メモリを持っており、データをローカルで処理することができます。しかしディスクレスクライアントは、ネットワークから切り離されたり、その OS サーバーが正しく機能しない場合は機能できません。ディスクレスクライアントは、ネットワークを経由して継続的に機能する必要があるため、多大なネットワークトラフィックを発生させます。

以前の Solaris リリースでは、Solstice グラフィカル管理ツールでディスクレスクライアントが管理されました。Solaris 9 リリースでは、ディスクレスクライアントの `smbsservice` コマンドおよび `smdiskless` コマンドを使って、OS サービスおよびディスクレスクライアントサポートを管理できるようになりました。

OS サーバーおよびディスクレスクライアントのサポート情報

次の表に smosservice コマンドおよび smdiskless コマンドをサポートする Solaris リリースおよびアーキテクチャタイプを示します。

アーキテクチャタイプ	Solaris 2.6	Solaris 7	Solaris 8 1/01、4/01、7/01、10/01、2/02	Solaris 9
SPARC サーバー	○	○	○	○
x86 サーバー	○	○	○	○
SPARC クライアント	○	○	○	○
x86 クライアント	X	X	X	○

次の表に smosservice コマンドおよび smdiskless コマンドがサポートする OS サーバーとクライアントの組み合わせを示します。

	Solaris 2.6 リリースサポート	Solaris 7 リリースサポート	Solaris 8 1/01、4/01、7/01、10/01、2/02 サポート	Solaris 9 サポート
OS サーバー / クライアント OS リリース	Solaris 2.6-Solaris 2.6	Solaris 7-Solaris 2.6、または 7	Solaris 8 1/01、4/01、7/01、10/01、2/02-Solaris 2.6、7、または 8 1/01、4/01、7/01、10/01、2/02	Solaris 9-Solaris 2.6、7、8 1/01、4/01、7/01、10/01、2/02

ディスクレスクライアント管理機能

smosservice コマンドおよび smdiskless コマンドを使うと、ネットワークにディスクレスクライアントサポートを追加したり、維持したりすることができます。ネームサービスを使うと、システム情報を一元管理できるので、ホスト名などの重要なシステム情報をネットワーク上のすべてのシステムに複製する必要がありません。

smosservice コマンドおよび smdiskless コマンドを使うと次の作業が実行できます。

- ディスクレスクライアントサポートの追加および変更
- OS サービスの追加および削除
- LDAP、NIS、NIS+、またはファイル環境でのディスクレスクライアント情報の管理

ディスクレスクライアントコマンドは、ディスクレスクライアントの起動の設定にのみ使用できます。このコマンドは、リモートインストールまたはプロファイルサービスなど、ほかのサービスの設定では使用できません。リモートインストールを設定するには、`sysidcfg` ファイルにディスクレスクライアント仕様を定義する必要があります。詳細については、『Solaris 9 12/03 インストールガイド』を参照してください。

ディスクレスクライアントコマンドの使用

次の表のコマンドを使って独自のシェルスクリプトを記述すると、簡単にディスクレスクライアント環境を設定および管理できます。

表 7-3 ディスクレスクライアントコマンド

コマンド	サブコマンド	作業
<code>/usr/sadm/bin/smosservice</code>	<code>add</code>	OS サービスを追加する
	<code>delete</code>	OS サービスを削除する
	<code>list</code>	OS サービスをリスト表示する
	<code>patch</code>	OS サービスのパッチを管理する
<code>/usr/sadm/bin/smdiskless</code>	<code>add</code>	ディスクレスクライアントを OS サーバーに追加する
	<code>delete</code>	ディスクレスクライアントを OS サーバーから削除する
	<code>list</code>	OS サーバー上のディスクレスクライアントをリスト表示する
	<code>modify</code>	ディスクレスクライアントの属性を変更する

次に示す 2 種類の方法で、これらのコマンドに関するヘルプを参照することができます。

- コマンド、サブコマンド、および必要なオプションを入力するときに、`-h` オプションを使用する。たとえば、`smdiskless add` の使用方法を表示するには、次のように入力する。

```
% /usr/sadm/bin/smdiskless add -p my-password -u my-user-name -- -h
```

- `smdiskless(1M)` または `smosservice(1M)` のマニュアルページを参照する。

ディスクレスクライアント管理に必要な RBAC 権限

`smosservice` コマンド および `smdiskless` コマンドはスーパーユーザーとして使用できます。役割によるアクセス制御 (RBAC) を使用している場合、割り当てられた RBAC 権限に応じて、すべてのディスクレスクライアントコマンドまたはそのサブセットのいずれかを使用できます。次の表にディスクレスクライアントコマンドを使用するのに必要な RBAC 権限を示します。

表 7-4 ディスクレスクライアントの管理に必要な権限

RBAC 権限	コマンド	作業
基本的な Solaris ユーザー、 ネットワーク管理	<code>smosservice list</code>	OS サービスをリスト表示する
	<code>smosservice patch</code>	OS サービスパッチをリスト表示する
	<code>smdiskless list</code>	ディスクレスクライアントをリスト表示する
ネットワーク管理	<code>smdiskless add</code>	ディスクレスクライアントを追加する
システム管理者	すべてのコマンド	すべての作業

OS サービスの追加

Solaris OS サーバーとは、ディスクレスクライアントシステムをサポートするオペレーティングシステム (OS) サービスを提供するサーバーです。OS サーバーをサポートするか、`smosservice` コマンドを使用してスタンドアロンシステムを OS サーバーに変換することができます。

サポートする各プラットフォームグループおよび Solaris リリース用に、特定の OS サービスを OS サーバーに追加する必要があります。たとえば、Solaris 8 リリース上で実行中の SPARC® Sun4m システムをサポートする場合には、Sun4m/Solaris 8 OS サービスを OS サーバーに追加する必要があります。また、それぞれのシステムが属するプラットフォームグループが異なるため、Solaris 8 のリリースを実行する SPARC Sun4c システムまたは x86 システムをサポートする OS サービスも追加する必要があります。

OS サービスを追加するには、適切な Solaris CD またはディスクイメージへのアクセス権が必要です。

OS サーバーにパッチが適用された場合に OS サービスを追加する

OS サーバーに OS サービスを追加しようとする、サーバー上の OS と追加するサービスの OS のバージョンが一致しないというエラーメッセージが表示される場合があります。このメッセージは、OS サーバーにインストールされている OS のパッケージにパッチが適用されていても、追加しようとしている OS サービスのパッケージにはパッチが適用されていない (パッチがパッケージに組み込まれているため) 場合に表示されます。

たとえば、Solaris 7 リリースを実行するサーバーがあるとして、また、このサーバーには、パッチが適用された Solaris 2.6 SPARC sun4m OS サービスなど、ほかの OS サービスも追加されているとして、CD-ROM から Solaris 2.6 SPARC sun4c OS サービスをこのサーバーに追加しようとする、次のようなエラーメッセージが表示される場合があります。

```
Error: inconsistent revision, installed package appears to have been
patched resulting in it being different than the package on your media.
You will need to backout all patches that patch this package before
retrying the add OS service option.
```

OS サーバーに必要なディスク容量

ディスクレスクライアント環境を設定する前に、各ディスクレスクライアントディレクトリに必要なディスク容量があるかどうか確認する必要があります。

以前のバージョンの Solaris リリースでは、インストール中にディスクレスクライアントサポートについてのプロンプトが表示されました。Solaris 9 リリースでは、手動でインストール中に /export ファイルシステムを割り当てるか、またはインストール後に作成してください。具体的なディスク容量要件については、次の表を参照してください。

表 7-5 OS サーバーに必要なディスク容量

ディレクトリ	必要な容量 (M バイト)
/export/Solaris_version	10
/export/exec	800
/export/share	5
/export/swap/diskless_client	32 (デフォルトのサイズ)
/export/dump/diskless_client	32 (デフォルトのサイズ)
/export/root/templates/Solaris_version	30
/export/root/clone/Solaris_version/ machine_class	30 - 60 (マシンクラスによって異なる)

表 7-5 OS サーバーに必要なディスク容量 (続き)

ディレクトリ	必要な容量 (M バイト)
/export/root/diskless_client (上記のクローン)	30 - 60 (マシンクラスによって異なる)
/tftpboot/inetboot. <i>machine_class</i> .Solaris_ <i>version</i>	<i>machine_class.Solaris_version</i> ごとに 200 K バイト

第 8 章

ディスクレスクライアントの管理 (手順)

ここでは、Solaris 環境でディスクレスクライアントを管理する方法について説明します。

ディスクレスクライアントの管理手順については、135 ページの「ディスクレスクライアントの管理 (作業マップ)」を参照してください。

ディスクレスクライアント管理の概要については、第 7 章を参照してください。

Solstice AdminSuite™ ソフトウェアを使ってクライアントを管理する方法については、『Solstice AdminSuite 2.3 管理者ガイド』を参照してください。

ディスクレスクライアントの管理 (作業マップ)

次の表にディスクレスクライアントの管理に必要な手順を示します。

作業	説明	参照先
1. (省略可能) 既存のディスクレスクライアントサポートを削除する	Solstice AdminSuite 製品と一緒に追加されたディスクレスクライアントが存在する場合、Solaris リリースをインストールする前に admhostdel コマンドおよび admhostmod コマンドを使って、ディスクレスクライアントサポートおよび OS サービスを削除する	『Solstice AdminSuite 2.3 管理者ガイド』

作業	説明	参照先
2. (省略可能) Solaris 管理コンソールログインを有効にしてディスクレスクライアントのエラーメッセージを表示する	コンソールのメインウィンドウからログビューアを選択し、ディスクレスクライアントのエラーメッセージを表示する	56 ページの「Solaris 管理コンソールを起動する」
3. ディスクレスクライアントを追加する準備をする	サポートされているリリースを確認し、各ディスクレスクライアントの「プラットフォーム」、「メディアパス」、および「クラスタ」(またはソフトウェアグループ)を識別する	138 ページの「ディスクレスクライアントの追加の準備」
4. 必要な OS サービスを OS サーバーに追加する	smoservice コマンドを使用してサポートしたいディスクレスクライアントの OS サービスを追加する。サポートする各ディスクレスクライアントプラットフォームのプラットフォームおよびメディアパスを識別する必要がある	140 ページの「ディスクレスクライアントサポートの OS サービスの追加方法」
5. ディスクレスクライアントを追加する	smdiskless コマンドを使用してすべての必須情報を指定し、ディスクレスクライアントサポートを追加する	141 ページの「ディスクレスクライアントの追加方法」
6. ディスクレスクライアントを起動する	ディスクレスクライアントを起動して、ディスクレスクライアントサポートが正常に追加されたことを確認する	143 ページの「ディスクレスクライアントの起動方法」
7. (省略可能) ディスクレスクライアントサポートを削除する	必要がなくなった場合には、ディスクレスクライアントのサポートを削除する	143 ページの「ディスクレスクライアントサポートの削除方法」
8. (省略可能) ディスクレスクライアントの OS サービスを削除する	必要がなくなった場合には、ディスクレスクライアントの OS サービスを削除する	144 ページの「ディスクレスクライアントの OS サービスを削除する方法」
9. (省略可能) OS サービスにパッチを適用する	ディスクレスクライアント OS サービスのパッチを追加、削除、リスト表示、または同期化する	146 ページの「ディスクレスクライアントの OS パッチの追加方法」

ディスクレスクライアントの管理

ここでは、ディスクレスクライアントの管理に必要な手順について説明します。

ディスクレスクライアントを管理する場合には、次の点に注意してください。

- Solaris インストールプログラムでは、ディスクレスクライアントサポートのセットアップを促すメッセージが表示されません。ディスクレスクライアントをサポートするには、手動で `/export` パーティションを作成してください。インストール中またはインストール後に `/export` パーティションを作成してください。
- `/export` パーティションには、サポートするクライアントの数に応じて、最低 800 – 1000 MB の容量が必要です。詳細については、132 ページの「OS サーバーに必要なディスク容量」を参照してください。
- `smosservice` コマンドまたは `smdiskless` コマンドで指定したネームサービスは、`/etc/nsswitch.conf` ファイルで指定されたプライマリネームサービスと一致していなければなりません。`smdiskless` コマンドまたは `smosservice` コマンドにネームサービスを指定しない場合、デフォルトのネームサービスは `files` になります。

各ディスクレスクライアントについてプラットフォーム、メディアパス、およびクラスタを決定したら、OS サービスを追加する準備ができたこととなります。追加する各 OS サービスについて、次のディレクトリが作成され移植されます。

- `/export/Solaris_version/Solaris_version_instruction_set.all`
(`/export/exec/Solaris_version/Solaris_version_instruction_set.all` へのシンボリックリンク)
- `/export/Solaris_version`
- `/export/Solaris_version/var`
- `/export/Solaris_version/opt`
- `/export/share`
- `/export/root/templates/Solaris_version`
- `/export/root/clone`
- `/export/root/clone/Solaris_version`
- `/export/root/clone/Solaris_version /machine_class`

追加する各ディスクレスクライアントについて、次のデフォルトのディレクトリが OS サーバー上に作成されます。

- `/export/root/diskless_client`
- `/export/swap/diskless_client`
- `/tftpboot/diskless_client_ipaddress_in_hex /export/dump/diskless_client (-x dump オプションを指定した場合)`

注 `--x` オプションを使うと、ルート、`/swap`、および `/dump` ディレクトリのデフォルト位置を変更することができます。ただし、`/export` ファイルシステムの下にはこれらのディレクトリを作成しないでください。

▼ ディスクレスクライアントの追加の準備

サポートしているリリースが、OS サービスに指定されたシステム上で実行されていることを確認します。さらに、OS サーバリリースとディスクレスクライアントリリースの組み合わせがサポートされていることも確認します。

`smosservice add` コマンドを使用して OS サービスを追加する場合は、サポートしたい各ディスクレスクライアントプラットフォームの「プラットフォーム」、「メディアパス」、および「クラスタ」(またはソフトウェアグループ)を指定する必要があります。

1. サポートされる **Solaris** リリースの組み合わせが、対象の **OS** サーバおよびディスクレスクライアントで実行されているかどうか確認します。

詳細については、129 ページの「OS サーバおよびディスクレスクライアントのサポート情報」を参照してください。

2. 次の形式で使用されるディスクレスクライアントプラットフォームを識別します。

instruction_set.machine_class.Solaris_version

たとえば、次のようになります。

`sparc.sun4u.Solaris_9`

以下に、利用できるプラットフォームのオプションを示します。

<i>instruction_set</i>	<i>machine_class</i>	<i>Solaris_version</i>
sparc	sun4d*, sun4c*, sun4m*, および sun4u	Solaris_9, Solaris_8, Solaris_2.7, Solaris_2.6
i386	i86pc	Solaris_9, Solaris_8, Solaris_2, Solaris_2.6

* `sun4c` アーキテクチャは、Solaris 8 または Solaris 9 リリースではサポートされていません。`sun4d` アーキテクチャは、Solaris 9 リリースではサポートされていません。

3. メディアパスを特定します。これは、ディスクレスクライアントにインストールしたいオペレーティングシステムを含むディスクイメージのフルパスです。

Solaris オペレーティング環境は、複数の CD で配布されます。ただし、`smosservice` コマンドでは、複数の CD から OS を読み込むことはできません。ユーザーは、次の操作を行うために Solaris Software CD (および付属する

Languages CD) にあるスクリプトを実行する必要があります。

- サーバー上にインストールイメージを作成します。インストールサーバーのセットアップについての情報は、『Solaris 9 12/03 インストールガイド』を参照してください。
- 次のいずれかのスクリプトを使用して、必要な OS サービスを CD イメージから読み込みます。
 - Solaris 9 Software 1 of 2 CD 用 –
/cdrom/cdrom0/s0/Solaris_9/Tools/setup_install_server
 - Solaris 9 Software 2 of 2 CD 用 –
/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server
 - Solaris 9 Languages CD 用 –
/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server

たとえば、ローカルに接続された CD-ROM デバイス上の Solaris 9 Software 1 of 2 CD (SPARC Platform Edition) から setup_install_server スクリプトを使用する場合は、構文が次のようになります。

```
# mkdir /export/install/sparc_9
# cd /cd_mount_point/Solaris_9/Tools
# ./setup_install_server /export/install/sparc_9
```

- ディスクに Solaris CD イメージをインストールした後、ディスクイメージのパスを指定してください。たとえば、次のようになります。

```
/net/export/install/sparc_9
```

4. OS サービスを追加するときに SUNWCxall クラスタを特定します。

同じシステム (SPARC または x86) 上で同じオペレーティング環境を実行するディスクレスクライアントには、「同じクラスタ」を使用しなければなりません。たとえば、次のディスクレスクライアントについて考えてみましょう。

- sparc.sun4m.Solaris_9
- sparc.sun4u.Solaris_9

sun4u および sun4m システムでは SUNWCxall クラスタを要求するため、これらのディスクレスクライアントをセットアップするには、各ディスクレスクライアントに SUNWCxall クラスタを指定する必要があります。また同じシステムで同じオペレーティング環境 (ここでは Solaris_9) を実行するディスクレスクライアントでは同じクラスタを使用する必要があります。

注 – sun4u システム、または 8 ビットの高速カラーメモリーフレームバッファ (cgsix) が搭載されたシステムを使用している場合には、クラスタに必ず SUNWCxall を指定してください。

▼ ディスクレスクライアントサポートの OS サービスの追加方法

ディスクレスクライアントの OS サービスをサーバーに追加するには、次の手順を実行します。

1. サーバーでスーパーユーザーになるか、同等の役割を引き受けます。
詳細については、50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」を参照してください。
2. **Solaris** 管理コンソールサーバーが実行中であり、システムでディスクレスクライアントツールが使用できることを確認します。

```
# /usr/sadm/bin/smosservice list -H starbug:898 --
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from
starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
Platform
-----
```

3. OS サービスを追加します。

```
# /usr/sadm/bin/smosservice add -H hostname:898 -- -o hostname
-x mediapath=path -x platform=instruction-set.machine-class.Solaris-version
-x cluster=cluster-name -x locale=locale-name
```

add	指定された OS サービスを追加する
-H hostname:898	接続するホスト名とポートを指定する。 ポートを指定しない場合は、デフォルト ポート 898 に接続される
--	これ以降のサブコマンド引数が始まるこ とを示す
-x mediapath=path	Solaris イメージのフルパスを指定する
-x platform=instruction-set.machine-class.Solaris- version	追加する命令アーキテクチャ、マシク ラス、および Solaris バージョンを指定 する
-x cluster=cluster-name	インストールする Solaris クラスタを指 定する
-x locale=locale-name	インストールするロケールを指定する

注 - サーバーのスピードおよび選択した OS サービスの構成により、インストールプロセスには 45 分ほどかかることがあります。

詳細については、smossservice (1M) のマニュアルページを参照してください。

4. (省略可能) ほかの OS サービスを追加します。
5. OS サービスを追加し終わったら、OS サービスがインストールされているかどうか確認します。

```
# /usr/sadm/bin/smossservice list -H hostname:898 --
```

例 — ディスクレスクライアントサポートの OS サービスを追加する

この例では、サーバー starbug に Solaris 8 OS サービスを追加する方法を示します。サーバー starbug では Solaris 9 リリースを実行しています。

```
# /usr/sadm/bin/smossservice add -H starbug:898 -- -o starbug
-x mediapath=/net/install/export/sparc_8 -x platform=sparc.sun4u.Solaris_8
-x cluster=SUNWCXall -x locale=en_US
Authenticating as user: root
```

```
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password :: xxx
Loading Tool: com.sun.admin.ossservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.ossservermgr.cli.OsServerMgrCli from starbug:898
was successful.
```

▼ ディスクレスクライアントの追加方法

OS サービスを追加した後に、ディスクレスクライアントを追加するには、次の手順を実行します。

1. スーパーユーザーになるか、同等の役割を引き受けます。
詳細については、50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」を参照してください。
2. ディスクレスクライアントを追加します。

```
# /usr/sadm/bin/smdiskless add -- -i ip-address -e ethernet-address
-n client-name -x os=instruction-set.machine-class.Solaris-version
-x root=/export/root/client-name -x swap=/export/swap/client-name
-x swapsize=size -x tz=timezone -x locale=locale-name
```

<code>add</code>	指定したディスクレスクライアントを追加する
<code>--</code>	これ以降、サブコマンド引数が始まることを示す
<code>-i ip-address</code>	ディスクレスクライアントの IP アドレスを指定する
<code>-e ethernet-address</code>	ディスクレスクライアントの Ethernet アドレスを識別する
<code>-n client-name</code>	ディスクレスクライアントの名前を指定する
<code>-x os= instruction-set.machine-class.Solaris-version</code>	ディスクレスクライアントの命令アーキテクチャ、マシンクラス、OS、および Solaris バージョンを指定する
<code>-x root=root=/export/root/client-name</code>	ディスクレスクライアントのルートディレクトリを指定する
<code>-x swap=root=/export/root/client-name</code>	ディスクレスクライアントのスワップファイルを指定する
<code>-x swapsize=size</code>	スワップファイルのサイズをメガバイト (MB) で指定する。デフォルトは 24 (MB)
<code>-x tz=timezone</code>	ディスクレスクライアントの時間帯を指定する
<code>-x locale= locale-name</code>	ディスクレスクライアントをインストールするロケールを指定する

詳細については、`smdiskless (1M)` のマニュアルページを参照してください。

3. (省略可能) `smdiskless add` コマンドを続けて使用して、各ディスクレスクライアントを追加します。
4. ディスクレスクライアントがインストールされたことを確認します。

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

例 — ディスクレスクライアントを追加する

この例では、サーバー `starbug` から Solaris 8 クライアント `holoship` を追加する方法を示します。

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.103 -e 8:0:20:92:4e:f3
-n holoship -x os=sparc.sun4u.Solaris_8 -x root=/export/root/holoship
-x swap=/export/swap/holoship -x swapsize=128 -x tz=US/Mountain
-x locale=en_US
```

この例では、サーバー starbug から Solaris 7 クライアント inquisitor を追加する方法を示します。

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.102 -e 8:0:20:1f:31:be
-n inquisitor -x os=sparc.sun4u.Solaris_2.7 -x root=/export/root/inquisitor
-x swap=/export/swap/inquisitor -x swapsize=64 -x tz=US/Mountain
```

▼ ディスクレスクライアントの起動方法

1. OS サーバーの次の必要条件を確認します。

- ディスクレスクライアントおよび OS サービスの追加に使用するネームサービスがサーバーの /etc/nsswitch.conf ファイルのプライマリ名に一致していることを確認します。

一致しない場合は、ディスクレスクライアントが起動しません。

- rpc.bootparamd デモンが実行中かどうか確認します。実行していない場合には、それを起動してください。

2. ディスクレスクライアントを起動します。

```
ok boot net
```

▼ ディスクレスクライアントサポートの削除方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

詳細については、50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」を参照してください。

2. ディスクレスクライアントサポートを削除します。

```
# /usr/sadm/bin/smdiskless delete -- -o hostname:898 -n client-name
```

3. ディスクレスクライアントサポートが削除されたことを確認します。

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

例 — ディスクレスクライアントサポートを削除する

この例では、OS サーバー starbug からディスクレスクライアント holoship を削除する方法を示します。

```
# /usr/sadm/bin/smdiskless delete -- -o starbug -n holoship
Authenticating as user: root
```

```
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::
```

```
Starting SMC server version 2.0.0.
endpoint created: :898
SMC server is ready.
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug
was successful.
```

▼ ディスクレスクライアントの OS サービスを削除する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
詳細については、50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」を参照してください。

2. ディスクレスクライアントの OS サービスを削除します。

```
# /usr/sadm/bin/smosservice delete -H hostname:898 --
-x rmpatform=instruction-set.machine-class.Solaris-version
```

3. OS サービスが削除されたことを確認します。

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

例 — ディスクレスクライアントの OS サービスを削除する

次の例では、サーバー starbug からディスクレスクライアント OS サービス (sparc.all.Solaris_9) を削除する方法を示します。

```
# /usr/sadm/bin/smosservice delete -H starbug:898 --
-x rmpatform=sparc.all.Solaris_9
Authenticating as user: root
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug:898
was successful.
```

ディスクレスクライアント OS サービスにパッチを適用する

smossservice patch コマンドを使うと次の操作を実行できます。

- OS サーバー上に /export/diskless/Patches パッチスプールディレクトリを設定する
- パッチスプールディレクトリにパッチを追加する。追加するパッチにより、スプールの既存のパッチが不要になると、不要なパッチは /export/diskless/Patches/Archive に移動される
- パッチスプールディレクトリからパッチを削除する
- パッチスプールディレクトリ内のパッチをリスト表示する
- スプールされたパッチをクライアントに同期させる。クライアントが更新されたパッチを認識できるように、同期させた各クライアントをリブートする必要がある

注 - 推奨 OS パッチを定期的にインストールして、OS サーバーを最新の状態に保ってください。

パッチのダウンロード方法については、370 ページの「署名のない Solaris パッチのダウンロード方法」を参照してください。

ディスクレスクライアントの OS パッチの表示

ディスクレスクライアントのパッチは、パッチのタイプに応じて、異なるディレクトリにロギングされます。

- カーネルパッチは、ディスクレスクライアントの /var/sadm/patch ディレクトリにロギングされます。カーネルパッチを表示するには、ディスクレスクライアントで次のコマンドを入力します。

```
% patchadd -p
```

- /usr パッチは、OS サーバーの /export/Solaris_version/var/patch ディレクトリにロギングされます。パッチ ID ごとにディレクトリが作成されます。/usr パッチを表示するには、OS サーバーで次のコマンドを入力します。

```
% patchadd -S Solaris_8 -p
```

```
Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
```

OS およびアーキテクチャによりスプールされたすべてのパッチを表示するには、-p オプションを指定した smossservice コマンドを使用します。

▼ ディスクレスクライアントの OS パッチの追加方法

1. サーバーでスーパーユーザーになるか、同等の役割を引き受けます。
詳細については、50 ページの「スーパーユーザー (ルート) になるか役割を引き受ける方法」を参照してください。

2. ディスクレスクライアントシステムにログインし、シャットダウンします。

```
# init 0
```

3. パッチをスプールディレクトリに追加します。

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/patch-ID-revision  
Authenticating as user: root
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]  
Please enter a string value for: password ::  
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug  
Login to starbug as user root was successful.  
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug  
was successful.
```

追加するパッチが別のパッチに依存する場合は、次のメッセージが表示され、パッチを追加できません。

```
The patch patch-ID-revision could not be added  
because it is dependent on other patches which have not yet been spooled.  
You must add all required patches to the spool first.
```

4. パッチがスプールされたことを確認します。

```
# /usr/sadm/bin/smosservice patch -- -P
```

5. スプールされたパッチをディスクレスクライアントにプッシュします。

```
# /usr/sadm/bin/smosservice patch -- -m -U  
Authenticating as user: root
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]  
Please enter a string value for: password ::  
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug  
Login to starbug as user root was successful.  
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug  
was successful.
```

注 - パッチをディスクレスクライアントにプッシュおよび同期させるには、1 パッチあたり最大 90 分ほどかかることがあります。

6. パッチがディスクレスクライアントに適用されたことを確認します。

```
# /usr/sadm/bin/smosservice patch -- -P  
Authenticating as user: root
```

```

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starbug
was successful.
Patches In Spool Area
Os Rel Arch   Patch Id   Synopsis
-----
8           sparc  111879-
01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                                Patch
-----
Solaris_8

Patches Applied To Clone Areas
Clone Area                                Patch
-----
Solaris_8/sun4u

```

例 — ディスクレスクライアントの OS パッチを追加する

この例では、Solaris 8 パッチ (111879-01) をサーバーのディスクレスクライアントの OS サービスに追加する方法を示します。

```

# /usr/sadm/bin/smosservice patch -- -a /var/patches/111879-01
Authenticating as user: root
.
.
.
# /usr/sadm/bin/smosservice patch -- -P
Patches In Spool Area
Os Rel Arch   Patch Id   Synopsis
-----
8           sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr
.
.
.
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root
.
.
.
# /usr/sadm/bin/smosservice patch -- -P
Authenticating as user: root
.
.
.
Patches In Spool Area
Os Rel Arch   Patch Id   Synopsis
-----

```


ダンプサイズも、ディスクレスクライアントの追加時に `-x dumpsize` オプションを指定したかどうかによって異なる。

- OS サーバーの IP アドレスが `/export/root/client/etc/inet/hosts` ファイルに設定されているかどうかを確認する。

ネームサービス環境

- OS サーバーとクライアントの Ethernet アドレスおよび IP アドレスが正しくマップされていることを確認する。
- `/etc/bootparams` ファイルに、次のようにクライアントのルートおよびスワップ領域へのパスが含まれていることを確認する。

```
client root=os-server:/export/  
root/client swap=os-server:/export/  
swap/client swapsize=24
```

スワップのサイズは、ディスクレスクライアントの追加時に `-x swapsize` オプションを指定したかどうかによって異なる。ディスクレスクライアントの追加時に `-x dump` オプションを指定した場合は、次の行が表示される。

```
dump=os-server:/export/dump/client dumpsize=24
```

ダンプサイズも、ディスクレスクライアントの追加時に `-x dumpsize` オプションを指定したかどうかによって異なる。

問題

ディスクレスクライアントパニック

解決法

- OS サーバーの Ethernet アドレスが IP アドレスに正しくマップされていることを確認する。システムをあるネットワークから別のネットワークに物理的に移動した場合、システムの新しい IP アドレスを再マップするのを忘れていた可能性がある。
- クライアントの RARP、TFTP、または bootparam 要求に応答する「同じサブネット」の別のサーバーのデータベースに、クライアントのホスト名、IP アドレス、および Ethernet アドレスが存在しないことを確認する。インストールサーバーから OS をインストールするために、しばしばテストシステムがセットアップされる。このような場合、インストールサーバーはクライアントの RARP または bootparam 要求に対して、正しくない IP アドレスを返す。この不正なアドレスにより、間違ったアーキテクチャのブートプログラムをダウンロードしたり、クライアントのルートファイルシステムのマウントが失敗したりしている可能性がある。
- ディスクレスクライアントの TFTP 要求にインストールサーバー (または以前の OS サーバー) が応答しないことを確認する。このサーバーは不正なブートプログラムを転送している。ブートプログラムのアーキテクチャが異なる場合は、クライアントがただちにパニックになる。ブートプログラムが非 OS サーバーから読み込まれた場合、クライアントはそのルートパーティションを非 OS サーバー上に確保し、`/usr` パーティションを OS サーバー上に確保することがある。この状況では、ルートおよび `/usr` パーティションのアーキテクチャまたはバージョンが競合する場合、クライアントがパニックに陥る。

- インストールサーバーと OS サーバーの両方を使用している場合は、
/etc/dfs/dfstab ファイルに次のエントリがあることを確認する。

```
share -F nfs -o -ro /export/exec/Solaris_version_instruction_set.all/usr
```

ここで、*version*=2.6、2.7、8、または 9、また *instruction_set*=sparc または i386

- ディスクレスクライアントのルート、/swap パーティション、および /dump (指定されている場合) パーティションに共有エントリがあることを確認する。

```
share -F nfs -o rw=client,root=client /export/root/client
```

```
share -F nfs -o rw=client,root=client /export/swap/client
```

```
share -F nfs -o rw=client,root=client /export/dump/client
```

- OS サーバーで次のコマンドを入力し、共有されているファイルを確認する。

```
% share
```

OS サーバーでは、ディスクレスクライアントを追加したときに指定した、
/export/root/client および/export/swap/client_name (デフォルト)、または
ルート、/swap パーティション、および /dump パーティションを共有する必要がある。

次のエントリが /etc/dfs/dfstab ファイルにあるかを確認する。

```
share -F nfs -o ro /export/exec/Solaris_version_instruction_set.all/usr
```

```
share -F nfs -o rw=client,root=client /export/root/client
```

```
share -F nfs -o rw=client,root=client /export/swap/client
```

問題

OS サーバーがディスクレスクライアントの RARP 要求に応答しない

解決法

クライアントの目的の OS サーバーから、次のクライアントの Ethernet アドレスを使って snoop コマンドを root として実行する。

```
# snoop xx:xx:xx:xx:xx:xx
```

問題

ブートプログラムをダウンロードしたが、プロセスの初期にパニックが発生した

解決法

snoop コマンドを使用して、対象の OS サーバーがクライアントの TFTP および NFS 要求に応答するかどうかを確認する。

問題

- ディスクレスクライアントのハングアップ
- ディスクレスクライアントの RARP 要求に対するサーバーの応答が不正である

解決法

OS サーバーで次のデーモンを再起動する。

```
# /usr/sbin/rpc.bootparamd
```

```
# /usr/sbin/in.rarpd -a
```

第 9 章

システムのシャットダウンとブート

以下の各章で、システムのシャットダウンとブートについて説明します。

第 10 章	システムのシャットダウンとブートの概要とガイドラインを示します。
第 11 章	実行レベルとブートファイルに関する概要と作業手順について説明します。
第 12 章	システムのシャットダウン手順について説明します。
第 13 章	SPARC システムのブート手順について説明します。
第 14 章	x86 システムのブート手順について説明します。
第 15 章	SPARC システムと x86 システムのブートプロセスの概要を説明します。

第 10 章

システムのシャットダウンとブート (概要)

この章では、システムのシャットダウンとブートについて概要を説明します。Solaris オペレーティングシステムは、電子メールとネットワークリソースをいつでも利用できるように、停止することなく動作するように設計されています。しかし、システム構成の変更、定期保守、停電などの理由で、システムをシャットダウンまたはリポートしなければならない場合があります。

この章の内容は次のとおりです。

- 153 ページの「システムのシャットダウンとブートに関する新機能」
- 155 ページの「シャットダウンとブートについての参照先」
- 155 ページの「シャットダウンとブートの用語」
- 156 ページの「システムのシャットダウンに関するガイドライン」
- 156 ページの「システムのブートに関するガイドライン」
- 157 ページの「ネットワークからのシステムのブート」
- 157 ページの「システムをシャットダウンする場合」
- 158 ページの「システムをブートする場合」

システムのシャットダウンとブートに関する新機能

この節では、Solaris 9 におけるシステムのシャットダウンとブートに関する新機能について説明します。

PXE ネットワークブート

PXE (Preboot Execution Environment) ネットワーク起動プロトコルをサポートしている x86 システムでは、Solaris ブートフロッピーディスクを使用しなくても、ネットワークから直接 Solaris x86 プラットフォーム版をブートできます。PXE ネットワークブートは、Intel の PXE 仕様を実装しているデバイスでのみ動作します。

システム BIOS またはネットワークアダプタ BIOS のどちらか一方、またはその両方の BIOS 設定プログラムを使うことによって、クライアントシステム上で PXE ネットワークブートを使用できるようにします。いくつかのシステムでは、他のデバイスからのブートよりも先にネットワークブートが実行されるように、ブートデバイスの優先順位を調整する必要があります。各設定プログラムに関しては、製造業者のマニュアルを参照するか、またはブート中に表示される設定プログラムの指示を参照してください。

PXE 対応ネットワークアダプタの中には、ブート時にしばらく表示されるプロンプトに対して特定のキーを押すと、PXE ブートを実行する機能を持つものがあります。この機能は、PXE の設定を変更する必要がないので、通常はディスクドライブからブートを実行するシステムのインストールブートにおいて PXE を使用する場合に適しています。アダプタにこの機能がない場合は、システムのインストール後に行われるリブート時に、BIOS の設定で PXE を使用しないように設定してください。システムがディスクドライブからブートするようになります。

一部の初期バージョンの PXE ファームウェアでは、Solaris システムをブートすることができません。このようなファームウェアを使用する場合、システムがブートサーバーから PXE ネットワークブートストラッププログラムを読み込むことはできませんが、ブートストラップはパケットを転送しません。この問題が発生した場合は、アダプタの PXE ファームウェアをアップグレードしてください。ファームウェアのアップグレードに関する情報は、アダプタの製造業者の Web サイトから入手してください。詳細については、e1x1 (7D) および iprb (7D) のマニュアルページを参照してください。

ブートフロッピーディスクを使用するか使用しないかに関わらず、x86 システムのブートについては、212 ページの「x86: システムをネットワークからブートする方法」を参照してください。

シャットダウンとブートについての参照先

システムをシャットダウンおよびブートする手順については、次を参照してください。

シャットダウンとブート作業	参照先
SPARC システムまたは x86 システムのシャットダウン	第 12 章
SPARC システムのブート	第 13 章
x86 システムのブート	第 14 章
電源管理ソフトウェアによる SPARC システムの管理	power.conf (4) および pmconfig (1M) の各マニュアルページ

シャットダウンとブートの用語

この節では、シャットダウンとブートに関する用語について説明します。

- **実行レベルと init 状態** – 「実行レベル」とは、システムの状態を表す文字または数字のことで、どのシステムサービスを使用できるのかを示します。システムは常に定義済み実行レベルの 1 つで動作します。実行レベルを変更するために init プロセスが使用されるため、実行レベルは「init 状態」と呼ばれることもあります。システム管理者は、init (1M) コマンドを使用して、実行レベルを変更します。このマニュアルでは、init 状態を実行レベルと呼びます。
実行レベルの詳細については、161 ページの「実行レベル」を参照してください。
- **ブートタイプ** – 「ブートタイプ」とはシステムのブート方法を表します。次のようなブートタイプがあります。
 - **対話式ブート** – システムのブート方法に関する情報 (カーネルやデバイスのパス名など) を入力するプロンプトが表示される
 - **再構成用ブート** – システムが再構成され、新しく追加したハードウェアや新しい擬似デバイスがサポートされる
 - **回復ブート** – システムがハング状態になったとき、無効なエントリがあるためシステムが正常にブートできないとき、またはユーザーがログインできないときに使用する

システムのシャットダウンに関するガイドライン

システムをシャットダウンするときは、次の点に注意してください。

- システムのシャットダウンには、`init` および `shutdown` コマンドを使用します。これらのコマンドは、すべてのシステムプロセスとサービスを正常に終了させてからシャットダウンします。
- サーバーをシャットダウンする場合は、`shutdown` コマンドを使用してください。`shutdown` コマンドは、シャットダウンを実行する前に、サーバーにログインしているユーザーやサーバーのリソースをマウントしているシステムにシャットダウンを通知します。システムのシャットダウンについては、ユーザーが予定を立てられるようあらかじめ電子メールで知らせておくようにします。
- `shutdown` または `init` コマンドを使用してシステムをシャットダウンするには、スーパーユーザー権限が必要です。
- `shutdown` および `init` コマンドはどちらも実行レベルを引数に指定します。最もよく使用される実行レベルは次の3つです。
 - 実行レベル 3 - すべてのシステムリソースを使用でき、ユーザーもログインできる状態。デフォルトでは、システムをブートすると実行レベル 3 になります。通常の運用で使用されます。NFS リソースを共有できるマルチユーザーレベルとも呼ばれます。
 - 実行レベル 6 - オペレーティングシステムを停止して、`/etc/inittab` ファイルの `initdefault` エントリに定義されている状態でリブートします。
 - 実行レベル 0 - オペレーティングシステムがシャットダウンされ、安全に電源切断できる状態。システムの設置場所を変更したり、ハードウェアを追加または削除する場合は、システムを実行レベル 0 にする必要があります。

実行レベルの詳細については、第 11 章を参照してください。

システムのブートに関するガイドライン

システムをブートするときは、次の点に注意してください。

- シャットダウン後にシステムをブートするには、SPARC システムの場合は、PROM レベルで `boot` コマンドを使用します。x86 システムの場合は、一次ブートサブシステムメニューで `boot` コマンドを使用します。
- 電源を切断した後に再投入すればシステムをリブートできます。ただし、この方法ではシステムサービスやプロセスが突然終了してしまうので、適切なシャットダウンとは言えません。緊急時のリブート以外には使用しないようにします。

- SPARC システムと x86 システムとでは、ブート時に使用するハードウェアが異なります。これらのハードウェアの違いについては、第 15 章を参照してください。

ネットワークからのシステムのブート

次のような場合に、システムをネットワークからブートする必要があります。

- システムを最初にインストールする場合
- システムをローカルディスクからブートできない場合
- システムがディスクレスクライアントである場合

さらに、次の 2 つのネットワーク構成ブート方法も利用できます。

- RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol)
- DHCP (Dynamic Host Configuration Protocol)

デフォルトのネットワークブート方法は RARP に設定されています。

ネットワーク経由でシステムをブートする方法については、次の表を参考にしてください。

ネットワークブート作業	参照先
SPARC システムまたは SPARC ディスクレスクライアントのブート	第 13 章
x86 システムまたは x86 ディスクレスクライアントのブート	第 14 章
インストール時の DHCP クライアントのブート	『Solaris 9 12/03 インストールガイド』
DHCP マネージャを備えた DHCP クライアントの構成	『Solaris のシステム管理 (IP サービス)』

システムをシャットダウンする場合

表 10-1 に、システム管理作業とそれに伴って必要となるシャットダウンの種類を示します。

表 10-1 システムのシャットダウン

システムシャットダウンの理由	適切な実行レベル	参照先
停電のためシステムの電源を切断する	実行レベル 0。安全に電源を切れる状態	第 12 章
/etc/system ファイル内のカーネルパラメータを変更する	実行レベル 6 (システムのリブート)	第 12 章
ファイルシステムを保守する (システムデータのバックアップや復元など)	実行レベル S (シングルユーザーレベル)	第 12 章
/etc/system などのシステム構成ファイルを修復する	158 ページの「システムをブートする場合」を参照	なし
システムにハードウェアを追加する (または、システムからハードウェアを削除する)	再構成用ブート (ハードウェアを追加または削除したら電源を切断する)	第 27 章
ブート失敗の原因となっていた重要なシステムファイルを修復する	158 ページの「システムをブートする場合」を参照	なし
カーネルデバッグ (kadb) をブートして、システムの障害を調査する	実行レベル 0 (可能な場合)	第 12 章
ハング状態から回復させ、クラッシュダンプを強制する	158 ページの「システムをブートする場合」を参照	なし

サーバーまたはスタンドアロンシステムのシャットダウンの例については、第 12 章を参照してください。

システムをブートする場合

次の表に、システム管理作業とそれに伴って必要となるブートタイプを示します。

表 10-2 システムのブート

システムリブートの理由	適切なブートタイプ	SPARC のブート手順の参照先	x86 のブート手順の参照先
停電のためシステムの電源を切断する	システムの電源を再投入する	第 12 章	第 12 章

表 10-2 システムのブート (続き)

システムリブートの理由	適切なブートタイプ	SPARC のブート手順の参照先	x86 のブート手順の参照先
/etc/system ファイル内のカーネルパラメータを変更する	システムを実行レベル 3 でリブートする (NFS リソースを共有できるマルチユーザーレベル)	194 ページの「SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」	208 ページの「x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」
ファイルシステムを保守する (システムデータのバックアップや復元など)	実行レベル S で Control + D を押し、システムを実行レベル 3 に戻す	195 ページの「SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法」	209 ページの「x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法」
/etc/system などのシステム構成ファイルを修復する	対話式ブート	196 ページの「SPARC: システムを対話式でブートする方法」	210 ページの「x86: システムを対話式でブートする方法」
システムにハードウェアを追加する (または、システムからハードウェアを削除する)	再構成ブート (ハードウェアを追加または削除したら電源を投入する)	502 ページの「SPARC: 二次ディスクを接続してブートする方法」	513 ページの「x86: 二次ディスクを接続してブートする方法」
カーネルデバッグ (kadb) をブートして、システムの障害を調査する	kabd をブートする	201 ページの「SPARC: カーネルデバッグ (kadb) を使ってシステムをブートする方法」	218 ページの「x86: カーネルデバッグ (kadb) を使ってシステムをブートする方法」
ブート失敗の原因となっていた重要なシステムファイルを修復する	回復ブート	199 ページの「SPARC: 復元を目的としてシステムをブートする方法」	213 ページの「x86: 復元を目的としてシステムをブートする方法」
ハング状態から回復させ、クラッシュダンプを強制する	回復ブート	202 ページの「SPARC: クラッシュダンプを強制してシステムをリブートする方法」の例を参照	219 ページの「x86: クラッシュダンプを強制してシステムをリブートする方法」の例を参照

システムブートの例については、第 13 章または第 14 章を参照してください。

第 11 章

実行レベルとブートファイル (手順)

この章では、実行レベルとブートファイルに関する概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 171 ページの「実行制御スクリプトを使用してサービスを起動または停止する方法」
- 172 ページの「実行制御スクリプトを追加する方法」
- 173 ページの「実行制御スクリプトを無効にする方法」

この章の内容は次のとおりです。

- 161 ページの「実行レベル」
- 163 ページの「/etc/inittab ファイル」
- 166 ページの「実行制御スクリプト」
- 174 ページの「x86: ブートファイル」

実行レベル

システムの「実行レベル」(「init 状態」とも呼ばれる)は、ユーザーが使用できるサービスとリソースを定義します。システムが一度に持つことのできる実行レベルは 1 つだけです。

Solaris 環境には 8 つの実行レベルがあります (次の表を参照)。デフォルトの実行レベル 3 は、/etc/inittab ファイルに指定されています。

表 11-1 Solaris 実行レベル

実行レベル	init 状態	種類	目的
0	電源切断状態	電源切断	オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにする。
s または S	シングルユーザー状態	シングルユーザー	一部のファイルシステムがマウントされ使用可能な状態で、シングルユーザーとして動作する。
1	システム管理状態	シングルユーザー	すべての使用可能なファイルシステムにアクセスする。ユーザーロゲインは使用不可である。
2	マルチユーザー状態	マルチユーザー	通常の運用に使用する。複数のユーザーがシステムとすべてのファイルシステムにアクセスできる。NFS サーバーデーモンを除く、すべてのデーモンが動作する。
3	NFS リソースを共有したマルチユーザーレベル	マルチユーザー	NFS リソースを共有する通常の運用に使用する。Solaris 環境におけるデフォルトの実行レベル。
4	マルチユーザー状態 (予備)		現在は使用できない。
5	電源切断状態	電源切断	オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにする。可能であれば、この機能をサポートしているシステムでは電源を自動的に切断する。
6	リポート状態	リポート	システムをシャットダウンして実行レベル 0 にした後、NFS リソースを共有するマルチユーザーレベル (または、inittab ファイルに指定されたデフォルトのレベル) でリポートする。

システムの実行レベルを確認する方法

who -r コマンドを使用すると、実行レベルに関する情報が表示されます。

```
$ who -r
```

who -r コマンドを使用して、システムの現在の実行レベルを調べます (ただし、実行レベル 0 を除く)。

例 — システムの実行レベルを確認する

次の例では、システムの現在の実行レベルと以前の実行レベルに関する情報を表示します。

```
$ who -r
.      run-level 31  Dec 13 10:102  33  04 55
$
```

1. 現在の実行レベル
2. 実行レベルが最後に変更された日時
3. 現在の実行レベル
4. 最後にリブートしてからシステムがこの実行レベルになった回数
5. 以前の実行レベル

/etc/inittab ファイル

init または shutdown コマンドを使用してシステムをブートしたり実行レベルを変更したりすると、init デーモンは、/etc/inittab ファイルから情報を読み取ってプロセスを起動します。/etc/inittab ファイルには、init プロセスにとって重要な3つの情報が定義されています。

- システムのデフォルトの実行レベル
- 起動、監視するプロセス、および停止時に再起動するプロセス
- システムが新しい実行レベルに移行したとき行う処理

/etc/inittab ファイル内の各エントリは、次のフィールドからなります。

```
id:rstate :action :process
```

次の表に、inittab エントリの各フィールドを要約します。

表 11-2 inittab ファイルのフィールドの説明

フィールド	説明
<i>id</i>	エントリに固有の (一意の) 識別子。
<i>rstate</i>	このエントリが適用される実行レベルのリスト。
<i>action</i>	プロセスフィールドに指定されたプロセスの実行方法。指定できる値は、initdefault、sysinit、boot、bootwait、wait、respawn。 initdefault はデフォルトの実行レベル。他の action キーワードについては、inittab(4) のマニュアルページを参照。

表 11-2 inittab ファイルのフィールドの説明 (続き)

フィールド	説明
<code>process</code>	実行するコマンドまたはスクリプトを定義する。

例 — デフォルトの inittab ファイル

次の例では、Solaris リリースでインストールされるデフォルトの inittab ファイルを注釈付きで示します。

```

1 ap::sysinit:/sbin/autopush -f /etc/iu.ap
2 ap::sysinit:/sbin/soconfig -f /etc/sock2path
3 fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2<>/dev/msglog </dev/console
4 is:3:initdefault:
5 p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0>/dev/msglog 2<>/dev/...
6 sS:s:wait:/sbin/rcS >/dev/msglog 2<>/dev/msglog </dev/console
7 s0:0:wait:/sbin/rc0 >/dev/msglog 2<>/dev/msglog </dev/console
8 s1:1:respawn:/sbin/rc1 >/dev/msglog 2<>/dev/msglog </dev/console
9 s2:23:wait:/sbin/rc2 >/dev/msglog 2<>/dev/msglog </dev/console
10 s3:3:wait:/sbin/rc3 >/dev/msglog 2<>/dev/msglog </dev/console
11 s5:5:wait:/sbin/rc5 >/dev/msglog 2<>/dev/msglog </dev/console
12 s6:6:wait:/sbin/rc6 >/dev/msglog 2<>/dev/msglog </dev/console
13 fw:0:wait:/sbin/uadmin 2 0 >/dev/msglog 2<>/dev/msglog </dev/console
14 of:5:wait:/sbin/uadmin 2 6 >/dev/msglog 2<>/dev/msglog </dev/console
15 rb:6:wait:/sbin/uadmin 2 1 >/dev/msglog 2<>/dev/msglog </dev/console
16 sc:234:respawn:/usr/lib/saf/sac -t 300
17 co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
    -T terminal-type -d /dev/console -l console
-m ldterm,ttcompat

```

1. STREAMS モジュールを初期化します。
2. ソケット転送プロバイダを構成します。
3. ファイルシステムを初期化します。
4. デフォルトの実行レベルを定義します。
5. 電源障害の場合のシャットダウンを指定します。
6. シングルユーザーレベルを定義します。
7. 実行レベル 0 を定義します。
8. 実行レベル 1 を定義します。
9. 実行レベル 2 を定義します。
10. 実行レベル 3 を定義します。
11. 実行レベル 5 を定義します。
12. 実行レベル 6 を定義します。
13. 未使用レベル `firmware` を定義します。
14. 未使用レベル `off` を定義します。
15. 未使用レベル `reboot` を定義します。
16. サービスアクセスコントローラを初期化します。
17. コンソールを初期化し、端末のタイプを識別します。

システムが実行レベル 3 になると実行される処理

1. `init` プロセスが起動されます。`init` プロセスは、`/etc/default/init` ファイルを読み取って環境変数を設定します。デフォルトでは、`TIMEZONE` 変数だけが設定されます。
2. `init` は `inittab` ファイルを読み取り、次の処理を行います。
 - a. デフォルトの実行レベル 3 を定義する `initdefault` エントリを識別します。
 - b. `action` フィールドが `sysinit` になっているすべてのプロセスエントリを実行して、ユーザーがログインする前に特別な初期設定処理がすべて行われるようにします。
 - c. `rstate` フィールドが 3 になっている (デフォルトの実行レベル 3 と一致する) プロセスエントリを実行します。

`init` プロセスが `inittab` ファイルを使用する方法の詳細については、`init` (1M) のマニュアルページを参照してください。

次の表に、実行レベル 3 の `action` フィールドに使用するキーワードの説明を示します。

表 11-3 実行レベル 3 の `action` キーワードの説明

キーワード	説明
<code>powerfail</code>	<code>init</code> プロセスが電源切断シグナルを受信したときだけにプロセスを起動する。
<code>respawn</code>	プロセスを起動し、そのプロセスが終了したら再起動する。
<code>wait</code>	プロセスを起動し、そのプロセスが終了するまで待ってから、この実行レベルの次のエントリに進む。

次の表に、実行レベル 3 で実行されるプロセス (またはコマンド) の説明を示します。

表 11-4 実行レベル 3 のコマンドの説明

コマンドまたはスクリプト名	説明
<code>/usr/sbin/shutdown</code>	システムをシャットダウンする。 <code>init</code> プロセスは、システムが電源切断シグナルを受信した場合にのみ <code>shutdown</code> コマンドを実行する
<code>/sbin/rcS</code>	ルート (<code>/</code>)、 <code>/usr</code> 、 <code>/tmp</code> 、 <code>/var</code> 、 <code>/var/adm</code> 、および <code>/var/run</code> ファイルシステムをチェックし、マウントする
<code>/sbin/rc2</code>	標準のシステムプロセスを起動して、システムを実行レベル 2 (マルチユーザーレベル) に移行する

表 11-4 実行レベル 3 のコマンドの説明 (続き)

コマンドまたはスクリプト名	説明
/sbin/rc3	実行レベル 3 で使用される NFS リソース共有を開始する
/usr/lib/saf/sac -t 30	ポートモニターを起動する。このプロセスは失敗すると再起動される
/usr/lib/saf/ttymon -g -h -p " `uname -n` console login: " -T <i>terminal_type</i> -d /dev/console -l console	コンソールでのログイン要求を監視する <i>ttymon</i> プロセスを起動する。このプロセスは失敗すると再起動される SPARC システムの <i>terminal_type</i> は sun である x86 システムの <i>terminal_type</i> は AT386 である

実行制御スクリプト

注 - Solaris 環境でシステムサービスを起動および停止する方法は、将来のリリースで変更される可能性があります。

Solaris ソフトウェア環境では、一連の詳細な実行制御 (*rc*) スクリプトを使用して実行レベルの移行を制御しています。各実行レベルには次の *rc* スクリプトが対応しています。このスクリプトは、*/sbin* ディレクトリにあります。

- rc0
- rc1
- rc2
- rc3
- rc5
- rc6
- rcS

/sbin ディレクトリ内の各 *rc* スクリプトには */etc/rcn.d* という名前のディレクトリが対応しており、その中にはその実行レベルのさまざまな処理を実行するスクリプトがあります。たとえば、*/etc/rc2.d* には、実行レベル 2 のプロセスを起動および停止するためのファイル (スクリプト) があります。

```
# ls /etc/rc2.d
K03samba*           S20syssetup*       S72slpd*           S88utmpd*
K06mipagent*       S21perf*           S73cachefs.daemon* S89PRESERVE*
K07dmi*             S30sysid.net*     S73nfs.client*    S89bdconfig@
K07snmpdx*         S40llc2*           S74autofs*        S90wbem*
K16apache*         S42ncakmod*       S74syslog*        S91afbinit*
```

K21dhcp*	S47pppd*	S74xntpd*	S91gfbinit*
K26sshd*	S69inet*	S75cron*	S91ifbinit*
K27boot.server*	S70sckm*	S75flashprom*	S92volmgt*
K28kdc*	S70uucp*	S75savecore*	S93cacheos.finish*
K28kdc.master*	S71ldap.client*	S76nscd*	S94ncalogd*
K28nfs.server*	S71rpc*	S77sf880dr*	S95Iim*
README	S71sysid.sys*	S80lp*	S95svm.sync*
S01MOUNTFSYS*	S72autoinstall*	S80spc*	S98efcode*
S05RMTMPFILES*	S72directory@	S85power*	S99audit*
S10lu*	S72inetsvc*	S88sendmail*	S99dtlogin*

/etc/rcn.d 内のスクリプトは常に、スクリプト名を ASCII 文字列としてソートした順に実行されます。スクリプト名の形式は次のとおりです。

[KS] [0-9] [0-9] *

名前が K で始まるスクリプトを実行すると、システムサービスが停止 (kill) します。名前が S で始まるスクリプトを実行すると、システムサービスが起動します。

実行制御スクリプトは、/etc/init.d ディレクトリにもあります。これらのファイルは、/etc/rcn.d ディレクトリ内の対応する実行制御スクリプトにリンクされています。

各実行制御スクリプトの処理については、次の節で要約します。

実行制御スクリプトのまとめ

以降の節に、実行レベルを変更したときにシステムサービスを起動および停止するための実行制御スクリプトをまとめてあります。

/sbin/rc0 スクリプト

/sbin/rc0 スクリプトは、/etc/rc0.d スクリプトを実行して次の作業を行います。

- システムサービスとデーモンの終了
- 実行中の全プロセスの停止
- 全ファイルシステムのマウント解除

/sbin/rc1 スクリプト

/sbin/rc1 スクリプトは、/etc/rc1.d スクリプトを実行して次の作業を行います。

- システムサービスとデーモンの終了
- 実行中のすべてのユーザープロセスの停止
- すべてのリモートファイルシステムのマウント解除

- 以前の実行レベルが S であった場合、すべてのローカルファイルシステムのマウント

/sbin/rc2 スクリプト

/sbin/rc2 スクリプトは、/etc/rc2.d スクリプトを実行して、機能別にグループ化された次の作業を行います。

ローカルシステム関連の作業

- 以前の実行レベルが S であった場合、すべてのローカルファイルシステムのマウント
- quota オプションを指定してマウントされたファイルシステムが 1 つでも存在する場合は、ディスク割り当てを有効にする
- エディタの一時ファイルを /usr/preserve ディレクトリに保存する
- /tmp ディレクトリ内のすべてのファイルおよびサブディレクトリを削除する
- 設定されている場合、システムアクティビティデータの収集、システムアカウント、システム監査を起動する
- システムロギングデーモン (syslogd) を起動し、デフォルトのダンプデバイスを設定し、/var/adm/messages ファイルをローテーションする
- /etc/dispadmin.conf ファイルが存在する場合は、デフォルトのスケジューリングクラスを設定する
- ローカルプリンタが設定され、印刷待ち行列がクリーンアップされている場合は、LP 印刷サービス (lpsched) を起動する
- 必要に応じて、電源管理を設定する
- utmpd デーモンを起動する
- cron および vold デーモンを起動する
- シリアルデバイスストリームを設定する
- WBEM サービスを設定する
- 必要に応じてボリュームを同期化し、mdmonitor デーモンを起動してボリュームの物理コンポーネントを監視する
- 必要に応じて、CDE デスクトップログインプロセス dtlogin を起動する

ネットワークサービスまたはセキュリティ関連の作業

- 必要に応じて、ネットワークインタフェースを構成し、ifconfig ネットマスクを設定し、ネットワークルーティングを設定する
- ネットワークサービス (inetd および rpcbind) デーモンを起動する
- 論理リンクコントローラ (llc2) が設定されている場合は起動する
- システムがネームサービス用に設定されているかどうか、およびシステムがクライアントまたはサーバーのどちらであるかによって、ネームサービスのドメイン名を設定し、各種ネームサービスデーモンを起動する

- 必要に応じて、keyserv、statd、lockd、xntpd の各デーモンを起動する
- すべての NFS エントリをマウントする
- 必要に応じて、Solaris NCA (Network Cache and Accelerator) および NCA ログインを設定する
- Solaris PPP サーバーまたはクライアントデーモン (pppoed または pppd) が設定されている場合は起動する
- LDAP キャッシュマネージャ (ldap_cachemgr) が設定されている場合は起動する
- ディレクトリサーバー (slapd) デーモンが設定されている場合は起動する
- DNS (in.named) デーモンが設定されている場合は起動する
- サービスロケーションプロトコル (slpd) デーモンが設定されている場合は起動する
- /etc/rctladm.conf および /etc/pooladm.conf ファイルが存在する場合は、システムリソース制御とシステムプールを設定する
- 必要に応じて、cachefsd、automount、sendmail の各デーモンを起動する
- http_server プロセスを起動する

インストール関連の作業

- システム起動時またはシステム停止時の Live Upgrade ソフトウェアのブート環境を設定する
- /etc/.UNCONFIGURE ファイルがあるかどうかをチェックして、システムを設定し直す必要があるかどうか調べる
- /.PREINSTALL または /AUTOINSTALL のどちらかが存在する場合、インストールメディアまたはブートサーバーからシステムをリブートする

ハードウェア関連の作業

- 必要に応じて、Sun Fire 15000 のキー管理デーモン (sckmd) を起動する
- 必要に応じて、Sun Fire 880 の動的再構成デーモン (sf880drd) を起動する
- フラッシュ PROM の更新スクリプトを実行する
- グラフィックフレームバッファまたはグラフィックアクセラレータを設定する
- 必要に応じて、FCode インタプリタデーモン (efdaemon) を実行する

実行レベルを変更したときは、次のサービスを移行します。

- Apache (tomcat)
- ブートサーバー (in.rarpd、rpc.bootparamd、または rpld)
- DHCP (in.dhcpd)
- Kerberos KDC (krb5kdc) および Kerberos 管理 (kadmind)
- Mobile IP (mipagent)
- NFS サーバー (nfsd、mountd、nfslogd)
- Samba (smbd および nmbd)
- ssh (secure shell) (sshd)
- Solstice Enterprise Agents (dmispd および snmpXdmid)

注 - 実行レベル 2 で起動されるシステムサービスとアプリケーションの多くは、システム上にインストールされているソフトウェアによって決まります。

/sbin/rc3 スクリプト

/sbin/rc3 スクリプトは、/etc/rc3.d スクリプトを実行して次の作業を行います。

- Apache サーバーデーモン (tomcat) が設定されている場合は起動する
- 必要に応じて、DHCP デーモン (in.dhcpd) を起動する
- Kerberos KDC (krb5kdc) および Kerberos 管理 (kadmind) デーモンが設定されている場合は起動する
- Mobile IP デーモン (mipagent) が設定されている場合は起動する
- Samba デーモン (smbd および nmbd) が設定されている場合は起動する
- 必要に応じて、ssh (secure shell) デーモン (sshd) を起動する
- Solstice Enterprise Agents (dmispd および snmpXdmid) を起動する
- /etc/dfs/sharetab ファイルをクリーンアップする
- 必要に応じて、NFS サーバーデーモン nfsd、mountd、および nfslogd を起動する
- ブートサーバーの場合は、rarpd、rpc.bootparamd、rpld デーモンを起動する

/sbin/rc5 および /sbin/rc6 スクリプト

/sbin/rc5 および /sbin/rc6 スクリプトは、/etc/rc0.d/K* スクリプトを実行して、次の作業を行います。

- すべてのアクティブなプロセスを停止する
- ファイルシステムのマウント解除

/sbin/rcS スクリプト

/sbin/rcS スクリプトは、/etc/rcS.d スクリプトを実行して、システムを実行レベル S に移行します。これらのスクリプトによって次の作業が行われます。

- 最小限のネットワークの確立
- ルート (/)、/usr、/tmp、/var、/var/adm、および /var/run ファイルシステムのチェックとマウント
- システム名の設定
- 擬似ファイルシステム (/proc と /dev/fd) のマウント

- 再構成用ブートのデバイスエントリの再構築
- シングルユーザーレベルでマウントされる他のファイルシステムのチェックとマウント

実行制御スクリプトを使用してサービスを起動または停止する

注 – Solaris 環境でシステムサービスを起動および停止する方法は、将来のリリースで変更される可能性があります。

実行レベルごとに対応するスクリプトを持つことの利点は、`/etc/init.d` ディレクトリ内の個々のスクリプトを実行することによって、システムの実行レベルを変更しないで (現在の実行レベルの) システムサービスを停止できる点です。

▼ 実行制御スクリプトを使用してサービスを起動または停止する方法

1. スーパーユーザーになります。
2. システムサービスを停止します。

```
# /etc/init.d/filename stop
```
3. システムサービスを再開します。

```
# /etc/init.d/filename start
```
4. サービスが停止または起動していることを確認します。

```
# pgrep -f service
```

例 — 実行制御スクリプトを使用してサービスを起動または停止する

NFS サーバーデーモンを停止するには、次のように入力します。

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
#
```

そして、NFS サーバーデーモンを再開するには、次のように入力します。

```

# /etc/init.d/nfs.server start
# pgrep -f nfs
341
343
347
345
# pgrep -f nfs -d, | xargs ps -fp
UID    PID  PPID  C   STIME TTY      TIME CMD
daemon 341   1    0   Aug 21 ?       0:00 /usr/lib/nfs/statd
root   343   1    0   Aug 21 ?       0:00 /usr/lib/nfs/lockd
root   347   1    0   Aug 21 ?       0:41 /usr/lib/nfs/nfsd
root   345   1    0   Aug 21 ?       0:02 /usr/lib/nfs/mountd

```

実行制御スクリプトを追加する

注 - Solaris 環境でシステムサービスを起動および停止する方法は、将来のリリースで変更される可能性があります。

サービスを起動または停止するための実行制御スクリプトを追加するには、そのスクリプトを `/etc/init.d` ディレクトリにコピーします。次に、サービスを起動または停止する `rcn.d` ディレクトリ内にリンクを作成します。

実行制御スクリプトの命名法の詳細については、`/etc/rcn.d` ディレクトリ内の `README` ファイルを参照してください。次に、実行制御スクリプトを追加する方法について説明します。

▼ 実行制御スクリプトを追加する方法

1. スーパーユーザーになります。
2. スクリプトを `/etc/init.d` ディレクトリにコピーします。

```

# cp filename /etc/init.d
# chmod 0744 /etc/init.d/filename
# chown root:sys /etc/init.d/filename

```

3. 適切な `rcn.d` ディレクトリへのリンクを作成します。

```

# cd /etc/init.d
# ln filename /etc/rc2.d/Snnfilename
# ln filename /etc/rcn.d/Knnfilename

```

4. スクリプトのリンクが指定のディレクトリ内にあることを確認します。

```

# ls /etc/init.d/ /etc/rc2.d/ /etc/rcn.d/

```

例 — 実行制御スクリプトを追加する

次の例は、xyz サービスの実行制御スクリプトを追加する方法を示しています。

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S100xyz
# ln xyz /etc/rc0.d/K100xyz
# ls /etc/init.d /etc/rc2.d /etc/rc0.d
```

実行制御スクリプトを無効にする

実行制御スクリプトを無効にするには、ファイル名の先頭に下線 () を付けてファイル名を変更します。下線またはドット (.) で始まるファイルは実行されません。接尾辞を追加してファイルをコピーすると、両方のファイルが実行されます。

▼ 実行制御スクリプトを無効にする方法

1. スーパーユーザーになります。
2. スクリプト名の先頭に下線 () を付けて、スクリプト名を変更します。

```
# cd /etc/rcn.d
# mv filename _filename
```

3. スクリプト名が変更されたことを確認します。

```
# ls _*
# _filename
```

例 — 実行制御スクリプトを無効にする

次の例は、S100datainit スクリプトの名前を変更する方法を示しています。

```
# cd /etc/rc2.d
# mv S100datainit _S100datainit
# ls _*
# _S100datainit
```

x86: ブートファイル

これまでに説明した実行制御スクリプトおよびブートファイル以外にも、Solaris x86 システムのブートに使用する追加ブートファイルがあります。

表 11-5 x86: ブートファイル

ファイル	説明
/etc/bootrc	Solaris リリースのブートに使用するメニューとオプションを含む
/boot	システムをブートするために必要なファイルとディレクトリを含む
/boot/mdboot	第 1 レベルのブートストラッププログラム (strap.com) をディスクからメモリーにロードする DOS 実行可能ファイル
/boot/mdbootbp	第 1 レベルのブートストラッププログラム (strap.com) をフロッピーディスクからメモリーにロードする DOS 実行可能ファイル
/boot/rc.d	インストールスクリプトを含むディレクトリ。内容の変更はできない
/boot/solaris	ブートサブシステムの項目を含むディレクトリ
/boot/solaris/boot.bin	Solaris カーネルまたはスタンドアロンの kadb をロードする。この実行可能ファイルは、ブートファームウェアサービスも提供する
/boot/solaris/boot.rc	Solaris x86 プラットフォーム版を識別し、DOS エミュレーションモードで Device Configuration Assistant (デバイス構成用補助) を実行する
/boot/solaris/bootconf.exe	Device Configuration Assistant (デバイス構成用補助) の DOS 実行可能ファイル
/boot/solaris/bootconf.txt	Device Configuration Assistant (デバイス構成用補助) の国際化されたメッセージを含むテキストファイル (bootconf.exe)
/boot/solaris/bootenv.rc	ブート環境の設定用の eeprom 変数を格納する
/boot/solaris/devicedb	master ファイル (リアルモードドライバでサポートされるすべての有効なデバイスのデータベース) を含むディレクトリ
/boot/solaris/drivers	リアルモードドライバを含むディレクトリ

表 11-5 x86: ブートファイル (続き)

ファイル	説明
/boot/solaris/itup2.exe	インストール時更新 (ITU) プロセスで実行される DOS 実行可能ファイル
/boot/solaris/machines	以前は使用されていたが、現在はもう使用されていないディレクトリ
/boot/solaris/nbp	ネットワークブート関連ファイル
/boot/solaris/strap.rc	どのロードモジュールをロードするか、メモリー内のどこにロードするかの指示を含むファイル
/boot/strap.com	第 2 レベルのブートストラッププログラムをメモリーにロードする DOS 実行可能ファイル

第 12 章

システムのシャットダウン (手順)

この章では、システムのシャットダウン手順について説明します。この章で説明する手順は次のとおりです。

- 179 ページの「システムにログインしているユーザーを知る方法」
- 179 ページの「サーバーをシャットダウンする方法」
- 183 ページの「スタンドアロンシステムをシャットダウンする方法」
- 185 ページの「すべてのデバイスの電源を落とす方法」

この章の内容は次のとおりです。

- 178 ページの「システムシャットダウンコマンド」
- 179 ページの「システムのダウン時間をユーザーに通知する」
- 184 ページの「すべてのデバイスの電源を落とす」

システムの実行レベルについての概要は、第 11 章を参照してください。

システムのシャットダウン

Solaris ソフトウェアは、電子メールやネットワークソフトウェアをいつでも利用できるように、停止することなく動作するように設計されています。しかし、システム管理作業を行う場合や緊急事態が発生した場合は、システムをシャットダウンして安全に電源を切断できる状態にする必要があります。次のような場合には、システムを一部のシステムサービスしか利用できない中間の実行レベルまで移行する必要があります。

- ハードウェアを追加または削除する
- 予定された停電に備える
- ファイルシステムの保守を行う (バックアップなど)

システムをシャットダウンする必要があるシステム管理作業の詳細については、第 10 章を参照してください。

システムの電源管理機能を使用する方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』を参照してください。

システムシャットダウンコマンド

システムをシャットダウンする最も基本的な方法は、`init` コマンドおよび `shutdown` コマンドを使用する方法です。どちらのコマンドもシステムを「クリーンな状態でシャットダウン」するため、すべてのファイルシステムに対する変更がディスクに書き出され、すべてのシステムサービス、プロセス、オペレーティングシステムが正常に終了します。

システムのアポルトキーシーケンスを使用したり、電源をオフにしてからオンにする方法では、システムサービスが突然終了してしまうので、クリーンなシャットダウン方法とはいえません。しかし、緊急時には、これらの方法を使用しなければならない場合もあります。システムの復元方法については、第 13 章または第 14 章を参照してください。

次の表に、いくつかのシャットダウンコマンドとその用途を要約します。

表 12-1 シャットダウンコマンド

コマンド	説明	用途
<code>shutdown</code>	<code>init</code> を呼び出してシステムをシャットダウンする実行可能なシェルスクリプト。デフォルトでは、システムは実行レベル S に移行する	実行レベル 3 で動作しているサーバーで使用する。サーバーにログインしているユーザーに、サーバーが間もなくシャットダウンされることが通知される。シャットダウンされるサーバーのリソースをマウントしているシステムにも通知される
<code>init</code>	すべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能ファイル	他のユーザーが影響を受けないスタンダードシステムで使用する。ユーザーはまもなく行われるシャットダウンについて通知されないため、シャットダウンにかかる時間が短くて済む
<code>reboot</code>	ディスクを同期させ、ブート命令を <code>uadmin</code> システムコールに渡す実行可能ファイル。実際にプロセスを停止するのは、 <code>uadmin</code> システムコールである	推奨されない。代わりに、 <code>init</code> コマンドを使用する
<code>halt</code>	ディスクを同期させ、プロセスを停止する実行可能ファイル	<code>/etc/rc0</code> スクリプトを実行しないので推奨されない。このスクリプトは、すべてのプロセスを停止し、ディスクを同期させ、すべてのファイルシステムをマウント解除する

システムのダウン時間をユーザーに通知する

shutdown コマンドは起動時に、ログインしているすべてのユーザーと、システムリソースをマウントしているすべてのシステムに、警告と最終メッセージという形でシャットダウンを通知します。

サーバーをシャットダウンする必要がある場合に、init コマンドではなく shutdown コマンドを使用することを推奨するのはこのためです。どちらを使用するにしても、ユーザーには予定されているシャットダウンについてあらかじめ電子メールで知らせるようにしてください。

システム上のどのユーザーに通知する必要があるかを確認するには、who(1) コマンドを使用します。このコマンドは、システムの現在の実行レベルを調べる場合にも役立ちます。162 ページの「システムの実行レベルを確認する方法」を参照してください。

▼ システムにログインしているユーザーを知る方法

1. シャットダウンするシステムにログインします。
2. ログイン中のユーザーを表示します。

```
$ who
```

例 — システムにログインしているユーザーを知る

次の例は、システムにログインしているユーザーを表示する方法を示しています。

```
$ who
holly 1      console      May  7 07:30
kryten     pts/0 2      May  7 07:35      (starbug) 4
lister     pts/1      May  7 07:40 3 (bluemidget)
```

1. ログインしているユーザーのユーザー名
2. ログインしているユーザーの端末回線
3. ユーザーがログインした日時
4. (省略可能) リモートシステムからログインしているユーザーのホスト名

▼ サーバーをシャットダウンする方法

サーバーをシャットダウンする必要がある場合は、次の手順に従います。

1. スーパーユーザーになります。
2. システムにユーザーがログインしているか調べます。

```
# who
```

ログインしているすべてのユーザーがリスト表示されます。システムがシャットダウンされることを、メールかブロードキャストメッセージで知らせることをお勧めします。

3. システムをシャットダウンします。

```
# shutdown -iinit-level -ggrace-period -y
```

-iinit-level	システムをデフォルトの S 以外の init レベルにする。0、1、2、5、6 のいずれかを指定できる
-g grace-period	システムがシャットダウンするまでの時間 (秒) を示す。デフォルト値は 60 秒。
-y	ユーザーの介入なしにシャットダウンを継続する。このオプションを指定しないと、シャットダウンを継続するかどうか 60 秒後にたずねられる

詳細については、shutdown (1M) のマニュアルページを参照してください。

4. シャットダウンを継続するかどうかたずねられたら、**y** を入力します。

```
Do you want to continue? (y or n): y
```

-y オプションを指定した場合、このプロンプトは表示されません。

5. プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。

```
Type Ctrl-d to proceed with normal startup,  
(or give root password for system maintenance): xxx
```

6. システム管理作業を終了したら、**Control + D** を押してデフォルトの実行レベルに戻ります。

7. 次の表を使用して、システムが **shutdown** コマンドで指定した実行レベルに移行したことを確認します。

指定した実行レベル	SPARC システムのプロンプト	x86 システムのプロンプト
S (シングルユーザーレベル)	#	#
0 (電源切断レベル)	ok または >	type any key to continue
実行レベル 3 (リモートリソースを共有できるマルチユーザーレベル)	hostname console login:	hostname console login:

SPARC: 例 — サーバーを実行レベル S にする

次の例では、shutdown コマンドを使用して、3 分後に、SPARC システムを実行レベル S (シングルユーザーレベル) にしています。

```
# who
root      console      Dec 13 14:30
# shutdown -g180 -y

Shutdown started.      Thu Dec 13 14:30:32 MST 2001

Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth will be shut down in 3 minutes
.
.
Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth will be shut down in 30 seconds
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Shutting down Solaris Management Console server on port 898.
Print services stopped.
Dec 13 14:34:00 earth syslogd: going down on signal 15
Killing user processes: done.

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxxx
Entering System Maintenance Mode ...
#
```

SPARC: 例 — サーバーを実行レベル 0 にする

次の例では、shutdown コマンドを使用して、SPARC システムを 5 分後に実行レベル 0 にしています。確認用プロンプトが表示されないように -y オプションを指定しています。

```
# who
root      console      Dec 12 08:08
rimmer    pts/0          Dec 11 14:48      (starbug)
pmorph    pts/1          Dec 13 12:31      (bluemidget)
# shutdown -i0 -g300 -y
Shutdown started.      Thu Dec 13 14:51:39 MST 2001

Broadcast Message from root (console) on earth Thu Dec 13 14:51:39...
The system earth will be shut down in 5 minutes
.
.
```

```

.
Changing to init state 0 - please wait
#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type help for more information
ok

```

システムを実行レベル 0 にしてすべてのデバイスの電源を落とす場合は、185 ページの「すべてのデバイスの電源を落とす方法」を参照してください。

SPARC: 例 — サーバーをリブートして実行レベル 3 にする

次の例では、`shutdown` コマンドを使用して SPARC システムをリブートし、2 分後に実行レベル 3 にしています。確認用プロンプトが表示されないように `-y` オプションを指定しています。

```

# who
root      console      Dec 12 08:08
rimmer    pts/0          Dec 11 14:48   (starbug)
pmorph    pts/1          Dec 13 12:31   (bluemidget)
# shutdown -i6 -g120 -y
Shutdown started.   Thu Dec 13 15:56:30

Broadcast Message from root (console) on earth Thu Dec 13 15:56:30...
The system earth will be shut down in 2 minutes
.
.
.
Changing to init state 6 - please wait
#
INIT: New run level: 6
The system is coming down. Please wait.
.
.
.
The system is down.
syncing file systems... done
rebooting...
.
.
.
earth console login:

```

次に進む手順

システムをシャットダウンした理由が何であれ、最終的には、すべてのファイルリソースが使用でき、ユーザーがログインできる実行レベル 3 に戻すことになるでしょう。システムをマルチユーザーレベルに戻す手順については、第 13 章または第 14 章を参照してください。

▼ スタンドアロンシステムをシャットダウンする方法

スタンドアロンシステムをシャットダウンする必要がある場合は、次の手順を実行してください。

1. スーパーユーザーになります。
2. システムをシャットダウンします。

```
# init run-level
```

run-level は新しい実行レベルを指定します。

詳細については、`init(1M)` のマニュアルページを参照してください。

3. 次の表を使用して、システムが `init` コマンドで指定した実行レベルに移行したことを確認します。

指定した実行レベル	SPARC システムのプロンプト	x86 システムのプロンプト
S (シングルユーザーレベル)	#	#
2 (マルチユーザーレベル)	#	#
0 (電源切断レベル)	ok または >	type any key to continue
3 (NFS リソースを共有できる マルチユーザーレベル)	hostname console login:	hostname console login:

x86: 例 — スタンドアロンシステムを実行レベル 0 にする

次の例では、`init` コマンドを使用して、スタンドアロンの x86 システムを安全に電源を落とせるレベルにします。

```
# init 0
#
INIT: New run level: 0
The system is coming down. Please wait.
.
```

```
.  
.  
The system is down.  
syncing file systems... [11] [10] [3] done  
Type any key to continue
```

システムを実行レベル 0 にしてすべてのデバイスの電源を落とす場合は、185 ページの「すべてのデバイスの電源を落とす方法」を参照してください。

SPARC: 例 — スタンドアロンシステムを実行レベル S にする

次の例では、`init` コマンドを使用して、スタンドアロンの SPARC システムを実行レベル S (シングルユーザーレベル) にしています。

```
# init S  
#  
INIT: New run level: S  
The system is coming down for administration. Please wait.  
Unmounting remote filesystems: /vol nfs done.  
Print services stopped.  
syslogd: going down on signal 15  
Killing user processes: done.  
INIT: SINGLE USER MODE  
  
Type Ctrl-d to proceed with normal startup,  
(or give root password for system maintenance): xxx  
Entering System Maintenance Mode  
#
```

次に進む手順

システムをシャットダウンした理由が何であれ、最終的には、すべてのファイルリソースが使用でき、ユーザーがログインできる実行レベル 3 に戻すことになるでしょう。システムをマルチユーザーレベルに戻す手順については、第 13 章または第 14 章を参照してください。

すべてのデバイスの電源を落とす

次のような場合は、すべてのシステムデバイスの電源を落とす必要があります。

- ハードウェアを置換または追加する
- システムの設置場所を変更する

- 予定された停電や自然災害 (接近中の雷雨など) に備える

電源を落とすシステムデバイスとは、CPU、モニター、外部デバイス (ディスク、テープ、プリンタ) などを意味します。

すべてのシステムデバイスの電源を落とす前に、前の節の説明に従って、システムをクリーンにシャットダウンする必要があります。

▼ すべてのデバイスの電源を落とす方法

1. 次のどちらかを選択して、システムをシャットダウンします。
 - a. サーバーをシャットダウンする場合は、**179** ページの「サーバーをシャットダウンする方法」を参照
 - b. スタンドアロンシステムをシャットダウンする場合は、**183** ページの「スタンドアロンシステムをシャットダウンする方法」を参照
2. システムをシャットダウンしたら、すべてのデバイスの電源を落とします。必要な場合は、電源ケーブルをコンセントから引き抜きます。
3. 電源が回復したら、次の手順に従ってシステムとデバイスの電源を投入します。
 - a. 電源ケーブルをコンセントに差し込みます。
 - b. モニターの電源を入れます。
 - c. ディスクドライブ、テープドライブ、プリンタの電源を入れます。
 - d. CPU の電源を入れます。
システムが実行レベル 3 になります。

第 13 章

SPARC: システムのブート (手順)

この章では、OpenBoot™ PROM モニターを使用する手順と、SPARC システムを異なる実行レベルでブートする手順について説明します。

SPARC システムのブート手順については、187 ページの「SPARC: システムのブート (作業マップ)」を参照してください。

ブートプロセスの概要については、第 10 章を参照してください。ブートに関する問題の障害追跡を行うには、『Solaris のシステム管理 (上級編)』の「リポートが失敗した場合の対処」を参照してください。

x86 システムのブート手順については、第 14 章を参照してください。

SPARC: システムのブート (作業マップ)

作業	説明	参照先
ブート PROM の使用	ブート PROM はシステムのブートに使用する。必要に応じて、次のいずれかの作業を行う PROM のリビジョン番号を確認する システムのブートに使用できるデバイスを確認する	189 ページの「SPARC: システムの PROM リビジョンを確認する方法」 189 ページの「SPARC: システム上のデバイスを確認する方法」

作業	説明	参照先
システムのブート	<p>デフォルトのブートデバイスを変更する (新しいディスクを追加した場合、あるいはシステムのブート方法を変更する必要がある場合)</p> <p>次のいずれかのブート方法を選択する</p> <p>実行レベル 3 でブートする - システムをシャットダウンするか、なんらかのシステムハードウェアの保守作業を行った後で使用する</p> <p>実行レベル S でブートする - なんらかのシステム保守作業 (ファイルシステムのバックアップなど) を行なった後で使用するこのレベルでは、一部のファイルシステムだけがマウントされ、ユーザーはシステムにログインできない</p> <p>対話式でブートする - テストのために、システムファイルまたはカーネルを一時的に変更した後で使用する</p> <p>ネットワークからブートする - システムをネットワークからブートするときに使用する。この方法は、ディスクレスクライアントをブートする場合に使用する</p> <p>復元を目的としてブートする - 損傷したファイルまたはファイルシステムによってシステムのブートが行えないときにシステムをブートするために使用する。復元を目的としてブートするには、必要に応じて次のどちらかまたは両方の作業を行う</p> <p>まず、システムを停止して復元を試みる</p> <p>システムをブートして、システムの正常なブートを妨げている重要なシステムファイルを修復する</p> <p>kadbd をブートする - システムに関する問題の障害追跡を行うときに使用する</p>	<p>191 ページの「SPARC: デフォルトのブートデバイスを変更する方法」</p> <p>194 ページの「SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」</p> <p>195 ページの「SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法」</p> <p>196 ページの「SPARC: システムを対話式でブートする方法」</p> <p>197 ページの「SPARC: システムをネットワークからブートする方法」</p> <p>198 ページの「SPARC: 復元を目的としてシステムを停止する方法」</p> <p>199 ページの「SPARC: 復元を目的としてシステムをブートする方法」</p> <p>198 ページの「SPARC: 復元を目的としてシステムを停止する方法」</p>

作業	説明	参照先
	クラッシュダンプを強制実行し、システムをリポートする - 障害追跡の目的でクラッシュダンプを強制実行するとき使用する	202 ページの「SPARC: クラッシュダンプを強制してシステムをリポートする方法」

SPARC: ブート PROM の使用

システム管理者は、通常 PROM レベルを使ってシステムをブートします。ただし、システムのブート方法の変更が必要な場合があります。たとえば、システムをマルチユーザーレベルにする前に、どのデバイスからブートするかを設定し直したり、ハードウェア診断プログラムを実行したりする場合があります。

次の作業を行う場合は、デフォルトのブートデバイスを変更する必要があります。

- 新しいドライブを永久または一時的にシステムに追加する
- ネットワークブート方法を変更する
- スタンドアロンシステムを一時的にネットワークからブートする

すべての PROM コマンドについては、`monitor(1M)` または `eeprom(1M)` のマニュアルページを参照してください。

SPARC: システムの PROM リビジョンを確認する方法

システムの PROM リビジョンレベルを `banner` コマンドで表示します。

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

ハードウェア構成情報が表示され、その中に PROM のリビジョン番号があります。この例では、PROM のリビジョン番号は 3.15 です。

▼ SPARC: システム上のデバイスを確認する方法

システム上のデバイスを確認して、ブートに適したデバイスを見つけ出すことが必要な場合があります。

`probe` コマンドを使用してシステムに接続されているデバイスを安全に見つけるためには、次のことを行なっておく必要があります。

- PROM の `auto-boot?` パラメータの値を `false` に変更する
- `reset-all` コマンドを発行して、システムのレジスタをクリアする

次のように `sifting probe` コマンドを使用すると、システム上で利用可能な `probe` コマンドを表示できます。

```
ok sifting probe
```

システムのレジスタをクリアしないで `probe` コマンドを実行すると、次のメッセージが表示されます。

```
ok probe-scsi
This command may hang the system if a Stop-A or halt command
has been executed. Please type reset-all to reset the system
before executing this command.
Do you wish to continue? (y/n) n
```

1. PROM の `auto-boot?` パラメータの値を `false` に変更します。

```
ok setenv auto-boot? false
```

2. システムのレジスタをクリアします。

```
ok reset-all
```

3. システム上のデバイスを確認します。

```
ok probe-device
```

4. (省略可能) 電源障害後や `reset` コマンドの使用後にシステムをリブートするには、`auto-boot?` パラメータを `true` に戻します。

```
ok setenv auto-boot? true
auto-boot? = true
```

SPARC: 例 — システム上のデバイスを確認する

次の例は、Ultra10 システムに接続されているデバイスの確認方法を示しています。

```
ok setenv auto-boot? false
auto-boot? = false
ok reset-all
Resetting ...
```

```
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.
```

```
ok probe-ide
Device 0 ( Primary Master )
        ATA Model: ST34321A

Device 1 ( Primary Slave )
        Not Present
```

```
Device 2 ( Secondary Master )
Removable ATAPI Model: CRD-8322B
```

```
Device 3 ( Secondary Slave )
Not Present
```

```
ok setenv auto-boot? true
auto-boot? = true
```

devalias コマンドを使うと、システムに接続されている「可能性のある」デバイスの別名と関連パスを確認できます。

```
ok devalias
screen /pci@1f,0/pci@1,1/SUNW,m64B@2
net /pci@1f,0/pci@1,1/network@1,1
cdrom /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f
disk /pci@1f,0/pci@1,1/ide@3/disk@0,0
disk3 /pci@1f,0/pci@1,1/ide@3/disk@3,0
disk2 /pci@1f,0/pci@1,1/ide@3/disk@2,0
disk1 /pci@1f,0/pci@1,1/ide@3/disk@1,0
disk0 /pci@1f,0/pci@1,1/ide@3/disk@0,0
ide /pci@1f,0/pci@1,1/ide@3
floppy /pci@1f,0/pci@1,1/ebus@1/fdthree
ttyb /pci@1f,0/pci@1,1/ebus@1/se:b
ttya /pci@1f,0/pci@1,1/ebus@1/se:a
keyboard! /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8:forcemode
keyboard /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
name aliases
```

▼ SPARC: デフォルトのブートデバイスを変更する方法

デフォルトのブートデバイスをほかのデバイスに変更するには、まずシステム上のデバイスを確認しておく必要があります。システム上のデバイスを確認する方法については、189 ページの「SPARC: システム上のデバイスを確認する方法」を参照してください。

1. スーパーユーザーになります。
2. 実行レベル 0 に変更します。

```
# init 0
```

```
ok PROM プロンプトが表示されます。
```

```
詳細については、init(1M) のマニュアルページを参照してください。
```

3. **boot-device** パラメータの値を変更します。

```
ok setenv boot-device device [n]
```

<code>boot-device</code>	ブートするデバイスを設定するパラメータを示す
<code>device[n]</code>	<code>boot-device</code> の値 (disk または net) を設定する。n はディスク番号

ディスク番号を確認する必要がある場合は、`probe` コマンドのいずれかを使用します。

4. デフォルトのブートデバイスが変更されていることを確認します。

```
ok printenv boot-device
```

5. 新しい `boot-device` 値を保存します。

```
ok reset
```

新しい `boot-device` 値が PROM に書き込まれます。

SPARC: 例 — デフォルトのブートデバイスを変更する

この例では、デフォルトのブートデバイスをディスクに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device disk
boot-device =          disk
ok printenv boot-device
boot-device          disk          disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Boot device: disk File and args:
SunOS Release 5.9 Version 64-bit
.
.
.
pluto console login:
```

この例では、デフォルトのブートデバイスをネットワークに設定しています。

```
# init 0
#
INIT: New run level: 0
.
```



```
.  
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device =          net
ok printenv boot-device
boot-device            net                disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

```
Boot device: net  File and args:
```

```
.  
.
.  
pluto console login:
```

SPARC: システムをリセットする方法

ok プロンプトで `reset` コマンドを実行します。

```
ok reset
```

セルフテストプログラムが実行され、ハードウェアで診断テストを行なった後、システムがリブートされます。

SPARC: システムのブート

システムの電源を切ってから入れ直すと、マルチユーザーのブートシーケンスが開始されます。次に示す手順は、ok PROM プロンプトからさまざまな実行レベルでブートする方法です。これらの手順では、特に指示がない限り、システムがクリーンな状態でシャットダウンしていることを前提とします。

`who -r` コマンドを使って、システムが指定した実行レベルになっていることを確認します。実行レベルについての説明は、第 11 章を参照してください。

▼ SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル 3 でブートします。

1. システムを実行レベル 3 でブートします。

```
ok boot
```

自動ブート処理では、一連のスタートアップメッセージ表示して、システムを実行レベル 3 にします。

詳細については、boot (1M) のマニュアルページを参照してください。

2. システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

SPARC: 例 — システムを実行レベル 3 (マルチユーザーレベル) でブートする

次の例は、システムを実行レベル 3 でブートしたときに表示されるメッセージを示しています。

```
ok boot
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Rebooting with command: boot
Boot device: disk:a File and args:
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
configuring IPv4 interfaces: hme0.
Hostname: starbug
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdisk/c0t0d0s7: is clean.
/dev/rdisk/c0t0d0s4: is clean.
NIS domainname is Solar.COM
starting rpc services: rpcbind keyserver ypbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4:
gateway starbug
syslog service starting.
Print services started.
volume management starting.
The system is ready.
```

starbug console login:

▼ SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル S でブートします。

1. システムを実行レベル S でブートします。

```
ok boot -s
```

2. 次のメッセージが表示されたら、スーパーユーザーのパスワードを入力します。

```
INIT: SINGLE USER MODE
Type Ctrl-d to proceed with normal startup,

(or give root password for system maintenance): xxx
```

3. システムが実行レベル S になっていることを確認します。

```
# who -r
.          run-level S  Jun 10 15:27    3      0
```

4. システム保守作業の後に、システムをマルチユーザー状態にするには、**Control + D**を押します。

SPARC: 例 — システムを実行レベル S (シングルユーザーレベル) でブートする

次の例は、システムを実行レベル S でブートしたときに表示されるメッセージを示しています。

```
ok boot -s
.
.
.
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
configuring IPv4 interfaces: le0.
Hostname: earth

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Sun Microsystems Inc.  SunOS 5.9  Generic May 2002
# who -r
.          run-level S  Jul 14 11:37    3      0  ?
```

(保守作業を行う)
<Control - D を押す

▼ SPARC: システムを対話式でブートする方法

次の手順に従ってシステムをブートし、代替カーネルまたは `/etc/system` ファイルを指定する必要があります。

1. システムを対話式でブートします。

```
ok boot -a
```

2. 次の表に示すように、システムプロンプトに答えてください。

システムプロンプト	操作
Enter filename [kernel/[sparcv9]/unix]:	ブートに使用するカーネルの名前を入力する。あるいは、そのまま Return キーを押してデフォルトのカーネルを使用する
Enter default directory for modules [/platform/`uname - i`/kernel /platform/`uname - m/kernel /kernel /usr/kernel]:	modules ディレクトリの代替パスを入力する。あるいは、そのまま Return キーを押してデフォルトのカーネルモジュールディレクトリを使用する
Name of system file [etc/system]:	代替システムファイルの名前を入力して Return キーを押す。/etc/system ファイルが破損している場合、/dev/null を入力する。あるいは、そのまま Return キーを押してデフォルトの etc/system ファイルを使用する
root filesystem type [ufs]:	そのまま Return キーを押してデフォルトのルート (/) ファイルシステムを使用する。ローカルディスクからのブートの場合は UFS、ネットワークブートの場合は NFS を入力する
Enter physical name of root device [physical_device_name]:	代替デバイス名を入力して、Return キーを押す。あるいは、そのまま Return キーを押してルートデバイスのデフォルトの物理名を使用する

3. 前の表の質問に応答するためのプロンプトが表示されない場合は、`boot -a` コマンドを正しく入力していることを確認してください。

SPARC: 例 — システムを対話式でブートする

次の例では、利用できるデフォルトの選択例 ([] で囲まれた部分) を示します。

```

ok boot -a
.
.
.
Rebooting with command: boot -a
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a File and args: -a
Enter filename [kernel/sparcv9/unix]: <Return キーを押す>
Enter default directory for modules [/platform/SUNW,Ultra-5_10/kernel
/platform/sun4u/kernel /kernel /usr/kernel]: <Return キーを押す>
Name of system file [etc/system]: <Return キーを押す>
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
root filesystem type [ufs]: <Return キーを押す>
Enter physical name of root device
[/pci@1f,0/pci@1,1/ide@3/disk@0,0:a]: <Return キーを押す>
configuring IPv4 interfaces: hme0.
Hostname: starbug
The system is coming up. Please wait.
checking ufs filesystems
.
.
.
The system is ready.
starbug console login:

```

▼ SPARC: システムをネットワークからブートする方法

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。たとえば、スタンドアロンのシステムがローカルディスクからブートできない場合、そのシステムを一時的にネットワークからブートできます。デフォルトのブートデバイスを変更または再設定する方法については、191 ページの「SPARC: デフォルトのブートデバイスを変更する方法」を参照してください。

sun4u システムでは次の 2 つのネットワーク構成ブート方法を利用できます。

- RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol)
- DHCP (Dynamic Host Configuration Protocol)

デフォルトのネットワークブート方法は RARP に設定されています。ネットワークで利用できるブートサーバーによって、RARP または DHCP を選択できます。

注 - Sun Ultra システムで DHCP ネットワークブート方法を使用するには、PROM のバージョンが 3.25.nn 以上でなければなりません。PROM のバージョンの確認方法については、189 ページの「SPARC: システムの PROM リビジョンを確認する方法」を参照してください。

RARP と DHCP の両方のネットワークブート方法を利用できる場合、どちらのサービスを使用するかを `boot` コマンドに一時的に指定できます。あるいは、NVRAM 別名を設定すれば、システムをリブートしても有効に PROM レベルでネットワークブート方法を永続的に保存することができます。次の `nvalias` コマンドの例では、Sun Ultra 10 システムにおいてデフォルトで DHCP でブートするように、ネットワークデバイスの別名を設定します。

```
ok nvalias net /pci@1f,4000/network@1,1:dhcp
```

この別名を設定している場合、`boot net` と入力するだけで、システムは DHCP ネットワークブート方法を使用してブートします。



注意 - `nvunalias` コマンドと `nvalias` コマンドの構文を十分理解するまで、`nvunalias` コマンドで NVRAMRC ファイルを変更しないでください。これらのコマンドの使用方法については、『*OpenBoot 3.x コマンド・リファレンスマニュアル*』を参照してください。

1. 必要な場合、システムをシャットダウンします。
2. ネットワークからブートする方法を決定し、次のどちらかを選択します。
どちらの方法でブートする場合でも、RARP または DHCP のブートサーバーがすでにネットワークに設定されていなければなりません。

- a. **DHCP** を使用してシステムをネットワークからブートするには、次のように入力します。

```
ok boot net[:dhcp]
```

上記 `nvalias` の例のように、デフォルトで DHCP でブートするように PROM 設定を変更してある場合は、`boot net` と指定するだけで、システムは DHCP でブートします。

- b. **RARP** を使用してシステムをネットワークからブートするには、次のように入力します。

```
ok boot net[:rarp]
```

RARP はデフォルトのネットワークブート方法です。このため、デフォルトで DHCP でブートするように PROM 値を変更してある場合にだけ、`boot net:rarp` と指定しなければなりません。

▼ SPARC: 復元を目的としてシステムを停止する方法

1. システムのアポートキーシーケンスを入力します。
モニターに、`ok PROM` プロンプトが表示されます。

```
ok
```

アポートキーシーケンスは、キーボードのタイプによって異なります。たとえば、Stop+A または L1+A を押します。端末では、Break キーを押します。

2. ファイルシステムを同期させます。

```
ok sync
```

3. **syncing file systems...** というメッセージが表示されたら、システムのアポートキーシーケンスをもう一度入力します。

4. 該当する **boot** コマンドを入力して、ブートプロセスを起動します。
詳細については、`boot(1M)` のマニュアルページを参照してください。

5. システムが指定した実行レベルになっていることを確認します。

```
# who -r
.          run-level 3  May  2 07:39      3      0  S
```

SPARC: 例 — 復元を目的としてシステムを停止する

```
<Stop + A を押す>
ok sync
syncing file systems...
<Stop + A を押す>
ok boot
```

▼ SPARC: 復元を目的としてシステムをブートする方法

次の手順は、`/etc/passwd` などの重要なファイルに、無効なエントリがあり正常にブートできない場合に使用します。

次の手順で使用されている変数 `devicename` は、修復するファイルシステムのデバイス名に置き換えてください。システムのデバイス名を調べる場合は、第 30 章を参照してください。

1. システムのアポートキーシーケンスを使用して、システムを停止します。
システムのアポートキーシーケンスは、ルートのパスワードがわからない場合やシステムにログインできない場合に使用します。詳細については、198 ページの「SPARC: 復元を目的としてシステムを停止する方法」を参照してください。
2. **Solaris 9 Installation CD** または **DVD** からブートしているか、あるいはネットワークからブートしているかによって、次の表のそれぞれの手順を使い分けてください。

ブート方法	手順
Solaris 9 installation CD または DVD	1. Solaris 9 Installation CD または DVD をドライブに挿入する 2. Solaris 9 Installation CD または DVD から、シングルユーザーモードでブートする <code>ok boot cdrom -s</code>
ネットワーク (インストールサーバーまたはリモート CD/DVD ドライブが使用できる場合)	次のコマンドを使う <code>ok boot net -s</code>

3. ファイル内に無効なエントリがあるファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /a
```

4. 新しくマウントしたファイルシステムに移動します。

```
# cd /a/file-system
```

5. 端末タイプを設定します。

```
# TERM=sun
# export TERM
```

6. エディタを使って、ファイルから無効なエントリを削除します。

```
# vi filename
```

7. ルート (/) ディレクトリに変更します。

```
# cd /
```

8. /a ディレクトリのマウントを解除します。

```
# umount /a
```

9. システムをリブートします。

```
# init 6
```

10. システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```


SPARC: 例 — 復元を目的としてシステムをブートする (パスワードファイルが損傷した場合)

次の例は、ローカルの CD-ROM からブートした後、重要なシステムファイル (この場合は /etc/passwd) を修復する方法を示しています。

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi passwd
(無効なエントリを削除する)
# cd /
# umount /a
# init 6
```

SPARC: 例 — ルートのパスワードを忘れた場合にシステムをブートする

次の例は、ルートのパスワードを忘れた場合に、ネットワークからブートしてシステムを復元する方法を示しています。この例では、ネットワークのブートサーバーが利用可能になっていることを前提とします。システムをリブートした後で必ず新しいルートパスワードを適用してください。

```
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
(ルートの暗号化パスワードを削除する)
# cd /
# umount /a
# init 6
```

▼ SPARC: カーネルデバッグ (kadb) を使ってシステムをブートする方法

1. システムを停止する必要がある場合は、システムのアボートキーシーケンスを入力します。

アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、Stop+A または L1+A を押します。端末では、Break キーを押します。

ok PROM プロンプトが表示されます。

2. ファイルシステムを同期させ、クラッシュダンプを書き出します。

```
> n
ok sync
```

3. **syncing file systems...** というメッセージが表示されたら、システムのアポートキーシーケンスをもう一度入力します。

4. カーネルデバッグを使ってシステムをブートします。

```
ok boot kadb
```

5. ブートメッセージで、システムがカーネルデバッグ (**kadb**) を使用してブートしているかどうかをチェックします。

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

SPARC: 例 — カーネルデバッグ (kadb) を使ってシステムをブートする

```
<Stop - A を押す>
ok sync
syncing file systems...
<Stop - A を押す>
ok boot kadb
```

SPARC: クラッシュダンプを強制してシステムをリブートする

場合によっては、障害追跡の目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できません。

システムのクラッシュダンプについては、『Solaris のシステム管理 (上級編)』の「システムクラッシュ情報の管理 (手順)」を参照してください。

▼ SPARC: クラッシュダンプを強制してシステムをリブートする方法

`savecore` 機能が使用できる場合は、次の手順に従ってクラッシュダンプを強制実行し、システムをリブートします。

1. システムのアポートキーシーケンスを入力します。

アポルトキーシーケンスは、キーボードのタイプによって異なります。たとえば、Stop + A または L1 + A を押します。端末では、Break キーを押します。ok PROM プロンプトが表示されます。

2. ファイルシステムを同期させ、クラッシュダンプを書き出します。

```
> n  
ok sync
```

クラッシュダンプがディスクに書き出されると、システムはそのままリブートします。

3. システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

SPARC: 例 — クラッシュダンプを強制してシステムをリブートする

```
<Stop + A を押す>  
ok sync
```


第 14 章

x86: システムのブート (手順)

この章では、x86 システムをブートする手順を説明します。

x86 システムのブート手順については、205 ページの「x86: システムのブート (作業マップ)」を参照してください。

ブートプロセスの概要については、第 10 章を参照してください。

SPARC システムのブート手順については、第 13 章を参照してください。

x86: システムのブート (作業マップ)

作業	説明	参照先
Solaris Device Configuration Assistant (デバイス構成用補助) のブート	システムのハードウェア構成を変更した後で使用する。このユーティリティを使用すると、異なるブートデバイスから Solaris システムをブートする、新しいハードウェアを構成する、間違って構成したハードウェアを構成し直すなど、デバイス関連やブート関連の作業を行うことができる	207 ページの「x86: Solaris Device Configuration Assistant (デバイス構成用補助) をブートする方法」
システムのブート	次のいずれかのブート方法を選択する	

作業	説明	参照先
	<p>実行レベル 3 でブートする - システムをシャットダウンするか、なんらかのシステムハードウェアの保守作業を行なった後で使用する</p> <p>実行レベル S でブートする - なんらかのシステム保守作業 (ファイルシステムのバックアップなど) を行なった後で使用する</p> <p>対話式でブートする - テストのために、システムファイルまたはカーネルを一時的に変更した後で使用する</p> <p>ネットワークからブートする - システムをネットワークからブートするときに使用する。この方法は、ディスクレスクライアントをブートする場合に使用する</p> <p>復元を目的としてブートする - 損傷したファイルによってシステムのブートが行えないときにシステムをブートするために使用する。復元を目的としてブートするには、必要に応じて次のどちらかまたは両方の作業を行う</p> <p>まず、システムを停止して復元を試みる</p> <p>システムをブートして、システムの正常なブートを妨げている重要なシステムファイルを修復する</p> <p>kadb をブートする - システムに関する問題の障害追跡を行うときに使用する</p> <p>クラッシュダンプを強制実行し、システムをリポートする - 障害追跡の目的でクラッシュダンプを強制実行するときに使用する</p>	<p>208 ページの「x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」</p> <p>209 ページの「x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法」</p> <p>210 ページの「x86: システムを対話式でブートする方法」</p> <p>212 ページの「x86: システムをネットワークからブートする方法」</p> <p>213 ページの「x86: 復元を目的としてシステムを停止する方法」</p> <p>213 ページの「x86: 復元を目的としてシステムをブートする方法」</p> <p>218 ページの「x86: カーネルデバッガ (kadb) を使ってシステムをブートする方法」</p> <p>219 ページの「x86: クラッシュダンプを強制してシステムをリポートする」</p>

x86: Solaris Device Configuration Assistant (デバイス構成用補助) のブート

Solaris Device Configuration Assistant (デバイス構成用補助) (Solaris x86 プラットフォーム版) は、さまざまなハードウェア構成作業やブート作業を実行するためのプログラムです。Solaris Device Configuration Assistant (デバイス構成用補助) にアクセスする方法は 2 つあります。

- Solaris ブートフロッピーディスク
- Solaris 9 Installation CD または DVD

この章で説明する手順では、Solaris Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクを挿入して、Device Configuration Assistant (デバイス構成用補助) をブートするように要求されることがあります。システムの BIOS が CD または DVD からのブートをサポートしている場合は、Solaris 9 Installation CD または DVD を挿入して、Device Configuration Assistant (デバイス構成用補助) をブートすることも可能です。

▼ x86: Solaris Device Configuration Assistant (デバイス構成用補助) をブートする方法

1. **Solaris Device Configuration** ブートフロッピーディスク、**Solaris 9 Installation CD** または **DVD** を適切なドライブに挿入します。

2. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

Device Configuration Assistant (デバイス構成用補助) の最初のメニューが、数分後に表示されます。

x86: システムのブート

次に、リセットボタンを使ってシステムを再起動する手順を示します。システムにリセットボタンがない場合は、電源スイッチを使ってシステムを再起動します。システムの状態によっては Control + Alt + Del キーを使って、システムの動作に割り込むことができます。

▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法

次の手順に従って、システム (現時点で実行レベル 0 になっている) を実行レベル 3 でブートします。

1. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。
「Current Boot Parameters」メニューが、数分後に表示されます。
2. システムを実行レベル 3 にするには、**b** を入力します。**Enter** キーを押します。
5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
3. システムが実行レベル 3 になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

x86: 例 — システムを実行レベル 3 (マルチユーザーレベル) でブートする

```
Type any key to continue
.
.
.
          <<< Current Boot Parameters>>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                               to enter boot interpreter
or      <ENTER>                                 to boot with defaults

          <<< timeout in 5 seconds>>>

Select (b)oot or (i)nterpreter: b
.
.
.
venus console login:
```


▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法

次の手順に従って、システム (現時点では実行レベル 0 になっている) システムを実行レベル S でブートします。

1. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

「Current Boot Parameters」メニューが、数分後に表示されます。

2. **b -s** と入力して、システムを実行レベル **S** でブートします。**Enter** キーを押します。
5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
3. プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。
4. **who -r** を使用して、システムが実行レベル **S** になっていることを確認します。

```
# who -r
.          run-level S  Jul 19 14:37      S      0 3
```

5. 実行レベルを **S** に移行して行う必要がある保守作業を行います。
6. システムを実行レベル **3** に戻すには、**Control + D** を押します。

x86: 例 — システムを実行レベル S (シングルユーザーレベル) でブートする

```
Type any key to continue
.
.
.

          <<< Current Boot Parameters>>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                  to boot with defaults

          <<< timeout in 5 seconds>>>

Select (b)oot or (i)nterpreter: b -s
.
.
.
```

```

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode
.
.
.
# who -r
.          run-level S  Jul 19 14:37      S      0  3
(システム保守作業を行う)
# <Control - D を押す>

```

▼ x86: システムを対話式でブートする方法

次の手順に従ってシステムをブートし、代替カーネルまたは /etc/system ファイルを指定する必要があります。

- 1. Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

「Primary Boot Subsystem」メニューが、数分後に表示されます。
- 2. Solaris** パーティションにアクティブを示すマークが付いていない場合、その **Solaris** パーティションをリストから選択して、**Enter** キーを押します。

5 秒以内に選択しない場合、アクティブなブートパーティションが自動的に選択されます。

「Current Boot Parameters」メニューが、数分後に表示されます。
- 3. b -a** と入力して、システムを対話式でブートします。**Enter** キーを押します。

5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
- 4.** 次の表に示すように、システムプロンプトに答えてください。

システムプロンプト	操作
Enter default directory for modules: [/platform/i86pc/kernel /kernel /usr/kernel]:	modules ディレクトリの代替パスを指定し、 Enter キーを押す。あるいは、そのまま Enter キーを押してデフォルトの modules ディレクトリを使用する

システムプロンプト	操作
Name of system file [etc/system]:	代替システムファイルの名前を入力して Enter キーを押す。あるいは、そのまま Enter キーを押してデフォルトの /etc/system ファイルを使う。 /etc/system ファイルが破損している場合は、ファイルとして /dev/null を入力する
root filesystem type [ufs]:	そのまま Enter キーを押してデフォルトの ルート (/) ファイルシステムを使用する。 ローカルディスクからのブートの場合は UFS、ネットワークブートの場合は NFS と入力する
Enter physical name of root device [physical_device_name]:	代替デバイス名を入力して、Enter キーを押す。あるいは、そのまま Enter キーを押して ルートデバイス bootpath のデフォルトの 物理名を使用する

x86: 例 — システムを対話式でブートする

次に、利用できるデフォルトの選択例を示します ([] で囲まれた部分)。

```
Type any key to continue
.
.
.

<<< Current Boot Parameters>>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                  to boot with defaults

<<< timeout in 5 seconds>>>
Select (b)oot or (i)nterpreter: b -a
Enter default directory for modules [/platform/i86pc/kernel /kernel
/usr/kernel]: <Enter キーを押す>
Name of system file [etc/system]:<Enter キーを押す>
SunOS Release 5.9 Version Generic 32-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
root filesystem type [ufs]: <Enter キーを押す>
Enter physical name of root device
[/pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a]: <Enter キーを押す>
configuring IPv4 interfaces: dnet0.
Hostname: venus
(fsck メッセージが表示される)
The system is coming up. Please wait
(メッセージが表示される)
venus console login:
```

▼ x86: システムをネットワークからブートする方法

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。たとえば、スタンドアロンのシステムがローカルディスクからブートできない場合は、このシステムを一時的にネットワークからブートできます。

システムが PXE ネットワークブートに対応している場合は、Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクや Solaris 9 Installation CD または DVD を使用しなくても、ネットワークから直接システムをブートできます。

Device Configuration Assistant (デバイス構成用補助) の「Boot Tasks」メニューにある「Set Network Configuration Strategy」メニューを使用して、適切なブート方法を選択できます。

1. ネットワークからブートする方法 (**RARP/bootparams** または **DHCP**) を決定します。
2つのネットワーク構成方法、RARP (Reverse Address Resolution Protocol) または DHCP (Dynamic Host Configuration Protocol) があります。デフォルトのネットワークブート方法は RARP に設定されています。ネットワークで利用できるブートサーバーによって、RARP または DHCP を選択できます。
PXE ネットワークブートは、DHCP でのみ使用できます。
2. **Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスク、Solaris 9 Installation CD または DVD** を適切なドライブに挿入します。
あるいは、システムまたはネットワークアダプタの BIOS 構成プログラムを使用して、PXE ネットワークブートを使用可能にします。
3. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。
「Solaris Device Configuration Assistant」画面が表示されます。
4. 「**F2_Continue**」(**F2** キー) を押して、デバイスを走査します。
デバイスの識別が行われ、識別されたデバイスを表示する画面が表示されます。
5. 「**F2_Continue**」(**F2** キー) を押して、ドライバを読み込みます。
ブート可能なドライバが読み込まれます。
「Boot Solaris」メニューが表示されます。
6. 「**F4_Boot Tasks**」(**F4** キー) を押します。
7. 「**Set Network Configuration Strategy**」を選択して、「**F2_Continue**」(**F2** キー) を押します。
8. 「**RARP**」または「**DHCP**」を選択して、「**F2_Continue**」(**F2** キー) を押します。
新しいネットワークブート方法を確認する画面が表示されます。

ここで選択したネットワークブート方法は、以降、このフロッピーディスクでブートしたときのデフォルトのネットワークブート方法として保存されます。

9. 「**F3_Back**」を押して、「**Boot Solaris**」メニューに戻ります。
10. 「**NET**」をブートデバイスとして選択します。そして、「**F2_Continue**」を押して、ネットワークデバイスをブートします。
「Solaris boot option」画面が表示されます。

▼ x86: 復元を目的としてシステムを停止する方法

1. 可能であれば、次のどちらかのコマンドを使用してシステムを停止します。
 - システムが動作している場合、スーパーユーザーになって `init 0` と入力してシステムを停止します。Type any key to continue プロンプトが表示されたら、任意のキーを押してシステムをリブートします。
 - システムが動作している場合、スーパーユーザーになって `init 6` と入力してシステムをリブートします。
2. マウスまたはキーボードからの入力にシステムが応答しない場合、リセットキーがあればそのキーを押してシステムをリブートします。あるいは、電源スイッチを使用してシステムをリブートします。

▼ x86: 復元を目的としてシステムをブートする方法

重要なシステムリソースを修復するには、次の手順に従ってシステムをブートします。この例は、Solaris 9 Installation CD またはネットワークからブートし、ルート (/) ファイルシステムをディスクにマウントし、`/etc/passwd` ファイルを修復する方法を示しています。

次の手順で使用されている変数 `devicename` は、修復するファイルシステムのデバイス名に置き換えてください。システムのデバイス名を調べる場合は、第 30 章を参照してください。

1. まず、システムのアボートキーシーケンスを使用して、システムを停止します。
システムのアボートキーシーケンスは、ルートのパスワードがわからない場合やシステムにログインできない場合に使用します。詳細については、213 ページの「x86: 復元を目的としてシステムを停止する方法」を参照してください。
2. **Solaris 9 Installation CD** または **DVD**、あるいはネットワークから、シングルユーザーモードでブートします。
 - a. **Device Configuration Assistant** (デバイス構成用補助) ブートフロッピーディスク、**Solaris 9 Installation CD** または **DVD** を適切なドライブに挿入します。
 - b. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

「Solaris Device Configuration Assistant」画面が表示されます。

- c. 「**F2_Continue**」(**F2** キー) を押します。
デバイスの識別が行われ、識別されたデバイスを表示する画面が表示されま
す。
 - d. 「**F2_Continue**」(**F2** キー) を押します。
ブート可能なドライバが読み込まれます。
「Boot Solaris」メニューが表示されます。
 - e. **CD-ROM** ドライブまたはネットワークデバイスを選択します。次
に「**F2_Continue**」(**F2** キー) を押します。
「Current Boot Parameters」メニューが表示されます。
 - f. プロンプトで **b -s** と入力して、**Enter** キーを押します。
数分後に、シングルユーザーモードの # プロンプトが表示されます。
3. 無効な **passwd** ファイルがあるルート (**/**) ファイルシステムをマウントします。

```
# mount /dev/dsk/devicename /a
```
 4. 新しくマウントした **etc** ディレクトリに移動します。

```
# cd /a/etc
```
 5. エディタを使用して **passwd** ファイルに必要な変更を加えます。

```
# vi filename
```
 6. ルート (**/**) ディレクトリに変更します。

```
# cd /
```
 7. **/a** ディレクトリのマウントを解除します。

```
# umount /a
```
 8. システムをリブートします。

```
# init 6
```
 9. システムが実行レベル **3** になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示さ
れます。

```
hostname console login:
```

x86: 例 — 復元を目的としてシステムをブートする

次の例は、ローカルの CD-ROM からブートした後で /etc/passwd ファイルを修復する方法を示しています。

Type any key to continue

SunOS Secondary Boot version 3.00

Solaris Intel Platform Edition Booting System

Running Configuration Assistant...

Autobooting from Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

.
.
.

Boot Solaris

Select one of the identified devices to boot the Solaris kernel and choose Continue.

To perform optional features, such as modifying the autoboot and property settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

```
[ ] NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ] DISK: (*) Target 0, QUANTUM FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[X] CD : Target 1:TOSHIBA CD-ROM XM-5602B 1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
```

F2_Continue F3_Back F4_Boot Tasks F6_Help

.
.
.

<<< Current Boot Parameters>>>

Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

Boot args: kernel/unix -r

Select the type of installation you want to perform:

```
1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start
```

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.

Select type of installation: **b -s**

```
.
.
.
# mount /dev/dsk/c0t0d0s0 /a
.
.
# cd /a/etc
# vi passwd
(無効なエントリを削除する)
# cd /
# umount /a
# init 6
```

x86: 例 — ルートのパスワードを忘れた場合にシステムをブートする

次の例では、ルートパスワードを忘れた場合にネットワークからブートして回復する方法について説明します。この例では、ブートサーバーがすでに有効になっているものとします。システムのリブート後に、必ず新しいルートパスワードを適用してください。

Type any key to continue

SunOS Secondary Boot version 3.00

Solaris Intel Platform Edition Booting System

Running Configuration Assistant...

Autobooting from Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

```
.
.
.
Boot Solaris
```

Select one of the identified devices to boot the Solaris kernel and

choose Continue.

To perform optional features, such as modifying the autoboot and property settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

```
[X] NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ] DISK: (*) Target 0, QUANTUM FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] CD : Target 1:TOSHIBA CD-ROM XM-5602B 1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
```

```
F2_Continue F3_Back F4_Boot Tasks F6_Help
```

```
.
.
.
```

```
<<< Current Boot Parameters>>>
```

```
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
```

```
Boot args: kernel/unix -r
```

Select the type of installation you want to perform:

```
1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start
```

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds, an interactive installation will be started.

Select type of installation: **b -s**

```
.
.
.
```

```
# mount /dev/dsk/c0t0d0s0 /a
```

```
.
.
.
```

```
# cd /a/etc
```

```
# vi shadow
```

(ルートの暗号化パスワード文字列を削除する)

```
# cd /
```

```
# umount /a
```

```
# init 6
```

▼ x86: カーネルデバッグ (kadb) を使ってシステムをブートする方法

1. **Type any key to continue** プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。
システムが停止している場合は、電源スイッチを押してシステムを起動します。
2. **b kadb** と入力して、カーネルデバッグをブートします。**Enter** キーを押します。
5秒以内に選択しない場合、システムは自動的に実行レベル3でブートします。
3. システムが実行レベル3になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

`hostname console login:`
4. **F1 + A** を押して、カーネルデバッグにアクセスできることを確認します。
カーネルデバッグがブートすると、`kadb[0]`: プロンプトが表示されます。

x86: 例 — カーネルデバッグ (kadb) を使ってシステムをブートする

```
Type any key to continue
.
.
.
      <<< Current Boot Parameters>>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                  to boot with defaults

      <<< timeout in 5 seconds>>>

Select (b)oot or (i)nterpreter: b kadb
.
.
.
naboo console login: (ログイン名とパスワードを入力する)
(F1 + A を押して、カーネルデバッグにアクセスできることを確認する)
```

x86: クラッシュダンプを強制してシステムをリブートする

場合によっては、障害追跡の目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できます。

システムのクラッシュダンプについては、『Solaris のシステム管理 (上級編)』の「システムクラッシュ情報の管理 (手順)」を参照してください。

▼ x86: クラッシュダンプを強制してシステムをリブートする方法

`kadb[0]`: プロンプトを表示して、クラッシュダンプを強制するには、カーネルデバッグオプション `kadb` を指定してシステムをブートする必要があります。

注 - カーネルデバッグ (`kadb`) に入るにはテキストモードでなければなりません。したがって、まずウィンドウシステムを終了してください。

1. **F1 + A** を押します。

```
kadb[0]:
```

```
kadb[0]: プロンプトが表示されます。
```

2. 次のコマンドを `kadb[0]`: プロンプトで入力します。

```
<F1 + A を押す>
```

```
kadb[0]: vfs_syncall/W ffffffff
```

```
kadb[0]: 0>eip
```

```
kadb[0]: :c
```

```
kadb[0]: :c
```

```
kadb[0]: :c
```

最初の `:c` を入力すると、システムはパニックを起こします。そこでもう一度 `:c` を入力します。システムは再度パニックを起こすため、3 度目の `:c` を入力し、クラッシュダンプを強制して、システムをリブートします。

クラッシュダンプがディスクに書き込まれた後、システムはリブートします。

3. コンソールログインプロンプトでログインして、システムがリブートされていることを確認します。

第 15 章

ブートプロセス (参照情報)

この章では、SPARC システム と x86 システムをブートするためのファームウェアについて説明します。また、各プラットフォームのブートプロセスの概要も説明します。

この章の内容は次のとおりです。

- 221 ページの「SPARC: ブート PROM」
- 222 ページの「SPARC: ブートプロセス」
- 222 ページの「x86: PC BIOS」
- 223 ページの「x86: ブートサブシステム」
- 228 ページの「x86: ブートプロセス」

システムのブート手順については、第 13 章または第 14 章を参照してください。

SPARC: ブート PROM

SPARC システムごとに、「モニター」と呼ばれるプログラムを格納している PROM (プログラム可能な読み取り専用メモリー) チップがあります。モニターは、Solaris のカーネルが使用される前に、システムの動作を制御します。システムをオンにすると、モニターはシステムのハードウェアやメモリーを検査するセルフテスト手順を実行します。エラーが検出されなければ、システムは、自動ブートプロセスを開始します。

注 - Solaris システムが動作する前に、以前のシステムの一部には PROM アップグレードが必要なものもあります。詳細については、ご購入先にお問い合わせください。

SPARC: ブートプロセス

次の表では、SPARC システムのブートプロセスについて説明します。

表 15-1 SPARC: ブートプロセスの説明

ブート段階	説明
ブート PROM	1. PROM は、システム識別情報を表示し、セルフテスト診断を実行してシステムのハードウェアとメモリーを検査する
ブートプログラム	2. PROM は一次ブートプログラム <code>bootblk</code> を読み込む。このプログラムの目的は、デフォルトのブートデバイスから二次ブートプログラム (<code>ufs</code> ファイルシステムにある) を読み込むことにある 3. <code>bootblk</code> プログラムは二次ブートプログラム <code>ufsboot</code> を検索して実行し、それをメモリーに読み込む 4. <code>ufsboot</code> プログラムが読み込まれた後、 <code>ufsboot</code> プログラムはカーネルを読み込む
カーネル初期設定	5. カーネルが自身を初期設定し、ファイルの読み込みに <code>ufsboot</code> を使ってモジュールのロードを開始する。カーネルはルート (/) ファイルシステムをマウントするのに十分なモジュールをロードすると、 <code>ufsboot</code> プログラムの対応づけを解除し、自身のリソースを使って動作を続ける 6. カーネルがユーザープロセスを作成し、 <code>/etc/inittab</code> ファイルを読み取って他のプロセスを起動する <code>/sbin/init</code> プロセスを起動する
init	7. <code>/sbin/init</code> プロセスが実行制御 (<code>rc</code>) スクリプトを起動し、このスクリプトは一連の他のスクリプトを実行する。それらのスクリプト (<code>/sbin/rc*</code>) はファイルシステムを検査、マウントし、さまざまなプロセスを実行して、システム保守作業を実行する

x86: PC BIOS

カーネルが起動される前は、システムは、PC 上のファームウェアインタフェースである、読み取り専用メモリー (ROM) の Basic Input/Output System (BIOS) によって制御されます。

ハードウェアアダプタにオンボード BIOS を搭載することができ、それによってデバイスの物理的特性を表示したり、デバイスにアクセスしたりすることができます。

起動シーケンスの間、PC の BIOS はアダプタの BIOS があるかどうか調べ、あれば、それぞれのアダプタ BIOS をロードして、実行します。アダプタの BIOS はそれぞれセルフテスト診断を実行して、デバイス情報を表示します。

x86: ブートサブシステム

Solaris のブート時に、ブートシステムについて次の選択を行うことができます。

- 一次ブートサブシステム (「Partition Boot」メニュー) — この最初のメニューは、複数のオペレーティング環境がディスク上にある場合に表示されます。このメニューで、インストールされたどのオペレーティング環境からブートするかを指定できます。デフォルトでは、アクティブなオペレーティング環境からブートします。

Solaris オペレーティング環境以外のオペレーティング環境からブートするように指定すると、次の 2 つのメニューは表示されません。

- 自動ブートプロセスへの割り込み — 自動ブートプロセスに割り込むと、Solaris Device Configuration Assistant (デバイス構成用補助) にアクセスできます。
Solaris Device Configuration Assistant (デバイス構成用補助) を使用すると、異なるブートデバイスから Solaris システムをブートする、新しいハードウェアを構成する、間違っって構成したハードウェアを構成し直すなど、デバイス関連やブート関連の作業を実行できます。
- 「Current Boot Parameters」メニュー — このメニューには 2 つの形式があります。1 つは、通常の Solaris ブート用で、もう 1 つは Solaris インストールブート用です。
 - 通常の「Current Boot Parameters」メニューでは、オプションを付けて Solaris システムをブートしたり、ブートインタプリタを実行したりできます。
 - インストール用の「Current Boot Parameters」メニューでは、インストールの種類を選択したり、ブートをカスタマイズしたりできます。

次の表に、主な x86 ブートインタフェースの目的を示します。以降の節では、各ブートインタフェースを詳細に説明し、例を示します。

表 15-2 x86: ブートサブシステム

ブートサブシステム	目的
一次ブートサブシステム	このメニューは、現在ブート中のディスクに複数のオペレーティング環境 (Solaris オペレーティング環境も含む) が含まれている場合に表示される

表 15-2 x86: ブートサブシステム (続き)

ブートサブシステム	目的
二次ブートサブシステム	Solaris リリースをブートするたびに表示される。自動ブートプロセスに割り込んで、Solaris Device Configuration Assistant を実行することを選択していない限り、Solaris リリースが自動的にブートされる
Solaris Device Configuration Assistant ブートフロッピーディスク	「Solaris Device Configuration Assistant」メニューを表示するには次の 2 つの方法がある 1. Solaris Device Configuration Assistant ブートフロッピーディスクまたは Solaris 9 Installation CD (CD-ROM ドライブからブートできるシステムのみ) を使用してシステムをブートする 2. インストールしたディスクから Solaris ソフトウェアをブートするときに自動ブートプロセスに割り込む
「Current Boot Parameters」メニュー	このメニューは、Solaris をディスク、CD-ROM、またはネットワークからブートするときに使用する。このメニューにはブートオプションが表示される

注 – Solaris Device Configuration Assistant ブートフロッピーディスクを作成する必要がある場合は http://soldc.sun.com/support/drivers/dca_diskettes を参照してください。

ブートプロセス中は、ブートサブシステムメニューは異なるデバイスとブートオプションを表示します。システムは何回かのタイムアウトの後で応答を受け付けなくなった場合、デフォルトの設定値を使って自動的にブートを継続します。ブートサブシステムメニューが表示されるたびに、ブートプロセスを停止することができます。また、自動的に継続させることもできます。

次の節では、各ブートサブシステムの画面の例を示します。

x86: Solaris リリースのブート

デバイスを識別する段階では、Solaris Device Configuration Assistant (デバイス構成用補助) は次の処理を実行します。

- システムにインストールされているデバイスを走査する
- 識別されたデバイスを表示する
- キーボードの選択やデバイスとそのリソースの編集などのオプションの作業を可能にする

ブートの段階では、Solaris Device Configuration Assistant (デバイス構成用補助) は次の処理を実行します。

- ブートするデバイスのリストを表示する。アスタリスク (*) が付いたデバイスがデフォルトのブートデバイスである
- 自動ブート設定やプロパティ設定の編集、ネットワーク構成方法の選択などのオプションの作業を可能にする

次の節では、デバイスを識別する段階で表示されるメニューの例を示します。デバイス出力は、各システム構成によって異なります。

x86: デバイス識別段階で表示される画面

Solaris Device Configuration Assistant (デバイス構成用補助) がシステム上のデバイスを識別するときに、いくつかの画面が表示されます。

x86: Device Configuration Assistant 画面

この画面は、Solaris Device Configuration Assistant (デバイス構成用補助) をブートするたびに表示されます。Solaris Device Configuration Assistant (デバイス構成用補助) はシステムがブートするたびに実行されますが、自動ブートプロセスではこのメニューは省略されます。

```
Solaris Device Configuration Assistant
```

```
The Solaris(TM) (Intel Platform Edition) Device Configuration Assistant
scans to identify system hardware, lists identified devices, and can
boot the Solaris software from a specified device. This program must be
used to install the Solaris operating environment, add a driver,
or change the hardware on the system.
```

```
> To perform a full scan to identify all system hardware, choose Continue.
```

```
> To diagnose possible full scan failures, choose Specific Scan.
```

```
> To add new or updated device drivers, choose Add Driver.
```

```
About navigation...
```

- The mouse cannot be used.
- If the keyboard does not have function keys or they do not respond, press ESC. The legend at the bottom of the screen will change to show the ESC keys to use for navigation.
- The F2 key performs the default action.

```
F2_Continue
```

```
F3_Specific Scan
```

```
F4_Add Driver
```

```
F6_Help
```

x86: Bus Enumeration 画面

この画面は、Solaris Device Configuration Assistant (デバイス構成用補助) が自動検出できるデバイスのハードウェア構成データを集めているときに一時的に表示されます。

Bus Enumeration

Determining bus types and gathering hardware configuration data ...

Please wait ...

x86: Scanning Devices 画面

この画面は、特別なドライバだけで検出できるデバイスを Solaris Device Configuration Assistant (デバイス構成用補助) が手動で走査しているときに表示されます。

Scanning Devices

The system is being scanned to identify system hardware.

If the scanning stalls, press the system's reset button. When the system reboots, choose Specific Scan or Help.

Scanning: Floppy disk controller

#####

| | | | | | |
0 20 40 60 80 100

Please wait ...

x86: Identified Devices 画面

この画面は、システムで識別されたデバイスを表示します。この画面からは、「Boot Solaris」メニューに移動でき、また、キーボード構成の設定、デバイスの表示や編集、シリアルコンソールの設定、構成の保存や削除など、オプションのデバイス作業も実行できます。

Identified Devices

The following devices have been identified on this system. To identify devices not on this list or to modify device characteristics, such as keyboard configuration, choose Device Tasks. Platform types may be included in this list.

ISA: Floppy disk controller

ISA: Motherboard

ISA: PnP bios: 16550-compatible serial controller

ISA: PnP bios: 16550-compatible serial controller

ISA: PnP bios: Mouse controller

ISA: PnP bios: Parallel port

ISA: System keyboard (US-English)

PCI: Bus Mastering IDE controller

PCI: Universal Serial Bus

```
PCI: VGA compatible display adapter
F2_Continue  F3_Back  F4_Device Tasks  F6_Help
```

x86: ブート段階で表示されるメニュー

この段階では、システムをブートする方法を指定できます。

x86: Boot Solaris メニュー

「Boot Solaris」メニューからは、Solaris リリースをブートするデバイスを選択できます。また、autoboot やプロパティ設定の表示や編集など、オプションの作業も実行できます。ブートデバイスを選択し、「Continue」を選択すると、Solaris カーネルがブートを開始します。

```
Boot Solaris

Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[X] DISK: (*) Target 0:QUANTUM FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] CD : Target 1:TOSHIBA CD-ROM XM-5602B 1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1

F2_Continue  F3_Back  F4_Boot Tasks  F6_Help
```

x86: Current Boot Parameters メニュー

このメニューは、ローカルディスクから Solaris リリースをブートするたびに表示されます。デフォルトの Solaris カーネルからブートしたい場合は、5 秒間のタイムアウトが経過するのを待ちます。別のオプションでブートする場合は、タイムアウト期間が経過する前に、適切なオプションを選択します。

```
<<< Current Boot Parameters>>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
```

```
Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                          to enter boot interpreter
or      <ENTER>                             to boot with defaults
```

```
<<< timeout in 5 seconds>>>
```

```
Select (b)oot or (i)nterpreter:
```

x86: ブートプロセス

次の表では、x86 システムのブートプロセスについて説明します。

表 15-3 x86: ブートプロセスの説明

ブート段階	説明
BIOS	<p>1. システムの電源をオンにすると、BIOS がセルフテスト診断を実行してシステムのハードウェアとメモリーを検査する。エラーが検出されなければ、システムは自動的にブートを開始する。エラーが検出されると、復元オプションを示すエラーメッセージが表示される</p> <p>追加のハードウェアデバイスの BIOS がここで実行される</p> <p>2. BIOS ブートプログラムが、ブートデバイスの最初の物理セクターを読もうとする。ブートデバイス上のこの最初のディスクセクターにはマスターブートレコード mboot が格納されており、このレコードが読み込まれて実行される。mboot ファイルが見つからなかった場合は、エラーメッセージが表示される</p>
ブートプログラム	<p>3. マスターブートレコード mboot には、アクティブなパーティションと、Solaris ブートプログラム pboot の位置を見つけるのに必要なディスク情報が格納されており、mboot は、pboot を読み込んで実行する</p> <p>4. Solaris ブートプログラム pboot が一次ブートプログラム bootblk を読み込む。このプログラムは、ufs ファイルシステムにある二次ブートプログラムを読み込む</p> <p>5. ブート可能なパーティションが複数ある場合、bootblk は fdisk テーブルを読んでデフォルトのブートパーティションを探し、利用可能なパーティションのメニューを作成して表示する。ブートするパーティションを選択する際に、30 秒経過するとタイムアウトになる。これは、ブート可能なパーティションが複数ある場合にのみ発生する</p>

表 15-3 x86: ブートプロセスの説明 (続き)

ブート段階	説明
	<p>6. bootblk はルート (/) ファイルシステムで二次ブートプログラム boot.bin または ufsboot を探し、実行する。Solaris Device Configuration Assistant (デバイス構成用補助) を起動するときは、5 秒以内に自動ブートに割り込みをかける</p> <p>7. 二次ブートプログラム boot.bin または ufsboot が /etc/bootrc スクリプトを実行するコマンドインタプリタを起動する。ここでシステムのブート方法を選択するメニューが表示される。デフォルトでは、カーネルが読み込まれ、実行される。ブートオプションを指定したり、ブートインタプリタを起動したりするときは、5 秒以内に行う</p>
カーネル初期設定	<p>8. カーネルは二次ブートプログラム boot.bin または ufsboot を使用してファイルを読み込み、カーネル自身を初期化して、モジュールの読み込みを開始する。カーネルはルート (/) ファイルシステムをマウントするのに十分なモジュールを読み込むと、二次ブートプログラムの対応づけを解除し、自身のリソースを使って動作を続ける</p> <p>9. カーネルがユーザープロセスを作成し、/etc/inittab ファイルを読み取って他のプロセスを起動する /sbin/init プロセスを起動する</p>
init	<p>10. /sbin/init プロセスは実行制御 (rc) スクリプトを開始し、このスクリプトが一連の他のスクリプトを実行する。それらのスクリプト (/sbin/rc*) はファイルシステムを検査、マウントし、さまざまなプロセスを実行して、システム保守作業を実行する</p>

第 16 章

リムーバブルメディアの管理

以下の各章では、リムーバブルメディアの管理について説明します。

第 17 章	コマンド行からリムーバブルメディアを管理する方法についての概要を説明します。
第 18 章	コマンド行からリムーバブルメディアにアクセスする手順について説明します。
第 19 章	コマンド行からリムーバブルメディアをフォーマットする手順について説明します。
第 20 章	データ CD やオーディオ CD を作成する手順について説明します。

第 17 章

リムーバブルメディアの管理 (概要)

この章では、Solaris 環境でリムーバブルメディアを管理するための一般的なガイドラインについて説明します。

この章の内容は次のとおりです。

- 233 ページの「リムーバブルメディアの管理における新機能」
- 234 ページの「リムーバブルメディアの管理についての参照先」
- 234 ページの「リムーバブルメディアの機能と利点」
- 235 ページの「自動マウントと手動によるマウントの比較」
- 236 ページの「ボリューム管理を使って行える操作」

リムーバブルメディアの管理における新機能

ボリューム管理の機能が拡張されて、リムーバブルメディアが完全にサポートされるようになりました。この拡張により、DVD-ROM、Iomega と USB (Universal Serial Bus) の Zip ドライブと Jaz ドライブ、CD-ROM、およびフロッピーディスクは挿入時にマウントされ、読み取ることができるようになりました。

共通デスクトップ環境 (CDE) のボリューム管理と Solaris のコマンド行の両方を使用すると、リムーバブルメディアを完全に管理することができます。

ボリューム管理の機能の向上によって、次のことが可能になりました。

- 新しい `rmformat` コマンドによる、リムーバブルメディアのフォーマット、ラベル付け、および読み取りまたは書き込みソフトウェア保護の設定。このコマンドは以前にリムーバブルメディアのフォーマットに使用していた `fdformat` コマンドに代わって使用するものです。
- `mkfs_pcfs` コマンドと `fsck_pcfs` コマンドを使用してのリムーバブルメディアへの PCFS ファイルシステムの作成と検証。

- SPARC システム上のリムーバブルメディアへの `fdisk` パーティションと PCFS ファイルシステムの作成。x86 システムへのデータ転送を容易にします。

リムーバブルメディアを使用する場合、次のガイドラインがあります。

- DVD メディア間でデータを転送するときは、UDFS と PCFS を使用します。
- 書き換え可能メディア (UFS ファイルシステムを持つ PCMCIA メモリーカードやフロッピーディスクなど) 間でファイルを転送するときは、`tar` または `cpio` コマンドを使用します。SPARC システム上に作成された UFS ファイルシステムは、x86 システム上に作成された PCMCIA またはフロッピーディスク上の UFS ファイルシステムとは異なります。
- Jaz ドライブや Zip ドライブ、またはフロッピーディスク上の重要なファイルを保護するには、書き込み保護を設定します。Iomega メディアにはパスワードを適用します。

リムーバブルメディアの管理についての参照先

リムーバブルメディアを管理する手順については、次を参照してください。

リムーバブルメディアの管理作業	参照先
リムーバブルメディアへのアクセス	第 18 章
リムーバブルメディアのフォーマット	第 19 章
データ CD とオーディオ CD の作成	第 20 章

共通デスクトップ環境のファイルマネージャでリムーバブルメディアを使用する方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』を参照してください。

リムーバブルメディアの機能と利点

Solaris 環境には、ユーザーとソフトウェア開発者用に、リムーバブルメディアを扱うための標準インタフェースが用意されています。ボリューム管理と呼ばれるこのインタフェースには、主に次の 3 つの利点があります。

- リムーバブルメディアを自動的にマウントすることによって、操作を簡単にします。手動によるマウントと自動マウントの比較については、次の節を参照してください。
- スーパーユーザーでなくても、リムーバブルメディアにアクセスできるようになります。
- ネットワーク上の他のシステムがローカルシステム上のリムーバブルメディアに自動的にアクセスできるようになります。詳細については、第 18 章を参照してください。

自動マウントと手動によるマウントの比較

次の表は、リムーバブルメディアの手動によるマウント (ボリューム管理を使用しない場合) と自動マウント (ボリューム管理を使用する場合) に関する手順を比較したものです。

表 17-1 手動によるマウントと自動マウントの比較

手順	手動によるマウント	自動マウント
1	メディアを挿入する	メディアを挿入する
2	スーパーユーザーになる	フロッピーディスクの場合は、 <code>volcheck</code> コマンドを使用する
3	メディアデバイスの位置を確認する	ボリュームマネージャ (<code>volfd</code>) は、リムーバブルメディアを手動でマウントするのに必要な作業のほとんどを自動的に実行する
4	マウントポイントを作成する	
5	マウントポイントが現在のディレクトリではないことを確認する	
6	<code>mount</code> に適切なオプションを付けて、デバイスをマウントする	
7	スーパーユーザーを終了する	
8	メディア上のファイルを操作する	メディア上のファイルを操作する
9	スーパーユーザーになる	
10	メディアデバイスのマウントを解除する	

表 17-1 手動によるマウントと自動マウントの比較 (続き)

手順	手動によるマウント	自動マウント
11	メディアを取り出す	メディアを取り出す
12	スーパーユーザーを終了する	

ボリューム管理を使って行える操作

ボリューム管理を使用すると、手動によるマウントの場合と同様にリムーバブルメディアにアクセスできますが、その操作ははるかに容易になり、スーパーユーザーになる必要もありません。リムーバブルメディアは、操作しやすいように、覚えやすい位置にマウントされます。

表 17-2 ボリュームマネージャによって管理されたリムーバブルメディア上のデータにアクセスする方法

アクセス	操作	ファイルを検索する場所
最初のフロッピーディスク上のファイル	フロッピーディスクを挿入して、 <code>volcheck</code> と入力する	<code>/floppy</code>
最初のリムーバブルハードディスク上のファイル	リムーバブルハードディスクを挿入して、 <code>volcheck</code> と入力する	<code>/rmdisk/jaz0</code> または <code>/rmdisk/zip0</code>
最初の CD 上のファイル	CD を挿入して、数秒間待つ	<code>/cdrom/volume-name</code>
最初の DVD 上のファイル	DVD を挿入して、数秒間待つ	<code>/dvd/volume-name</code>
最初の PCMCIA 上のファイル	PCMCIA を挿入して、数秒間待つ	<code>/pcmem/pcmem0</code>

システムに複数の種類のリムーバブルメディアデバイスがある場合は、そのアクセスポイントについて、次の表を参照してください。

表 17-3 リムーバブルメディアにアクセスする場所

メディアデバイス	ファイルシステムにアクセスするためのパス	raw データにアクセスするための場所
最初のフロッピーディスクドライブ	<code>/floppy/floppy0</code>	<code>/vol/dev/aliases/floppy0</code>

表 17-3 リムーバブルメディアにアクセスする場所 (続き)

メディアデバイス	ファイルシステムにアクセスするためのパス	raw データにアクセスするための場所
2 番目のフロッピーディスクドライブ	/floppy/floppy1	/vol/dev/aliases/floppy1
最初の CD-ROM ドライブ	/cdrom/cdrom0	/vol/dev/aliases/cdrom0
2 番目の CD-ROM ドライブ	/cdrom/cdrom1	/vol/dev/aliases/cdrom1
最初のリムーバブルハードディスク	/rmdisk/jaz0	/vol/dev/aliases/jaz0
	/rmdisk/zip0	/vol/dev/aliases/zip0
最初の PCMCIA ドライブ	/pcmem/pcmem0	/vol/dev/aliases/pcmem0

第 18 章

リムーバブルメディアへのアクセス (手順)

この章では、Solaris 環境でコマンド行からリムーバブルメディアにアクセスする方法について説明します。

リムーバブルメディアへのアクセスに関連する手順については、次の節を参照してください。

- 239 ページの「リムーバブルメディアへのアクセス (作業マップ)」
- 248 ページの「リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)」

リムーバブルメディアの概要については、第 17 章を参照してください。

リムーバブルメディアへのアクセス (作業マップ)

作業	説明	参照先
1. (省略可能) リムーバブルメディアドライブを追加する	必要に応じて、リムーバブルメディアドライブをシステムに追加する	243 ページの「新しいリムーバブルメディアドライブを追加する方法」
2. (省略可能) リムーバブルメディアにアクセスするときにボリューム管理 (vol1d) を使用するかどうかを決める	デフォルトでは、ボリューム管理 (vol1d) は実行される。リムーバブルメディアにアクセスするときにボリューム管理を使用するかどうかを決める	243 ページの「ボリューム管理 (vol1d) の終了と起動」

作業	説明	参照先
3. リムーバブルメディアにアクセスする	ボリューム管理を使用して/使用しないで、各種のリムーバブルメディアにアクセスする	244 ページの「リムーバブルメディア上の情報にアクセスする方法」
4. (省略可能) ファイルまたはディレクトリをコピーする	ファイルまたはディレクトリを、ファイルシステムの別の場所からコピーするように、リムーバブルメディアからコピーする	245 ページの「リムーバブルメディア上の情報をコピーする方法」
5. (省略可能) 音楽用 CD/DVD を再生するようにシステムを設定する	音楽用 CD/DVD を再生するようにシステムを設定できるが、再生には Sun 以外のソフトウェアが必要になる	245 ページの「音楽用 CD/DVD を再生する方法」
6. メディアが使用中であるかどうかを調べる	メディアを取り出す前に、それが使用中であるかどうかを調べる	247 ページの「リムーバブルメディアが使用中かどうかを調べる方法」
7. メディアを取り出す	作業が終了したら、ドライブからメディアを取り出す	247 ページの「リムーバブルメディアを取り出す方法」

リムーバブルメディアへのアクセス (概要)

リムーバブルメディア上の情報にアクセスするには、ボリュームマネージャを使用する方法と使用しない方法があります。CDE のファイルマネージャを使用してリムーバブルメディア上の情報にアクセスする方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の「ファイル・マネージャでのリムーバブル・メディアの使い方」を参照してください。

Solaris 8 6/00 リリースから、ボリュームマネージャ (vol1d) ですべてのリムーバブルメディアデバイスを動的に管理できるようになりました。これに伴い、`/dev/rdsk/cntndnsn` や `/dev/dsk/cntndnsn` などのデバイス名でのリムーバブルメディアへのアクセスは、できなくなります。

リムーバブルメディア名の使用

リムーバブルメディアには、さまざまな名前前でアクセスできます。次の表に、ボリューム管理を使用して/使用しないでアクセスできるさまざまなメディア名を示します。

表 18-1 リムーバブルメディア名

メディア	ボリューム管理デバイスの名前	ボリューム管理デバイスの別名	デバイス名
最初のフロッピーディスクドライブ	/floppy	/vol/dev/aliases/floppy0	/dev/rdiskette /vol/dev/rdiskette0/ <i>volume-name</i>
	/cdrom0	/vol/dev/aliases/cdrom0	/vol/dev/rdsk/cntn[dn]/
	/cdrom1 /cdrom2	/vol/dev/aliases/cdrom1 /vol/dev/aliases/cdrom2	<i>volume-name</i>
最初、2番目、3番目の Jaz (ジャズ) ドライブ	/rmdisk/jaz0	/vol/dev/aliases/jaz0	/vol/dev/rdsk/cntndn/
	/rmdisk/jaz1	/vol/dev/aliases/jaz1	<i>volume-name</i>
	/rmdisk/jaz2	/vol/dev/aliases/jaz2	
最初、2番目、3番目の Zip ドライブ	/rmdisk/zip0	/vol/dev/aliases/zip0	/vol/dev/rdsk/cntndn/
	/rmdisk/zip1	/vol/dev/aliases/zip1	<i>volume-name</i>
	/rmdisk/zip2	/vol/dev/aliases/zip2	
最初、2番目、3番目の PCMCIA ドライブ	/pcmem/pcmem0	/vol/dev/aliases/pcmem0	/vol/dev/rdsk/cntndn/
	/pcmem/pcmem1	/vol/dev/aliases/pcmem1	<i>volume-name</i>
	/pcmem/pcmem2	/vol/dev/aliases/pcmem2	

特定の Solaris コマンドで使用するリムーバブルメディア名を確認する場合は、次の表を使用してください。

Solaris コマンド	デバイス名	使用例
ls, more, vi	/floppy	ls /floppy/myfiles/
	/cdrom	more /cdrom/myfiles/filea
	/rmdisk/zip0	
	/rmdisk/jaz0	
	/pcmem/pcmem0	
fsck, newfs, mkfs	/vol/dev/aliases/floppy0	newfs /vol/dev/aliases/floppy0
	/vol/dev/rdsk/cntndn	mkfs -F udfs/vol/dev/rdsk/cntndn

リムーバブルメディア上のデータにアクセスするためのガイドライン

ほとんどの CD と DVD は、ISO 9660 標準でフォーマットされています。このフォーマットには移植性があるため、ほとんどの CD と DVD をボリューム管理によってマウントできます。ただし、UFS ファイルシステムを持つ CD と DVD にはアーキテクチャ間の移植性がないため、設計されたときのアーキテクチャで使用する必要があります。

たとえば、SPARC システム用にフォーマットされた UFS ファイルシステムを持つ CD と DVD は x86 システムでは認識されません。同様に、x86 システム用にフォーマットされた UFS CD は、ボリューム管理によって SPARC システム上にマウントすることはできません。同じ制限が、フロッピーディスクにもあてはまります。実際には、同じビット構造を共有するアーキテクチャもあります。このため、場合によっては、あるアーキテクチャに固有の UFS フォーマットが別のアーキテクチャによって認識されることもあります。UFS ファイルシステム構造はこのような互換性を保証するように設計されたものではありません。

さまざまなフォーマットに対応するために、CD または DVD は、実際にはハードディスクのパーティションに似たスライスに分割されます。9660 部分は移植可能ですが、UFS 部分はアーキテクチャに固有です。CD または DVD のマウントで問題が生じた場合、特にそれがインストール用 CD または DVD の場合は、その UFS ファイルシステムが、使用しているシステムのアーキテクチャに適しているかどうかを CD または DVD のラベルで確認してください。

Jaz ドライブおよび Zip ドライブへのアクセス

次の状況に応じて、Jaz ドライブや Zip ドライブへのアクセス方法が以前の Solaris リリースから変更されたかどうかを判断できます。

- Solaris 8 6/00 リリースから Solaris 9 リリースへアップグレードする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることができます。
- 新規に Solaris 9 リリースをインストールする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることはできません。

以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスする場合は、次の手順に従ってください。

1. /etc/vold.conf ファイル内の次の行でテキストの初めに # マークを挿入し、コメント扱いにします。

```
# use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

2. システムをリブートします。

▼ 新しいリムーバブルメディアドライブを追加する方法

新しいリムーバブルメディアドライブを追加する場合は、`/reconfigure` ファイルを作成し、システムをリブートして、ボリューム管理に新しいメディアドライブを認識させる必要があります。

1. スーパーユーザーになります。
2. `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```
3. システムを実行レベル **0** にします。

```
# init 0
```
4. システムの電源を切ります。
5. 新しいリムーバブルメディアドライブを接続します。
詳細な手順については、該当するハードウェアのマニュアルを参照してください。
6. システムの電源を入れます。
システムが自動的にマルチユーザーモードになります。

ボリューム管理 (vold) の終了と起動

場合によっては、ボリューム管理を使用しないで、メディアを管理した方がよいことがあります。この節では、ボリューム管理をいったん終了し、後で再起動する方法について説明します。

▼ ボリューム管理 (vold) を終了させる方法

1. メディアが使用中でないことを確認します。
メディアを使用中のすべてのユーザーを確認できたか保証がない場合は、247 ページの「リムーバブルメディアが使用中かどうかを調べる方法」の手順に従って `fuser` コマンドを使用してください。
2. スーパーユーザーになります。
3. `volmgt stop` コマンドを入力します。

```
# /etc/init.d/volmgt stop  
#
```

▼ ボリューム管理 (vold) を再起動する方法

1. スーパーユーザーになります。
2. **volmgt start** コマンドを入力します。

```
# /etc/init.d/volmgt start
volume management starting.
```

▼ リムーバブルメディア上の情報にアクセスする方法

1. メディアを挿入します。
数秒後にメディアがマウントされます。
2. ドライブにメディアが入っているかどうかを確認します。

```
% volcheck
コマンド行インタフェースを使用し、適切なデバイス名を指定して情報にアクセス
します。デバイス名については、表 18-1 を参照してください。
```

3. メディアの内容をリスト表示します。

```
% ls /media
```

例 — リムーバブルメディア上の情報にアクセスする

次のように入力して、フロッピーディスク上の情報にアクセスします。

```
$ volcheck
$ ls /floppy
myfile
```

次のように入力して、Jaz ドライブ上の情報にアクセスします。

```
$ volcheck
$ ls /rmdisk
jaz0/          jaz1/
```

次のように入力して、CD-ROM 上の情報にアクセスします。

```
$ volcheck
$ ls /cdrom
solaris_9_sparc/
```

次のように入力して、CD-ROM 上のシンボリックリンクを表示します。

```
$ ls -lL /cdrom/cdrom0
total 166
drwxr-xr-x  4 root    root          2048 Jul 21 05:18 MU
```

```
drwxr-xr-x  4 root    root          2048 Jul 21 05:18 Solaris_7_MU3
-rwxr-xr-x  1 root    root          30952 Jul 21 05:18 backout_mu
-rwxr-xr-x  1 root    root          49604 Jul 21 05:18 install_mu
```

次のように入力して、PCMCIA メモリーカード上の情報にアクセスします。

```
$ ls /pcmem/pcmem0
pcmem0 myfiles
```

▼ リムーバブルメディア上の情報をコピーする方法

リムーバブルメディア上のファイルやディレクトリへのアクセスは、他のファイルシステムの場合とまったく同じように行えます。ただし、所有権とアクセス権については重大な制限があります。

たとえば、あるユーザーが、CD 上のファイルを自分のファイルシステムにコピーした場合、そのユーザーはファイルの所有者になりますが、書き込み権は与えられません (CD 上のファイルには書き込み権がないため)。アクセス権は各自で変更しなければなりません。

1. メディアがマウントされていることを確認します。

```
$ ls /media
```

ls コマンドは、マウントされたメディアの内容を表示します。内容が表示されない場合は、244 ページの「リムーバブルメディア上の情報にアクセスする方法」を参照してください。

2. (省略可能) ファイルまたはディレクトリをコピーします。

たとえば、CD の場合は次のように入力します。

```
$ cp /cdrom/sol_8_u3_sparc_2/Solaris_8/EA/products/Live*/README*
$ ls -l
-r--r--r--  1 pmorph  users          3002 May  9 08:09 README_Live_Upgrade
```

たとえば、PCMCIA メモリーカードの場合は次のように入力します。

```
$ cp /pcmem/pcmem0/readme2.doc .
$ cp -r /pcmem/pcmem0/morefiles .
```

▼ 音楽用 CD/DVD を再生する方法

Solaris リリースを実行しているシステムに接続されたリムーバブルメディアドライブから音楽用のメディアを再生する場合は、xmcd などのパブリックドメインソフトウェアにアクセスする必要があります。xmcd は、次の場所から入手できます。

- <http://www.ibiblio.org/tkan/xmcd>

このサイトには、xmcd ソフトウェアの更新情報が頻繁に追加され、Sun Blade™ システムなどの Sun 製の新しいハードウェアで再生するための新バージョンの xmcd も含まれています。

- http://www.sun.com/software/solaris/freeware/pkggs_download.html

xmcd ソフトウェアを CDDA (CD Digital Audio) サポートと併用して音楽用のメディアを再生するときは、次のことに注意してください。

- Sun Blade™ システムではバージョン 3.1 以上の xmcd を使用してください (このバージョンには CDDA サポート機能があるため)。CDDA は Sun Blade システムで CD の応答待ちをするために必要です。
- xmcd を起動し、オプションボタン (ハンマーとねじ回しが表示されたボタン) をクリックして、「CDDA playback」をクリックすることで CDDA を使用可能にしてください。
- CDDA が使用可能になると、オーディオはオーディオデバイスに送られるよう指示されるため、ヘッドホンや外部スピーカーはメディアドライブそのものではなくオーディオデバイスに接続する必要があります。
- CDDA は他のマシンでも使用可能にできます。Sun Blade システムでメディアを再生するためには、CDDA を使用可能にする必要があります。

次の問題にも考慮してください。

- CD-ROM とオーディオデバイスが内部接続されていないシステムで xmcd を標準の再生機能と組み合わせて使用する場合は、ヘッドホンを CD-ROM ドライブのヘッドホンポートに挿入する必要があります。
- CD-ROM とオーディオデバイスが内部接続されているシステムで xmcd を標準の再生機能と組み合わせて使用する場合は、次のどちらの方法も可能です。
 1. ヘッドホンを CD-ROM ドライブのヘッドホンポートに挿入する。
 2. ヘッドホンをオーディオデバイスのヘッドホンポートに挿入する。

ただし、2 番目を選択した場合は、次の作業を行う必要があります。

- 内蔵 CD を入力デバイスとして選択する。
- モニターボリュームが 0 以外の値になっていることを確認する。

これらの作業はどちらも、sdtaudiocontrol のレコードパネルから実行することができます。

xmcd ソフトウェアをインストールすると、音楽用 CD を CD-ROM ドライブに挿入し、xmcd のコントロールパネルを起動するだけで再生できます。

1. **xmcd** ソフトウェアをインストールします。
2. メディアをメディアドライブに挿入します。
3. **xmcd** コマンドを起動します。

```
% ./xmcd &
```

▼ リムーバブルメディアが使用中かどうかを調べる方法

1. スーパーユーザーになります。
2. メディアにアクセスしているプロセスを特定します。

```
# fuser -u [-k] /media
```

-u	メディアを使用しているプロセスなどを表示する
-k	メディアにアクセスしているプロセスを強制終了する

fuser コマンドの使用方法については、fuser(1M) のマニュアルページを参照してください。

3. (省略可能) メディアにアクセスしているプロセスを強制終了します。

```
# fuser -u -k /media
```



注意 - メディアにアクセスしているプロセスの強制終了は、緊急の場合にのみ行います。

4. プロセスが終了していることを確認します。

```
# pgrep process-ID
```

例 — メディアが使用中かどうかを調べる

次の例は、プロセス 26230c (所有者 ripley) が /cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0 ディレクトリにアクセスしていることを示しています。

```
# fuser -u /cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0  
/cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0: 26230c(ripley)
```

▼ リムーバブルメディアを取り出す方法

1. メディアが使用中でないことを確認します。

シェルまたはアプリケーションがメディア上のファイルまたはディレクトリのいずれかにアクセスしている場合、メディアは「使用中」であることを忘れないでください。CD を使用しているシェルやアプリケーションなどをすべて検出したかどうか分からない (デスクトップツールの背後に隠れているシェルがアクセスしている

可能性がある) 場合は、247 ページの「リムーバブルメディアが使用中かどうかを調べる方法」の手順に従って `fuser` コマンドを使用してください。

2. メディアを取り出します。

```
# eject media
```

たとえば、CD の場合は次のように入力します。

```
# eject cdrom
```

たとえば、PCMCIA メモリーカードの場合は次のように入力します。

```
# eject pcmem0
```

リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)

次の表に、リモートシステム上のリムーバブルメディアへのアクセスに必要な作業を示します。

作業	説明	参照先
1. ローカルのメディアをリモートシステムで使用できるようにする	必要に応じて、リムーバブルメディアドライブをシステムに追加する	248 ページの「ローカルのメディアを他のシステムで使用可能にする方法」
2. リモートシステム上のリムーバブルメディアにアクセスする	メディアをドライブに挿入する	244 ページの「リムーバブルメディア上の情報にアクセスする方法」

▼ ローカルのメディアを他のシステムで使用可能にする方法

システムのメディアドライブを共有するようにシステムを設定すると、そのドライブに読み込まれているメディアが他のシステムでも使用できるようになります。(これは音楽用の CD には当てはまりません。) リムーバブルメディアドライブを共有すると、252 ページの「リモートシステム上のリムーバブルメディアにアクセスする方法」の手順に従ってそのドライブをマウントするだけで、ドライブにロードされているメディアを他のシステムでも使用できます。

1. スーパーユーザーになります。
2. NFS デーモン (`nfsd`) が実行されているかどうかを調べます。


```
# ps -ef | grep nfsd
root 14533    1 17 10:46:55 ?        0:00 /usr/lib/nfs/nfsd -a 16
root 14656   289  7 14:06:02 pts/3 0:00 grep nfsd
```

デーモンが実行されている場合、`/usr/lib/nfs/nfsd` の行は、上記のように表示されます。デーモンが実行されていない場合は、`grep nfsd` の行だけが表示されます。

3. `nfsd` の状態を確認し、次のいずれかの手順を選択します。

- a. `nfsd` が実行されている場合は、手順 8 に進む。

- b. `nfsd` が実行されていない場合は、手順 4 に進む。

4. `nfsd` が共有するための `dummy` ディレクトリを作成します。

```
# mkdir /dummy-dir
```

`dummy-dir`

たとえば、`dummy` などの任意のディレクトリ名にすることができ。このディレクトリには、ファイルは含まれない。これは、NFS デーモンを「呼び起こして」、共有されたメディアドライブを認識させることを目的としている。

5. 次のエントリを `/etc/dfs/dfstab` ファイルに追加します。

```
share -F nfs -o ro [-d comment] /dummy-dir
```

NFS デーモンを起動すると、このエントリ「呼び起こし」が参照され、共有されたメディアドライブが認識されます。コメント (`-d` が前に付く) はオプションです。

6. NFS デーモンを起動します。

```
# /etc/init.d/nfs.server start
```

7. NFS デーモンが実際に実行されていることを確認します。

```
# ps -ef | grep nfsd
root 14533    1 17 10:46:55 ?        0:00 /usr/lib/nfs/nfsd -a 16
root 14656   289  7 14:06:02 pts/3 0:00 /grep nfsd
```

8. 現在ドライブにあるメディアを取り出します。

```
# eject media
```

9. `root` の書き込み権を `/etc/rmmount.conf` ファイルに割り当てます。

```
# chmod 644 /etc/rmmount.conf
```

10. 次の行を `/etc/rmmount.conf` ファイルに追加します。

```
# File System Sharing
share media*
```

上記の行によって、システムの CD-ROM ドライブにロードされる CD が共有されます。ただし、share (1M) のマニュアルページに説明されているように、特定の CD (複数も可) に共有範囲を限定することができます。

11. `/etc/rmmount.conf` ファイルから書き込み権を削除します。

```
# chmod 444 /etc/rmmount.conf
```

この手順によって、ファイルはそのデフォルトのアクセス権に戻ります。

12. メディアを読み込みます。

ここで読み込むメディアはすべて、他のシステムで使用できるようになります。必ずドライブのランプの点滅が消えるまで待って、この作業を確認するようにしてください。

メディアにアクセスするには、リモートユーザーは、252 ページの「リモートシステム上のリムーバブルメディアにアクセスする方法」の手順に従って、名前によりそのメディアをマウントする必要があります。

13. `share` コマンドを使用して、メディアが実際に他のシステムで使用できるかどうかを確認します。

メディアが使用可能な場合は、その共有設定が表示されます。共有される dummy ディレクトリも表示されます。

```
# share
- /dummy ro "dummy dir to wake up NFS daemon"
- /cdrom/sol_9_sparc ro ""
```

例 — ローカル CD を他のシステムで使用可能にする

次の例は、ローカルの CD をネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```
# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118     1  0 08:24:39 ?          0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
share -F nfs -o ro /dummy
# eject cdrom0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(次の行をファイルシステム共有セクションに追加する)
share cdrom*
# chmod 444 /etc/rmmount.conf
(CD をロードする)
# share
- /dummy ro ""
- /cdrom/sol_9_sparc/s5 ro ""
- /cdrom/sol_9_sparc/s4 ro ""
- /cdrom/sol_9_sparc/s3 ro ""
```

```

-           /cdrom/sol_9_sparc/s2  ro  ""
-           /cdrom/sol_9_sparc/s1  ro  ""
-           /cdrom/sol_9_sparc/s0  ro  ""
#

```

例 — ローカルのフロッピーディスクを他のシステムで使用可能にする

次の例は、ローカルのフロッピーディスクをネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```

# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118    1  0 08:24:39 ?          0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
share -F nfs -o ro /dummy
# eject floppy0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(次の行をファイルシステム共有セクションに追加する)
share floppy*
# chmod 444 /etc/rmmount.conf
(フロッピーディスクをロードする)
# volcheck -v
media was found
# share
-           /dummy  ro  ""
-           /floppy/myfiles  rw  ""

```

例 — ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする

次の例は、ローカルの PCMCIA メモリーカードをネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```

# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118    1  0 08:24:39 ?          0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
share -F nfs -o ro /dummy
# eject pcmem0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(次の行をファイルシステム共有セクションに追加する)
share floppy*
# chmod 444 /etc/rmmount.conf

```

```
(PCMCIA メモリーカードをロードする)
# volcheck -v
media was found
# share
-                /dummy    ro    ""
-                /pcmem/myfiles  rw    ""
```

▼ リモートシステム上のリムーバブルメディアにアクセスする方法

248 ページの「ローカルのメディアを他のシステムで使用可能にする方法」の指示に従って、リモートシステムがそのシステム上のメディアを共有している場合は、手動でそのメディアをファイルシステムにマウントすれば、リモートシステム上のメディアにアクセスすることができます。

1. マウントポイントとして使用する既存のディレクトリを指定するか、作成します。

```
$ mkdir directory
```

directory は、他のシステムのメディアのマウントポイントとして作成するディレクトリの名前です。

2. マウントするメディアの名前を確認します。

```
$ showmount -e system-name
export list for system-name:
/cdrom/sol_9_sparc (everyone)
```

3. スーパーユーザー権限で、メディアをマウントします。

```
# mount -F nfs -o ro system-name:/media/media-name local-mount-point
```

<i>system-name</i>	マウントするメディアを持つシステムの名前
<i>media-name</i>	マウントするメディアの名前
<i>local-mount-point</i>	リモートのメディアのマウント先のローカルディレクトリ

4. スーパーユーザーをログアウトします。
5. メディアがマウントされていることを確認します。

```
$ ls /media
```

例 — 他のシステム上の CD にアクセスする

次の例は、sol_9_sparc という CD をリモートシステム starbug からローカルシステムの /cdrom ディレクトリにマウントする方法を示しています。

```
$ showmount -e starbug
export list for starbug:
/cdrom/sol_9_sparc (everyone)
$ su
Password: password
# mount -F nfs -o ro starbug:/cdrom/sol_9_sparc /cdrom
# exit
$ ls /cdrom
cdrom0      sol_9_sparc
```

例 — 他のシステム上のフロッピーディスクにアクセスする

次の例は、myfiles というフロッピーディスクをリモートシステム mars からローカルシステムの /floppy ディレクトリにマウントする方法を示しています。

```
$ cd /net/mars
$ ls /floppy
floppy0      myfiles
$ su
Password: password
# mount -F nfs rw mars:/floppy/myfiles /floppy
# exit
$ ls /floppy
myfiles
```

例 — 他のシステム上の PCMCIA メモリーカードにアクセスする

次の例は、myfiles という PCMCIA メモリーカードをリモートシステム mars からローカルシステムの /pcmem ディレクトリにマウントする方法を示しています。

```
$ cd /net/mars
$ ls /pcmem
pcmem0      myfiles
$ su
Password: password
# mount -F nfs rw mars:/pcmem/myfiles /pcmem
# exit
$ ls /pcmem
myfiles
```


第 19 章

リムーバブルメディアのフォーマット (手順)

この章では、Solaris 環境でコマンド行からリムーバブルメディアをフォーマットする方法について説明します。

リムーバブルメディアのフォーマットに関する手順については、255 ページの「リムーバブルメディアのフォーマット (作業マップ)」を参照してください。

リムーバブルメディアの概要については、第 17 章を参照してください。

リムーバブルメディアのフォーマット (作業マップ)

作業	説明	参照先
1. フォーマットされていないメディアのロード	メディアをドライブに挿入して、 <code>volcheck</code> コマンドを実行する	259 ページの「リムーバブルメディアを読み込む方法」
2. メディアのフォーマット	リムーバブルメディアをフォーマットする	261 ページの「リムーバブルメディアをフォーマットする方法 (<code>rmformat</code>)」
3. (省略可能) UFS ファイルシステムの追加	ファイル転送にフロッピーディスクを使用する場合は、UFS ファイルシステムを追加する	262 ページの「ファイルシステムの追加用にリムーバブルメディアをフォーマットする方法」

作業	説明	参照先
4. (省略可能) メディアの検査	メディア上のファイルシステムの整合性を検査する	263 ページの「リムーバブルメディア上のファイルシステムを検査する方法」
5. (省略可能) メディア上の不良ブロックの修復	必要に応じて、メディア上に不良ブロックがあれば修復する	264 ページの「リムーバブルメディア上の不良ブロックを修復する方法」
6. (省略可能) 読み取り/書き込み保護とパスワードによる保護の適用	必要に応じて、読み取り/書き込み保護やパスワードによる保護をメディアに適用する	265 ページの「リムーバブルメディアの書き込み保護を有効または無効にする方法」

リムーバブルメディアのフォーマットの概要

rmformat コマンドは、スーパーユーザー以外で使用できるユーティリティであり、書き込み可能なリムーバブルメディアのフォーマットや保護に使用できます。

rmformat コマンドには、次の3つのフォーマットオプションがあります。

- **quick** — このオプションは、トラックを検証せずに、あるいは、検証するトラックを制限して、リムーバブルメディアをフォーマットします。
- **long** — このオプションは、リムーバブルメディアを完全にフォーマットします。このオプションを使用してドライブ自身によるメディア全体の検証を行うデバイスもあります。
- **force** — このオプションは、ユーザーへの確認なしに、リムーバブルメディアを完全にフォーマットします。パスワードによる保護機能を備えたメディアでは、このオプションはフォーマットを行う前にパスワードをクリアします。この機能はパスワードを忘れてしまったときに便利です。パスワードによる保護機能を備えていないメディアでは、long オプションのフォーマットが行われます。

リムーバブルメディアのフォーマットのガイドライン

リムーバブルメディアをフォーマットするときは、次のことに注意してください。

- ファイルマネージャのウィンドウを閉じて、ファイルマネージャを終了してください。

ファイルマネージャは、フォーマットされていないメディアを挿入すると、フォーマットウィンドウを自動的に表示します。このウィンドウが表示されないようにするために、ファイルマネージャを終了してください。ファイルマネージャを開いたままにしておきたい場合は、フォーマットウィンドウが表示されてから、そのフォーマットウィンドウを終了してください。

- ボリュームマネージャ (vold) はファイルシステムを自動的にマウントするため、メディアに既存のファイルシステムが含まれている場合は、メディアをフォーマットする前にマウント解除する必要があります。

リムーバブルメディアのハードウェア面での考慮事項

この節では、リムーバブルメディアのハードウェアの面で考慮すべき事項について説明します。

フロッピーディスクのハードウェア面での考慮事項

フロッピーディスクをフォーマットするときは、次のことに注意してください。

- フロッピーディスク名について、表 18-1 を参照してください。
- 名前の付いていない (「ラベル」がない) フロッピーディスクには、noname というデフォルト名が割り当てられます。

Solaris システムは、Solaris システム用と DOS システム用にフロッピーディスクをフォーマットできます。ただし、ハードウェアプラットフォームによっていくつかの制限があります。次の表に、これらの制限がまとめてあります。

プラットフォームの種類	フロッピーディスクのフォーマット仕様
SPARC システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
	UDFS 用
x86 システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
	UDFS 用

UFS 用にフォーマットされたフロッピーディスクは、それらがフォーマットされたハードウェアプラットフォームに制限されます。つまり、SPARC システムでフォーマットされた UFS フロッピーディスクは、x86 システム上の UFS には使用できません。同様に、x86 システム上でフォーマットされたフロッピーディスクは SPARC シス

テム上の UFS には使用できません。これは、SPARC と x86 とでは UFS フォーマットが異なるためです。SPARC はリトルエンディアンによるビットコーディング、x86 はビッグエンディアンによるビットコーディングを採用しています。

SunOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、SunOS ファイルシステムをサポートするための構造からなります。DOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、MS-DOS または NEC-DOS のどちらかのファイルシステムをサポートする構造からなります。フロッピーディスクを準備するために必要な手順は、ファイルシステムによって異なります。したがって、フロッピーディスクをフォーマットする前には、どの作業が必要かを決めてください。詳細については、255 ページの「リムーバブルメディアのフォーマット (作業マップ)」を参照してください。

Solaris システム (SPARC または x86 のどちらか) では、次の密度でフロッピーディスクをフォーマットできます。

フロッピーディスクのサイズ	フロッピーディスクの密度	容量
3.5 インチ	高密度 (HD)	1.44M バイト
3.5 インチ	倍密度 (DD)	720K バイト

デフォルトで、フロッピーディスクドライブは、それに近い密度にフロッピーディスクをフォーマットします。つまり、デフォルトでは、1.44M バイトのドライブは、フロッピーディスクが実際に 1.44M バイトのフロッピーディスクかどうかに関係なく、特に指示しないかぎり、そのフロッピーディスクを 1.44M バイト用にフォーマットしようとしています。言い換えれば、フロッピーディスクもドライブも、その容量以下にフォーマットすることは可能です。

PCMCIA メモリーカードのハードウェア面での考慮事項

Solaris システムは、Solaris システム用と DOS システム用に PCMCIA メモリーカードをフォーマットできます。ただし、ハードウェアプラットフォームによっていくつかの制限があります。次の表に、これらの制限がまとめてあります。

プラットフォームの種類	PCMCIA メモリーカードのフォーマット仕様
SPARC システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
x86 システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用

UFS 用にフォーマットされた PCMCIA メモリーカードは、それらがフォーマットされたハードウェアプラットフォームに制限されます。つまり、SPARC システムでフォーマットされた UFS PCMCIA メモリーカードは、x86 システム上の UFS には使用できません。同様に、x86 システムでフォーマットされた PCMCIA メモリーカードは SPARC システムでは使用できません。これは、SPARC と x86 とでは UFS フォーマットが異なるためです。

UFS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、UFS ファイルシステムをサポートする構造からなります。DOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、MS-DOS または NEC-DOS のどちらかのファイルシステムをサポートする構造からなります。PCMCIA メモリーカードを準備するために必要な手順はファイルシステムによって異なります。したがって、PCMCIA メモリーカードをフォーマットする前には、どの作業が必要かを決めてください。

▼ リムーバブルメディアを読み込む方法

1. メディアを挿入します。
2. メディアがフォーマットされていることを確認します。
フォーマットされているかどうか分からない場合は、メディアを挿入し、手順 3 の説明に従って、コンソールの状態メッセージを確認してください。メディアをフォーマットする必要がある場合は、261 ページの「リムーバブルメディアをフォーマットする方法 (rmformat)」を参照してください。
3. ボリューム管理に通知します。

```
$ volcheck -v  
media was found
```

次の 2 つの状態メッセージのいずれかが表示されます。

media was found	<p>ボリューム管理がメディアを検出し、表 18-1 に記述されたディレクトリにそのメディアをマウントしようとする。</p> <p>メディアが正しくフォーマットされている場合は、コンソールにエラーメッセージが表示されない。</p> <p>メディアがフォーマットされていない場合でも「media was found」メッセージは表示されるが、次のようなエラーメッセージがコンソールに表示される。</p> <pre>fd0: unformatted diskette or no diskette in the drive fd0: read failed (40 1 0) fd0: bad format</pre> <p>メディアをフォーマットしてからでないと、ボリューム管理はそれをマウントできない。詳細は、第 19 章を参照。</p>
no media was found	<p>ボリューム管理は、メディアを検出しなかった。メディアが正しく挿入されていることを確認して、volcheck をもう一度実行する。うまくいかない場合は、メディアをチェックする。損傷の可能性がある。メディアを手動でマウントしてみることもできる。</p>

4. メディアの内容を表示して、メディアがマウントされていることを確認します。たとえば、フロッピーディスクの場合は次のように入力します。

```
$ ls /floppy
floppy0 myfiles
```

前に説明したように、floppy0 はフロッピーディスクの実際の名前へのシンボリックリンクです。この場合、myfiles が実際の名前です。正しくフォーマットされていて、名前がない場合は、unnamed_floppy と呼ばれます。

/floppy ディレクトリの下に何も表示されない場合は、フロッピーディスクがマウントされていないか、正しくフォーマットされていないかのどちらかです。これを調べるには、mount コマンドを実行して、次のような /floppy で始まる行を探してください (通常は、リストの最後にあります)。

```
/floppy/name on /vol/dev/diskette0/name
```

このような行が表示されない場合、フロッピーディスクはマウントされていません。コンソールウィンドウにエラーメッセージが表示されていないかどうか確認してください。

▼ リムーバブルメディアをフォーマットする方法 (rmformat)

rmformat コマンドを使用すると、メディアをフォーマットすることができます。このコマンドは、デフォルトで、メディア上にパーティション 0 とパーティション 2 (メディア全体) という 2 つのパーティションを作成します。

1. ボリュームマネージャが動作していることを確認します。動作していれば、デバイス名のニックネームを使用できます。

```
$ ps -ef | grep vold
root  212      1  0   Nov 03 ?           0:01 /usr/sbin/vold
```

vold の起動方法については、244 ページの「ボリューム管理 (vold) を再起動する方法」を参照してください。メディアのデバイス名の確認方法については、240 ページの「リムーバブルメディア名の使用」を参照してください。

2. リムーバブルメディアをフォーマットします。

```
$ rmformat -F [ quick | long | force ] device-name
```

rmformat のフォーマットオプションについては、前出の節を参照してください。

rmformat コマンドの出力によって不良ブロックが見つかった場合は、264 ページの「リムーバブルメディア上の不良ブロックを修復する方法」の指示に従って不良ブロックを修復してください。

3. (省略可能) リムーバブルメディアに、Solaris 環境で使用する 8 文字のラベルを付けます。

```
$ rmformat -b label device-name
```

DOS ラベルの作成方法については、mkfs_pcfs(1M) のマニュアルページを参照してください。

例 — リムーバブルメディアをフォーマットする

次の例は、フロッピーディスクのフォーマット方法を示しています。

```
$ rmformat -F quick /dev/rdiskette
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

```
.....
```

次の例は、Zip ドライブのフォーマット方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/zip0
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

```
.....
```

▼ ファイルシステムの追加用にリムーバブルメディアをフォーマットする方法

1. リムーバブルメディアをフォーマットします。

```
$ rmformat -F quick device-name
```

2. (省略可能) 代替の **Solaris** パーティションテーブルを作成します。

```
$ rmformat -s slice-file device-name
```

スライスファイルの例は次のようになります。

```
slices: 0 = 0, 30MB, "wm", "home" :
        1 = 30MB, 51MB :
        2 = 0, 94MB, "wm", "backup" :
        6 = 81MB, 13MB
```

3. スーパーユーザーになります。
4. 適切なファイルシステムの種類を決定し、次のいずれかの作業を選択します。

- a. **UFS** ファイルシステムを作成する

```
# newfs device-name
```

- b. **UDFS** ファイルシステムを作成する

```
# mkfs -F udfs device-name
```

例 — UFS ファイルシステム用にフロッピーディスクをフォーマットする

次の例は、フロッピーディスクをフォーマットし、そのフロッピーディスク上に UFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/floppy0
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
$ su
# /usr/sbin/newfs /vol/dev/aliases/floppy0
newfs: construct a new file system /dev/rdiskette: (y/n)? y
/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors
      1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
   32, 640, 1184, 1792, 2336,
#
```

例 — UFS ファイルシステム用に PCMCIA メモリーカードをフォーマットする

次の例は、PCMCIA メモリーカードをフォーマットし、そのカード上に UFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/pcm0
$ su
# /usr/sbin/newfs -v /vol/dev/aliases/pcm0
newfs: construct a new file system /vol/dev/aliases/pcm0: (y/n)? y
.
.
.
#
```

例 — PCFS ファイルシステム用にリムーバブルメディアをフォーマットする

次の例は、代替 fdisk パーティションを作成する方法を示しています。

```
$ rmformat -F quick /dev/rdisk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
$ su
# fdisk /dev/rdisk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdisk/c0t4d0s2:c: (y/n)? y
#
```

次の例は、fdisk パーティションを作成せずに、PCFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
$ su
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
#
```

▼ リムーバブルメディア上のファイルシステムを検査する方法

1. スーパーユーザーになります。
2. ネームサービスを確認し、次のいずれかの作業を選択します。
 - a. UFS ファイルシステムを検査する

```
# fsck -F ufs device-name
```

b. UDFS ファイルシステムを検査する

```
# fsck -F udfs device-name
```

c. PCFS ファイルシステムを検査する

```
# fsck -F pcfs device-name
```

例 — リムーバブルメディア上の PCFS ファイルシステムを検査する

次の例は、メディア上の PCFS ファイルシステムの整合性を検査する方法を示しています。

```
# fsck -F pcfs /dev/rdisk/c0t4d0s2
** /dev/rdisk/c0t4d0s2
** ファイルシステムのメタデータを走査する
** メタデータの不一致があれば訂正する
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

▼ リムーバブルメディア上の不良ブロックを修復する方法

ドライブが不良ブロック管理をサポートしている場合にのみ、検証中に見つかった不良セクタを `rmformat` コマンドで検証、解析、および修復できます。ほとんどのフロッピーディスクや PCMCIA メモリカードは不良ブロック管理をサポートしていません。

ドライブが不良ブロック管理をサポートしている場合、不良ブロックを修復するための最大の努力が行われます。それでも不良ブロックを修復できなかった場合、修復に失敗したことを示すメッセージが表示されます。

1. リムーバブルメディア上の不良ブロックを修復します。

```
$ rmformat -c block-numbers device-name
```

`block-numbers` には、前の `rmformat` セッションで獲得したブロック番号を 10 進数、8 進数、または 16 進数形式で指定します。

2. リムーバブルメディアを検証します。


```
$ rmformat -V read device-name
```

リムーバブルメディアに読み取り/書き込み保護とパスワードによる保護を適用する

Iomega メディア (Zip ドライブや Jaz ドライブなど) には、読み取り保護または書き込み保護を適用し、パスワードを設定することができます。

▼ リムーバブルメディアの書き込み保護を有効または無効にする方法

1. 書き込み保護を有効にするか無効にするかを決定し、次のいずれかの作業を選択します。

- a. 書き込み保護を有効にする

```
$ rmformat -w enable device-name
```

- b. 書き込み保護を無効にする

```
$ rmformat -w disable device-name
```

2. リムーバブルメディアの書き込み保護が有効または無効になっていることを確認します。

```
$ rmformat -p device-name
```

▼ Iomega メディアの読み取り/書き込み保護とパスワードを有効または無効にする方法

パスワードによる保護機能をサポートしている Iomega メディアには、最大 32 文字のパスワードを適用できます。Iomega メディアに対して読み取り保護または書き込み保護を設定するときは、必ずパスワードを使用する必要があります。このとき、パスワードを提供するように促すプロンプトが表示されます。

パスワード機能をサポートしていないリムーバブルメディア上でパスワードを適用しようとする、警告メッセージが表示されます。

1. 読み取り保護または書き込み保護とパスワード保護を有効または無効のどちらにするかを決定します。

- a. 読み取り保護または書き込み保護を有効にする

```
$ rmformat -W enable device-name
Please enter password (最大 32 文字): xxx
Please reenter password:
```

```
$ rmformat -R enable device-name
Please enter password (最大 32 文字): xxx
Please reenter password:
```

- b. 読み取り保護または書き込み保護を無効にし、パスワードを削除する

```
$ rmformat -W disable device-name
Please enter password (最大 32 文字): xxx

$ rmformat -R disable device-name
Please enter password (最大 32 文字): xxx
```

2. リムーバブルメディアの読み取り保護または書き込み保護が有効または無効になっていることを確認します。

```
$ rmformat -p device-name
```

例 — 読み取り/書き込み保護を有効または無効にする

次の例は、Zip ドライブに対して書き込み保護を有効にし、パスワードを設定する方法を示しています。

```
$ rmformat -W enable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
Please reenter password: xxx
```

次の例は、Zip ドライブに対して書き込み保護を無効にし、パスワードを削除する方法を示しています。

```
$ rmformat -W disable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
```

次の例は、Zip ドライブに対して読み取り保護を有効にし、パスワードを設定する方法を示しています。

```
rmformat -R enable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
Please reenter password: xxx
```

次の例は、Zip ドライブに対して読み取り保護を無効にし、パスワードを削除する方法を示しています。

```
$ rmformat -R disable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
```

第 20 章

CD への書き込み (手順)

この章では、`cdrw` コマンドを使用して、データ CD やオーディオ CD を作成およびコピーする手順について説明します。

- 270 ページの「RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法」
- 270 ページの「CD デバイスを確認する方法」
- 271 ページの「CD メディアをチェックする方法」
- 272 ページの「データ CD 用の ISO 9660 ファイルシステムを作成する方法」
- 273 ページの「マルチセッションのデータ CD を作成する方法」
- 275 ページの「オーディオ CD を作成する方法」
- 276 ページの「オーディオトラックを CD から抽出する方法」
- 277 ページの「CD をコピーする方法」
- 278 ページの「CD-RW メディアを消去する方法」

オーディオ CD やデータ CD の取り扱い

Solaris 9 リリースには `cdrw` コマンドが用意されています。このコマンドを使用すると、Rock Ridge 拡張または Joliet 拡張を備えた ISO 9660 フォーマットで、CD-R や CD-RW のメディアデバイス上に CD ファイルシステムを書き込むことができます。

`cdrw` コマンドを使用すると、次の作業を行えます。

- データ CD の作成
- オーディオ CD の作成
- オーディオ CD からのオーディオデータの抽出
- CD のコピー
- CD-RW メディアの消去

`cdrw` コマンドは、Solaris 8 Operating Environment 1/01 の Software Supplement の CD から利用でき、Solaris 9 リリースには最初から組み込まれています。

推奨される CD-R または CD-RW デバイスについては、http://www.sun.com/io_technologies/pci/removable.html をご覧ください。

CD メディアに関するよく使われる用語

CD メディアについて言及するときによく使われる用語は、次のとおりです。

用語	説明
CD-R	CD 読み取りメディア。1 度だけ書き込みができ、その後は読み取り専用となる。
CD-RW	書き換え可能な CD メディア。書き込みと消去が可能。CD-RW デバイスだけが CD-RW メディアを読み取れる。
ISO 9660	ISO (Industry Standards Organization の略)。コンピュータの標準記憶フォーマットを設定する組織。 ISO 9660 ファイルシステムは、標準の CD-ROM ファイルシステムであり、主だったコンピュータプラットフォームで同じ CD-ROM を読み取れる。この標準は 1988 年に発行され、ハイシエラ (ネバダ州のハイシエラホテルにちなんで名づけられた) という業界団体によって作成された。CD-ROM ドライブを備えたほぼすべてのコンピュータが ISO 9660 ファイルシステムからファイルを読み取れる。
Joliet 拡張	Windows™ ファイルシステム情報を追加する。
Rock Ridge 拡張	UNIX™ ファイルシステム情報を追加する。(Rock Ridge は映画「Blazing Saddles」に出てくる町にちなんで名づけられた。) 注 - これらの拡張は排他的ではない。したがって、mkisofs コマンドに -R と -j の両オプションを指定して両方のシステムとの互換性を確保できる。(詳細については mkisofs(1M) のマニュアルページを参照。)
MMC 準拠のレコード	Multi Media Command の略。これらのレコーダが共通のコマンドセットに準拠していること意味する。ある MMC 準拠レコーダに書き込めるプログラムは、他のすべての MMC 準拠レコーダにも書き込むことができる。
Red Book CDDA	Compact Disc Digital Audio の略。デジタルオーディオをコンパクトディスクに格納するための業界標準方式。「Red Book」形式とも呼ばれる。この公式の業界仕様では、1 つまたは複数のオーディオファイルが 44.1 kHz のサンプリングレートで 16 ビットのステレオサウンドにサンプリングされることが要求される。

CD メディアを扱うときによく使われる用語は、次のとおりです。

用語	説明
ブランキング	CD-RW メディアからデータを消去する処理
mkisofs	ISO ファイルシステムを CD に書き込むためのコマンド
セッション	リードイン/リードアウト情報を持つ完全なトラック
トラック	完全なデータまたはオーディオの単位

データ CD やオーディオ CD への書き込み

CD への書き込み処理は、途中で中断されることなく、データストリームが一定に保たれている必要があります。cdrw -s オプションを使用してメディアへの書き込みをシミュレートし、システムが CD への書き込みに十分適した速度でデータを送れるかどうかを確認してみてください。

次の場合には、書き込みエラーが発生することがあります。

- メディアがドライブの速度に対応できない場合。たとえば、メディアの中には 2x または 4x の速度しか保証されていないものもあります。
- システムが書き込み処理に支障をきたすほど多数の大きなプロセスを実行している場合
- リモートシステム上のイメージがある場合、ネットワークの混雑によってイメージ読み取りに遅延が生じる可能性がある場合
- CD 間でコピーするときに、送り側ドライブの速度が受け側ドライブよりも遅い可能性がある場合

上記の問題が発生した場合は、cdrw -p オプションを使用して、デバイスの書き込み速度を遅くしてください。

たとえば、4x の速度での書き込みをシミュレートします。

```
$ cdrw -iS -p 4 image.iso
```

また、cdrw -c オプションを使用すると、80 分の CD をコピーするのに、規定のメディア容量が使われます。このオプションを指定しないと、cdrw コマンドは、オーディオ CD のコピーに 74 分というデフォルト値を使用します。

詳細については、cdrw(1) のマニュアルページを参照してください。

RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する

デフォルトでは、すべてのユーザーが Solaris 9 リリースで起動するリムーバブルメディアを利用できます。ただし、役割によるアクセス制御 (RBAC) で役割を設定して、リムーバブルメディアへのユーザーアクセスを制限することができます。リムーバブルメディアへのアクセスを制限するには、限られたユーザーに役割を割り当てます。

役割の使用方法については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の役割」を参照してください。

▼ RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris** 管理コンソールを起動します。

```
$ /usr/sadm/bin/smc &
```

コンソールの起動方法については、64 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」を参照してください。

3. デバイス管理の権利が含まれている役割を設定します。
詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「管理役割 ツールを使用して役割を作成する方法」を参照してください。
4. **cdrw** コマンドを使用する必要があるユーザーを、新しく作成した役割に追加します。
5. **/etc/security/policy.conf** ファイルの次の行をコメントにします。

```
AUTHS_GRANTED=solaris.device.cdrw
```

この手順を実行しないと、デバイス管理役割のメンバーだけでなく、すべてのユーザーが引き続き **cdrw** コマンドを利用できます。

このファイルを変更した後は、デバイス管理役割のメンバーだけが **cdrw** コマンドを使用できるようになります。メンバー以外のユーザーがこのコマンドを使おうとすると、アクセスが拒否され、次のメッセージが表示されます。

```
Authorization failed, Cannot access disks.
```

CD デバイスを確認する方法

システム上の CD デバイスを確認する場合は、**cdrw -l** コマンドを使用します。

```

$ cdrw -l
Looking for CD devices...
  Node | Connected Device | Device type
-----+-----+-----
cdrom0 | YAMAHA CRW8424S | 1.0d | CD Reader/Writer

```

特定の CD デバイスを使用する場合は、`-d` オプションを指定します。たとえば、次のようになります。

```
$ cdrw -a filename.wav -d cdrom2
```

メディアが空であるか、または既存の目次があるかどうかを確認する場合は、`cdrw -M` コマンドを使用します。

```

$ cdrw -M

Device : YAMAHA CRW8424S
Firmware : Rev. 1.0d (06/10/99)
Media is blank
%
```

▼ CD メディアをチェックする方法

`cdrw` コマンドは、`vold` を使用している状態でも使用していない状態でも機能します。ただし、`vold` デーモンを終了/起動する場合は、スーパーユーザーまたは役割のアクセス権が必要となります。

1. CD を CD-RW デバイスに挿入します。

CD は、デバイスが読み取ることができるものであれば、どんな種類の CD でもかまいません。

2. デバイスのリストを表示して、その CD-RW ドライブが正しく接続されていることを確認します。

```

$ cdrw -l
Looking for CD devices...
  Node | Connected Device | Device type
-----+-----+-----
cdrom1 | YAMAHA CRW8424S | 1.0d | CD Reader/Writer

```

3. (省略可能) そのドライブがリスト内がない場合、システムがそのデバイスを認識するように再構成ブートを行う必要があることがあります。

```
# touch /reconfigure
# init 6
```

あるいは、次のコマンドを実行して、システムをリブートしないで CD-RW デバイスを追加できます。

```
# drvconfig
# disks
```

次に、`vold` を再起動します。

```
# /etc/init.d/vold stop
# /etc/init.d/vold start
```

データ CD を作成する

まず `mkisofs` コマンドを使用してファイルとファイル情報を CD で使用されるハイシエラ形式に変換し、データを準備します。

▼ データ CD 用の ISO 9660 ファイルシステムを作成する方法

1. 空の CD を **CD-RW** デバイスに挿入します。
2. 新しい CD で **ISO 9660** ファイルシステムを作成します。

```
$ mkisofs -r /pathname > cd-file-system
```

<code>-r</code>	Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする
<code>/pathname</code>	ISO 9660 ファイルシステムの作成に使われるパス名を指定する
<code>> cd-file-system</code>	CD に書き込むファイルシステムの名前を指定する

3. CD のファイルシステムを **CD** にコピーします。

```
$ cdrw -i cd-file-system
```

<code>-i cd-file-system</code>	データ CD を作成するためのイメージファイルを示す
--------------------------------	----------------------------

例 — データ CD 用の ISO 9660 ファイルシステムを作成する

次の例は、データ CD 用の ISO 9660 ファイルシステムを作成する方法を示しています。

```
$ mkisofs -r /home/dubs/ufs_dir> ufs_cd
Total extents actually written = 56
Total translation table size: 0
Total rockridge attributes bytes: 329
```



```
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 8000
56 extents written (0 Mb)
```

次に、CD のファイルシステムを CD にコピーします。たとえば、次のようになります。

```
$ cdrw -i ufs_cd
Initializing device...done.
Writing track 1...done.
Finalizing (数分かかる)...done.
```

▼ マルチセッションのデータ CD を作成する方法

この手順では、複数のセッションを CD に書き込む方法について説明します。また、`infoA` と `infoB` の各ディレクトリを CD にコピーする例も示します。

1. 最初の CD セッション用のファイルシステムを作成します。

```
$ mkisofs -o infoA -r -V my_infoA /data/infoA
Total translation table size: 0
Total rockridge attributes bytes: 24507
Total directory bytes: 34816
Path table size(bytes): 98
Max brk space used 2e000
8929 extents written (17 Mb)
```

<code>-o infoA</code>	ISO ファイルシステムの名前 (この場合は <code>infoA</code>) を指定する
<code>-r</code>	Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする
<code>-V my_infoA</code>	<code>vol1d</code> がマウントポイントとして使用するボリュームラベル (この場合は <code>my_infoA</code>) を指定する
<code>/data/infoA</code>	作成する ISO イメージディレクトリを指定する

2. 最初のセッションの ISO ファイルシステムを CD にコピーします。

```
$ cdrw -iO infoA
Initializing device...done.
Writing track 1...done.
done.
Finalizing (数分かかる)...done.
```

```
-i infoA
```

CD に書き込むイメージファイルの名前 (*infoA*) を指定する

```
-O
```

書き込むために CD を開いたままにしておく

3. CD が排出された後で、再度挿入します。

4. 次の書き込みセッションに含める CD メディアのパス名を確認します。

```
$ eject -n
.
.
.
cdrom0 -> /vol/dev/rdisk/c2t4d0/my_infoA
/vol/dev/... パス名を書き留めてください。
```

5. 次のセッションを書き込む CD 上の次に書き込み可能なアドレスを確認します。

```
% cdrw -M /cdrom
Device : YAMAHA CRW8424S
Firmware : Rev. 1.0d (06/10/99)

Track No. |Type |Start address
-----+-----+-----
1 |Audio |0
2 |Audio |33057
3 |Data |60887
4 |Data |68087
5 |Data |75287
Leadout |Data |84218

Last session start address: 75287
Next writable address: 91118
Next writable address: 出力に記述されているアドレスを書き留めて、次のセッションの書き込み時にこのアドレスを使用できるようにします。
```

6. 次の CD セッション用の ISO ファイルシステムを作成し、CD に書き込みます。

```
$ mkisofs -o infoB -r -C 0,91118 -M /vol/dev/rdisk/c2t4d0/my_infoA
/data/infoB
Total translation table size: 0
Total rockridge attributes bytes: 16602
Total directory bytes: 22528
Path table size(bytes): 86
Max brk space used 20000
97196 extents written (189 Mb)
```

```
-o infoB
```

ISO ファイルシステムの名前を指定する

```
-r
```

Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする

<code>-c 0,91118</code>	最初のセッションの開始アドレスと、次に書き込み可能なアドレスを指定する
<code>-M /vol/dev/rdisk/c2t4d0/my_infoA</code>	マージする既存の ISO イメージのパスを示す
<code>/data/infoB</code>	作成する ISO イメージディレクトリを指定する

オーディオ CD を作成する

`cdrw` コマンドを使用すると、個々のオーディオトラックまたは `.au` と `.wav` ファイルからオーディオ CD を作成できます。

次のオーディオ形式がサポートされています。

形式	説明
<code>sun</code>	Red Book CDDA 形式のデータが入る Sun <code>.au</code> ファイル
<code>wav</code>	Red Book CDDA 形式のデータが入る RIFF (<code>.wav</code>) ファイル
<code>cda</code>	raw CD オーディオデータ (「リトルエンディアン」バイト順序により、44.1 kHz のサンプリングレートでサンプリングされた 16 ビットの PCM ステレオ) が入る <code>.cda</code> ファイル
<code>aur</code>	<code>.aur</code> ファイル。「ビッグエンディアン」バイト順序による raw CD データが入る

オーディオ形式を指定しなかった場合、`cdrw` コマンドはファイル拡張子に基づいてオーディオファイルの形式を判断しようとします。ファイル拡張子の太文字と小文字は区別されません。

▼ オーディオ CD を作成する方法

この手順では、オーディオファイルを CD にコピーする方法について説明します。

1. 空の CD を **CD-RW** デバイスに挿入します。
2. オーディオファイルが入っているディレクトリに移動します。

```
$ cd /myaudiodir
```

3. オーディオファイルを **CD** にコピーします。

```
$ cdrw -a track1.wav track2.wav track3.wav
```

-a オプションによってオーディオ CD が作成されます。

例 — オーディオ CD を作成する

次の例は、オーディオ CD を作成する方法を示しています。

```
$ cdrw -a bark.wav chirp.au meow.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (数分かかる)...done.
```

次の例は、マルチセッションのオーディオ CD を作成する方法を示しています。最初のセッションの書き込みが終わると、CD が排出されます。次の書き込みセッションの前に CD を再度挿入します。

```
$ cdrw -a0 groucho.wav chico.au harpo.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (数分かかる)...done.
<CD を再挿入する>
$ cdrw -a zeppo.au
Initializing device...done.
Writing track 1...done.
done.
Finalizing (数分かかる)...done.
```

▼ オーディオトラックを CD から抽出する方法

オーディオトラックを CD から抽出して、新しい CD にコピーする場合は、次の手順に従います。

`cdrw -T` オプションを使ってオーディオファイル形式を指定しなかった場合、`cdrw` コマンドはファイル名拡張子を使ってオーディオファイル形式を判断します。たとえば、`cdrw` コマンドは、このファイルが `.wav` ファイルであることを検知します。

```
$ cdrw -x 1 testme.wav
```

1. オーディオ CD を CD-RW デバイスに挿入します。
2. オーディオトラックを抽出します。

```
$ cdrw -x -T audio-type 1 audio-file
```

<code>-x</code>	オーディオ CD からオーディオデータを抽出する
<code>T audio-type</code>	抽出されるオーディオファイルの形式を指定する。サポートされているオーディオ形式は、 <code>sun</code> 、 <code>wav</code> 、 <code>cda</code> 、 <code>aur</code>

- 抽出したトラックを新しい CD にコピーします。

```
$ cdrw -a audio-file
```

例 — オーディオトラックを CD から抽出してオーディオ CD を作成する方法

次の例は、オーディオ CD から最初のトラックを抽出し、そのファイルに `song1.wav` という名前を付ける方法を示しています。

```
$ cdrw -x -T wav 1 song1.wav
Extracting audio from track 1...done.
```

次の例は、オーディオ CD にトラックをコピーする方法を示しています。

```
$ cdrw -a song1.wav
Initializing device...done.
Writing track 1...done.
Finalizing (数分かかる)...done.
```

▼ CD をコピーする方法

この手順では、まずすべてのトラックをオーディオ CD から抽出してディレクトリに入れ、次にそれらのトラックをすべて空の CD にコピーする方法について説明します。

注 - デフォルトでは、`cdrw` コマンドは CD を `/tmp` ディレクトリにコピーします。コピーを行うには、最高 700M バイトの空き領域が必要です。CD をコピーするのに必要な空き領域が `/tmp` ディレクトリに不足している場合は、`-m` オプションを使って代替ディレクトリを指定します。

- オーディオ CD を CD-RW デバイスに挿入します。
- オーディオ CD からトラックを抽出します。

```
$ mkdir music_dir
$ cdrw -c -m music_dir
```

トラックごとに `Extracting audio ...` メッセージが表示されます。

すべてのトラックが抽出されると、CD が排出されます。

3. 空の CD を挿入して、**Return** キーを押します。
トラックの抽出が終わると、オーディオ CD が排出され、空の CD を挿入するよう指示するプロンプトが表示されます。

例 — CD をコピーする

次の例は、CD 間でコピーする方法を示しています。この作業を行うには、CD-RW デバイスが 2 台必要です。

```
$ cdrw -c -s cdrom0 -d cdrom1
```

▼ CD-RW メディアを消去する方法

CD を書き換える前に、既存の CD-RW データを消去する必要があります。

1. 次のいずれかの手順を選択して、メディア全体を消去するか、CD 上の最後のセッションだけを消去します。

- a. 最後のセッションだけを消去する

```
$ cdrw -d cdrom0 -b session
```

-b session オプションを使って最後のセッションだけを消去する場合は、-b all オプションを使ってメディア全体を消去する場合に比べて短い時間で済みます。cdrw コマンドを使用して、1 セッションだけでデータ CD またはオーディオ CD を作成した場合でも、-b session オプションを使用できます。

- b. メディア全体を消去する

```
$ cdrw -d cdrom0 -b all
```

第 21 章

ソフトウェアの管理

以下の各章では、Solaris 環境におけるソフトウェアの管理について説明します。

第 22 章	ソフトウェア製品の追加と削除についての概要を説明します。
第 23 章	ソフトウェアパッケージを追加および削除する手順について説明します。
第 24 章	Solaris パッチの管理情報の概要を説明します。
第 25 章	Solaris パッチの管理手順を説明します。

第 22 章

ソフトウェアの管理 (概要)

ソフトウェアの管理には、スタンドアロンシステム、サーバー、およびそのクライアントへのソフトウェアの追加やソフトウェアの削除が含まれます。この章では、ソフトウェアのインストールや管理に使用できる各種ツールについて説明します。

この章では、新しいシステムでの Solaris ソフトウェアのインストールについては説明しません。また、新バージョンの Solaris ソフトウェアのインストールやアップグレードについても説明しません。Solaris ソフトウェアのインストールやアップグレードについては、『Solaris 9 12/03 インストールガイド』を参照してください。

この章の内容は次のとおりです。

- 281 ページの「Solaris 9 Update リリースの新しいソフトウェア管理機能」
- 282 ページの「Solaris 9 リリースの新しいソフトウェア管理機能」
- 284 ページの「ソフトウェア管理作業についての参照先」
- 284 ページの「ソフトウェアパッケージの概要」
- 290 ページの「ソフトウェアパッケージを管理するためのツール」
- 291 ページの「ソフトウェアパッケージの追加または削除 (pkgadd)」
- 292 ページの「ソフトウェアパッケージの追加にあたっての重要な注意点 (pkgadd)」
- 292 ページの「パッケージの削除に関するガイドライン (pkgrm)」
- 293 ページの「パッケージ追加時のユーザーの対話操作を省略する (pkgadd)」

ソフトウェアを管理するための手順については、第 23 章を参照してください。

Solaris 9 Update リリースの新しいソフトウェア管理機能

この節では、この Solaris リリースの新しいソフトウェア管理機能について説明します。

pkgadd と patchadd による署名付きパッケージ およびパッチのサポート

Solaris 9 12/03 – この Solaris リリースでは、拡張された pkgadd コマンドと patchadd コマンドを使うことで、デジタル署名を含む Solaris パッケージと Solaris パッチを安全にダウンロードできるようになりました。

以前の Solaris リリースでは、Solaris パッチ管理ツールをダウンロードし、smpatch コマンドと PatchPro を併用することで、署名付きパッチの管理を行えました。smpatch コマンドを使って署名付きパッチを管理する手順については、355 ページの「smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)」を参照してください。

patchadd コマンドを使って署名付きパッチを追加する手順については、351 ページの「patchadd コマンドによる署名付きパッチの追加 (作業マップ)」を参照してください。

pkgadd コマンドを使って署名付きパッケージを追加する手順については、322 ページの「署名付きパッケージの追加と削除 (作業マップ)」を参照してください。

prodreg コマンドの拡張機能

Solaris 9 4/03 – prodreg コマンドのオプションを使って、コマンド行から Solaris Product Registry にアクセスし、管理できます。

prodreg コマンドを使ってソフトウェアパッケージを管理する方法については、303 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)」を参照してください。

Solaris 9 リリースの新しいソフトウェア 管理機能

この節では、Solaris 9 リリースの新しいソフトウェア管理機能について説明します。

署名付きパッチ

Solaris 2.6、7、8、9 リリースのすべてのパッチには、デジタル署名が付いています。有効なデジタル署名は、署名が適用された以降にパッチの変更が行われていないことを保証します。

署名付きパッチには、パッチがシステムに適用される前に確認できるデジタル署名が含まれているため、パッチのダウンロードやパッチの適用を安全に行うことができます。

署名付きパッチは、Java™ アーカイブ形式のファイル (*abc.jar*) 内に格納され、SunSolve OnlineSM から入手できます。

`smpatch` コマンドを使って署名付きパッチを追加する方法については、355 ページの「`smpatch` コマンドで署名付きパッチを管理するための準備 (作業マップ)」を参照してください。`smpatch` コマンドに関する問題の障害追跡については、<http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq> を参照してください。

Solaris Product Registry 3.0

Solaris Product Registry 3.0 は、ソフトウェアパッケージのインストールとアンインストールを行うための GUI ツールです。

この製品を使ってソフトウェアパッケージを管理する方法については、298 ページの「Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)」を参照してください。

パッチアナライザ

Solaris™ Web Start プログラムを使用して Solaris 9 Update リリースにアップグレードするときに、パッチアナライザを使用すると、システムの分析が行われて、Solaris Update リリースへのアップグレードによってどのパッチ (存在する場合) が削除またはダウングレードされるかが確認されます。Solaris 9 リリースにアップグレードするときは、パッチアナライザを使用する必要はありません。

Solaris 9 Update リリースへのアップグレード時にこのツールを使用する方法については、『Solaris 9 12/03 インストールガイド』の「Solaris Update リリースへのアップグレード」を参照してください。

Solaris 管理コンソールのパッチマネージャ

Solaris 管理コンソールには、パッチを管理するための新しいパッチツールが用意されています。このパッチツールを使用するだけで、Solaris 9 リリースを実行しているシステムにパッチを追加できます。

Solaris 管理コンソールの起動方法については、56 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

ソフトウェア管理作業についての参照先

ソフトウェアを管理する手順については、次の表を参照してください。

ソフトウェア管理作業	参照先
Solaris ソフトウェアのインストール	『Solaris 9 12/03 インストールガイド』
インストール後の Solaris ソフトウェアパッケージの追加または削除	第 23 章
インストール後の Solaris パッチの追加または削除	第 25 章
ソフトウェアパッケージ問題の障害追跡	『Solaris のシステム管理 (上級編)』の「ソフトウェアパッケージで発生する問題の解決 (手順)」

ソフトウェアパッケージの概要

ソフトウェア管理には、ソフトウェア製品のインストールと削除が含まれます。Sun および Sun 以外のベンダーは、「パッケージ」という形式でソフトウェア製品を提供しています。

「パッケージング」という用語は一般に、ソフトウェア製品が使用されるシステムに、その製品を配布してインストールする方法を指します。パッケージは、定義済みフォーマットによるファイルとディレクトリの集まりです。このフォーマットは、アプリケーションバイナリインタフェース (ABI) に準拠します。ABI は、System V インタフェース定義を補足するものです。Solaris オペレーティング環境には、このフォーマットを解釈し、パッケージをインストールまたは削除したり、パッケージインストールを検査したりする方法を提供する 1 組のユーティリティがあります。

パッチは、既存のソフトウェアの正常な実行の妨げとなっているファイルとディレクトリを置換または更新するためのファイルとディレクトリの集まりです。パッチの詳細については、第 24 章を参照してください。

署名付きパッケージおよびパッチ

パッケージにはデジタル署名を含めることができます。有効なデジタル署名付きのパッケージは、署名が適用された以降はパッケージの変更が行われていないことを保証します。署名付きパッケージを使用すると、パッケージをシステムに追加する前にデジタル署名を検証できるため、パッケージを安全にダウンロードまたは追加できます。

同じことが、署名付きパッチにも言えます。有効なデジタル署名付きのパッチは、署名が適用された以降はパッチの変更が行われていないことを保証します。署名付きパッチを使用すると、パッチをシステムに追加する前にデジタル署名を検証できるため、パッチを安全にダウンロードまたは追加できます。

署名付きパッチをシステムに追加する方法については、351 ページの「`patchadd` コマンドによる署名付きパッチの追加 (作業マップ)」を参照してください。

署名付きパッケージを作成する方法については、『*Application Packaging Developer's Guide* (英語版)』を参照してください。

署名付きパッケージは、デジタル署名が含まれる点以外は、署名なしパッケージとまったく同一です。署名付きパッケージのインストール、照会、または削除は、既存の Solaris パッケージングツールを使って行えます。また、署名付きパッケージと署名なしパッケージは、バイナリレベルでも互換性があります。

デジタル署名されたパッケージまたはパッチをシステムに追加するには、信頼される証明書を使ってパッケージキーストアを設定しておく必要があります。これらの証明書は、パッケージやパッチ上のデジタル署名の有効性を確認する際に使用されます。

次の表は、署名付きパッケージおよびパッチに関する一般的な用語の一覧です。

用語	定義
キーストア	証明書と鍵を格納するためのリポジトリ。それらの情報が必要になると、このリポジトリが検索される。 <ul style="list-style-type: none">■ Java キーストア – Solaris リリースでデフォルトでインストールされる証明書のリポジトリ。 Java キーストアは通常、<code>/usr/j2se/jre/lib/security</code> ディレクトリに格納される。■ パッケージキーストア – 署名付きパッケージおよびパッチをシステムに追加する際にインポートする証明書用のリポジトリ。 パッケージキーストアは、デフォルトで <code>/var/sadm/security</code> ディレクトリに格納される。

用語	定義
信頼される証明書	別のエンティティに属する公開鍵を備えた証明書。「信頼される証明書」という呼び名は、証明書に含まれている公開鍵が、その証明書のサブジェクトまたは所有者によって示された本人のものであることを、キーストアの所有者が信頼することに由来している。この信頼を表明するために、証明書の発行者はその証明書に署名する。 信頼される証明書は、署名を検証する際やセキュリティ保護されたサーバー (SSL サーバー) への接続を確立する際に使用される。
ユーザー鍵	暗号鍵に関する機密情報を保持する。この情報は、不正なアクセスを防ぐために、セキュリティが施された形式で格納される。ユーザー鍵は、ユーザーの非公開鍵と対応する公開鍵証明書から構成される。

署名付きパッケージまたはパッチをシステムに追加する手順は、基本的に次の3つから構成される。

1. pkgadm コマンドを使ってシステムのパッケージキーストアに証明書を追加する
2. (省略可能) pkgadm コマンドを使って証明書を一覧表示する
3. pkgadd コマンドを使ってパッケージを追加するか、patchadd コマンドを使ってパッチを追加する

署名付きパッケージをシステムに追加する手順については、322 ページの「署名付きパッケージの追加と削除 (作業マップ)」を参照してください。署名付きパッチをシステムに追加する手順については、351 ページの「patchadd コマンドによる署名付きパッチの追加 (作業マップ)」を参照してください。

Sun の証明書による署名付きパッケージおよびパッチの検証

「ストリーム形式」の SVR4 署名付きのパッケージまたはパッチには、PEM でコード化された PKCS7 署名が埋め込まれています。この署名には少なくとも、パッケージまたはパッチの暗号化されたダイジェストと署名者の X.509 公開鍵証明書とが含まれています。また、パッケージまたはパッチには「証明書チェーン」を含めることもできます。証明書チェーンを使うと、署名者の証明書からローカルに格納されている信頼される証明書へ、信頼関係のチェーンを形成することができます。

PEM でコード化された PKCS7 署名を使って、次のことを検証します。

- そのパッケージに署名したエンティティがそのパッケージの作成者である
- そのエンティティが実際にそのパッケージに署名した
- そのパッケージがエンティティによる署名後に変更されていない
- そのパッケージに署名したエンティティが信頼されたエンティティである

次の表は、署名付きパッケージおよびパッチに関する暗号化関連の用語の一覧です。

用語	定義
ASN.1	ASN.1 (Abstract Syntax Notation 1) は、一連の抽象オブジェクトを表現するための方式の 1 つである。たとえば、ASN.1 では、公開鍵証明書、公開鍵証明書を構成するすべてのオブジェクト、オブジェクトの収集順序などを定義している。ただし、ASN.1 では、オブジェクトを保存用または転送用にシリアルライズする方法は定義していない。
base64	base64 は、任意のバイナリデータを ASCII テキストとしてコード化する方式の 1 つである。
DER	DER (Distinguished Encoding Rules) は、ASN.1 オブジェクトのバイナリ表現の 1 つである。DER では、コンピューティング環境で ASN.1 オブジェクトを保存用または転送用にシリアルライズする方法を定義している。
PEM	PEM (Privacy Enhanced Message) は、base64 エンコーディングといくつかの任意のヘッダーを使って、ファイルを (DER またはその他のバイナリ形式で) コード化する方式である。もともとは、MIME タイプの電子メールメッセージをコード化するために使用された。また、PEM は、証明書と非公開鍵をファイルシステム上または電子メールメッセージ内のファイルにコード化する際にも広く使用される。
PKCS7	PKCS7 (Public Key Cryptography Standard #7) は、デジタル署名やデジタルエンベロープなどの暗号化データに対する汎用的な構文を定めた規格である。
X.509	ITU-T (International Telecommunication Union-Telcom) 勧告 X.509 は、広く採用されている X.509 公開鍵証明書の構文を定めた仕様である。 この勧告では、認証サービスを提供するためのフレームワークが定義されている。X.509 で定義された 2 つの認証レベルを次に示す。 <ul style="list-style-type: none"> ■ 単純な認証 – パスワードを使ってユーザーの身元を検証する方式 ■ 強い認証 – 暗号化技術を使って作成された資格情報に基づく方式。単純な認証でも不正なアクセスに対する多少の備えにはなるが、安全なサービスの提供基盤として使えるのは、強度の高い認証のみである。

Sun Microsystems から発行された認証済みデジタル署名を使って、ダウンロードされたパッケージやパッチが改ざんされていないことを検証できます。こうした証明書は、システムのキーストアにインポートされます。

Sun のすべての証明書は、先頃 GTE CyberTrust を買収した Baltimore Technologies によって発行されています。

Sun の証明書をインポートするときに特別なパスワードを使用することで、キーストアへのアクセスを保護できます。

pkgadm listcert コマンドを使えば、ローカルのパッケージキーストア内に格納されている証明書の情報を表示できます。たとえば、次のようになります。

```
# pkgadm listcert -P pass:store-pass
Keystore Alias: GTE CyberTrust Root
Common Name: GTE CyberTrust Root
Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC:65:A6...
```

次の表は、pkgadm listcert コマンドの出力を一覧にまとめたものです。

フィールド	説明
Keystore Alias (キーストア別名)	印刷、署名、削除のいずれかの目的で証明書を取得する場合、この名前を証明書の参照名として使用する必要がある。
Common Name (共通名)	証明書の共通の名前。信頼される証明書の場合、この名前はキーストア別名と同じになる。
Certificate Type (証明書の種類)	次の2つの種類がある。 <ul style="list-style-type: none"> ■ Trusted Certificate (信頼される証明書) - ほかの証明書を検証する際の信頼基盤として使用可能な証明書。信頼される証明書には非公開鍵は関連付けられない。 ■ Signing Certificate (署名証明書) - パッケージまたはパッチへの署名用として使用可能な証明書。署名証明書には非公開鍵が関連付けられる。
Issuer Common Name (発行者の共通名)	この証明書の署名、発行をしたエンティティの名前。信頼される証明書発行局 (CA、Certificate Authority) が発行した証明書の場合、発行者の共通名は、共通名と同じになる。
Validity Dates (有効な日付)	証明書の有効な日付。
MD5 Fingerprint (MD5 フィンガープリント)	証明書の MD5 ダイジェスト。このダイジェストを使えば、その証明書が証明元からの転送中に改変されなかったかどうかを検証できる。
SHA1 Fingerprint (SHA1 フィンガープリント)	MD5 フィンガープリントに似ているが、異なるアルゴリズムを使って計算される。

証明書を認証するには、各証明書の MD5 ハッシュおよび SHA1 ハッシュ(「フィンガープリント」とも呼ばれる)を、発行者によって公開されている既知の正しいフィンガープリントと比較します。

SunSolve Online の信頼される証明書

SunSolve Online は、以下の証明書を使って署名付きパッチ上のデジタル署名を検証します。なお、その際、PatchPro を含む以前の Solaris パッチ管理ツール (smpatch コマンド) を使います。

- ルート CA 証明書と呼ばれるトップレベルの証明書
- Class B CA 証明書と呼ばれるサブ CA (Sun Microsystems Inc.) による証明書
- Sun Enterprise™ Services によって発行される、「パッチ管理証明書」と呼ばれる証明書

「証明書発行局」は、パッチ上のデジタル署名の復号時に使用される公開鍵と、こうした公開鍵の所有者の関係を認証します。

Sun ルート CA 証明書、Sun Class B CA 証明書、およびパッチ署名証明書は、PatchPro を含む Solaris パッチ管理ツールに含まれています。この 3 つの証明書により、Sun のルート CA 証明書から Class B CA 証明書、Class B CA 証明書からパッチ管理証明書にさかのぼってパッチの信頼性を検証する認証チェーンが構成されます。最終的に、Sun のルート CA は、GTE CyberTrust CA によって認証されます。

Sun の信頼される証明書のインポート

署名付きパッケージおよびパッチを追加するための Sun の信頼される証明書を取得するには、次の方法を使います。

- **Java** キーストア – Sun のルート CA 証明書をインポートします。この証明書は、この Solaris リリースをインストールする際に Java キーストア内にデフォルトで格納されます。
- **Sun の PKI (Public Key Infrastructure)** サイト – システム上に Java キーストアが存在していない場合は、以下のサイトから証明書をインポートできます。
<https://ra.sun.com:11005/>
- **PatchPro** のキーストア – smpatch コマンドを使って署名付きパッチを追加するための PatchPro がすでにインストールされている場合、Java キーストアから Sun のルート CA 証明書をインポートできます。

パッケージキーストアのセットアップ

以前の Solaris リリースでは、パッチ管理ツールをダウンロードし、keytool コマンドを使って証明書をインポートすることで、PatchPro が使用する Java キーストアを作成することができました。

証明書の格納された Java キーストアがシステム上にすでに存在している場合、Sun Microsystems のルート CA 証明書を Java キーストアから `keytool` コマンドを使ってエクスポートできます。続いて、その証明書を、`pkgadm` コマンドを使ってパッケージキーストア内にインポートします。

ルート CA 証明書のパッケージキーストアへのインポートが完了すると、`pkgadd` コマンドや `patchadd` コマンドを使って署名付きパッケージおよびパッチをシステムに追加できるようになります。

注 – なお、Sun Microsystems のルート証明書が必要となるのは、Sun が署名したパッチやパッケージを追加する場合のみです。

証明書をパッケージキーストアにインポートする手順については、322 ページの「信頼される証明書をパッケージキーストアにインポートする方法 (`pkgadm addcert`)」を参照してください。

`pkgadd` コマンドを使って署名付きパッケージを追加するための完全な手順については、322 ページの「署名付きパッケージの追加と削除 (作業マップ)」を参照してください。

ソフトウェアパッケージを管理するためのツール

Solaris リリースをシステムにインストールした後でソフトウェアパッケージをシステムに追加したり、システムから削除するためのツールは、次のとおりです。

表 22-1 ソフトウェアパッケージツール

ソフトウェアパッケージ情報の追加、削除、表示に使用するツール	追加機能
Solaris Web Start プログラム	インストーラを起動して、Solaris 9 メディアパックに含まれている製品を追加する。個々のソフトウェアパッケージを追加することはできない

表 22-1 ソフトウェアパッケージツール (続き)

ソフトウェアパッケージ情報の追加、削除、表示に使用するツール	追加機能
Solaris Product Registry (GUI)	インストーラを起動して、ソフトウェア製品情報を追加、削除、または表示する。Solaris Web Start プログラムや Solaris pkgadd コマンドを使って初めからインストールされているソフトウェア製品に関する情報を削除または表示する場合は、Solaris Product Registry を使用する
Solaris Product Registry prodreg ビューア (コマンド行インタフェース)	prodreg コマンドでは、Solaris Web Start プログラムや Solaris pkgadd コマンドを使ってインストールされているソフトウェア製品に関する情報を削除または表示できる
パッケージコマンド (pkgadd、pkgrm、pkginfo)	これらのコマンドをスクリプトに組み込む、オプションのファイルを設定してユーザーの対話操作を省略したり特別なチェック作業を追加したりする、ソフトウェアパッケージをスプールディレクトリにコピーする

ソフトウェアパッケージの追加または削除 (pkgadd)

表 22-1 に記載されているソフトウェア管理ツールはすべて、インストール済みソフトウェアに関する情報の追加、削除、または照会に使用できます。Admintool、Solaris Product Registry の prodreg ビューア、および Web Start プログラムはいずれも、Solaris Product Registry に格納されているインストールデータにアクセスします。pkgadd コマンドや pkgrm コマンドといったパッケージツールでも、インストールデータにアクセスし、管理できます。

パッケージを追加する際、pkgadd コマンドは、ファイルを解凍して、インストール用メディアからローカルシステムのディスクにコピーします。パッケージを削除する際、pkgrm コマンドは、そのパッケージに関連するファイルが他のパッケージと共有されている場合を除いて、それらをすべて削除します。

パッケージファイルはパッケージ形式で配布され、配布されたままの状態では使用できません。pkgadd コマンドは、ソフトウェアパッケージの制御ファイルを解釈してから、製品ファイルを解凍して、システムのローカルディスクにインストールします。

pkgadd と pkgrm の各コマンドは、標準の場所にそのログ出力を記録しませんが、インストールまたは削除される製品を常時監視しています。pkgadd と pkgrm の各コマンドは、インストールまたは削除されたパッケージに関する情報をソフトウェア製品用データベースに格納します。

このデータベースを更新することによって、pkgadd と pkgrm の各コマンドは、システムにインストールされているすべてのソフトウェア製品の記録を保持します。

ソフトウェアパッケージの追加にあたっての重要な注意点 (pkgadd)

使用中のシステムでパッケージのインストールまたは削除を行う場合は、次の点に注意する必要があります。

- パッケージの命名規則 – Sun パッケージは、SUNWaccr、SUNWadmap、SUNWcsu などのように、必ず接頭辞 SUNW で始まります。Sun 以外のパッケージは、通常、その会社を表す接頭辞で始まります。
- インストール済みのソフトウェア – Web Start プログラム、Solaris Product Registry の prodreg ビューア (GUI または CLI)、Admintool、または pkginfo コマンドを使って、システムにインストールされているソフトウェアを特定できません。
- サーバーとクライアントによるソフトウェアの共有の状態 – クライアントのソフトウェアは、一部がサーバーに、一部がクライアントに置かれる場合があります。このような場合、クライアントにソフトウェアを追加するには、サーバーとクライアントの両方にパッケージを追加する必要があります。

パッケージの削除に関するガイドライン (pkgrm)

パッケージを削除するときは、rm コマンドを代わりに使いたい場合でも、これらのツールのいずれかを使用する必要があります。たとえば、rm コマンドを使用すると、バイナリ実行可能ファイルを削除することができますが、これは pkgrm コマンドを使用してバイナリ実行可能ファイルを含むソフトウェアパッケージを削除する場合は異なります。rm コマンドを使用してパッケージのファイルを削除すると、ソフトウェア製品用データベースが破壊されます。1つのファイルだけを削除したい場合は、removef コマンドを使用します。このコマンドは、該当ファイルがパッケージから削除されるように、ソフトウェア製品データベースを正しく更新します。詳細については、removef (1M) のマニュアルページを参照してください。

複数のバージョンのパッケージ (たとえば、複数バージョンの文書処理アプリケーション) をインストールしておきたい場合は、`pkgadd` コマンドを使用して、すでにインストール済みのパッケージとは異なるディレクトリに新しいバージョンをインストールしてください。パッケージがインストールされているディレクトリは、ベースディレクトリと呼ばれます。ベースディレクトリは、管理ファイルと呼ばれる特殊ファイルに `basedir` キーワードを設定することによって操作できます。管理ファイルの使用法とベースディレクトリの設定方法については、293 ページの「パッケージ追加時のユーザーの対話操作を省略する (`pkgadd`)」と、`admin(4)` のマニュアルページを参照してください。

注 - Solaris ソフトウェアをインストールするときにアップグレードオプションを使用すると、Solaris インストール用ソフトウェアは、ソフトウェア製品用データベースを調べて、すでにシステムにインストールされている製品があるかどうかを確認します。

パッケージ追加時のユーザーの対話操作を省略する (`pkgadd`)

管理ファイルの使用

`pkgadd -a` コマンドを使用すると、特殊な管理ファイルを調べて、インストールの進め方についての情報を入手できます。通常、`pkgadd` コマンドはいくつかのチェックを行い、指定されたパッケージを実際に追加する前に、プロンプトを表示してユーザーに確認します。ただし、管理ファイルを作成すれば、このようなチェックを省略して、ユーザーの確認なしでパッケージをインストールするように `pkgadd` コマンドに指示できます。

デフォルトでは、`pkgadd` コマンドは現在の作業用ディレクトリに管理ファイルがないか調べます。現在の作業用ディレクトリの中に管理ファイルを見つけることができなかつた場合、`pkgadd` コマンドは `/var/sadm/install/admin` ディレクトリに指定の管理ファイルがないか調べます。また、`pkgadd` コマンドには管理ファイルへの絶対パスも使用できます。



注意 - 管理ファイルは注意して使用してください。管理ファイルを使用して、`pkgadd` が通常実行するチェックとプロンプトを省略する場合は、事前にパッケージのファイルがインストールされている場所や、パッケージのインストール用スクリプトがどのように実行されるのかを知っておく必要があります。

次の例は、pkgadd コマンドがパッケージのインストール前にユーザーに確認のプロンプトを表示しないようにするための管理ファイルを示しています。

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

パッケージを追加するときにユーザーの対話操作を省略する以外にも、いろいろな目的で管理ファイルを使用できます。たとえば、管理ファイルを使用すれば、エラーが発生した場合に(ユーザーの対話操作なしに)パッケージのインストールを終了したり、pkgrm コマンドでパッケージを削除する場合に対話操作を省略できます。

また、特別なインストールディレクトリをパッケージに割り当てることもできます。この方法は、1つのシステム上で複数のバージョンのパッケージを管理する場合に役に立ちます。これを行うには、パッケージがインストールされる場所を示す代替ベースディレクトリを管理ファイル内に設定します(basedir キーワードを使用)。詳細については、admin(4) のマニュアルページを参照してください。

応答ファイルの使用 (pkgadd)

応答ファイルには、「対話型パッケージ」で尋ねられる特定の質問に対するユーザーの応答が格納されます。対話型パッケージには、パッケージをインストールする前にいくつかの質問(たとえば、パッケージのオプションをインストールするかどうかなど)をユーザーに尋ねる request スクリプトが格納されています。

インストールするパッケージが対話型パッケージであることがインストール前にわかっていて、さらにそのパッケージをこの先インストールするときにユーザーの対話操作を省略できるように応答を格納しておきたい場合は、pkgask コマンドを使用してユーザーの応答を保存できます。このコマンドの詳細については、pkgask(1M) のマニュアルページを参照してください。

request スクリプトが尋ねる質問への応答を格納した後は、pkgadd -r コマンドを使用して、ユーザーの対話操作なしにパッケージをインストールすることができます。

第 23 章

ソフトウェアの管理 (手順)

この章では、ソフトウェアパッケージを追加、検証、削除する方法について説明します。

ソフトウェアの管理作業に関連する手順については、次の節を参照してください。

- 297 ページの「Solaris Web Start プログラムを使ってソフトウェアをインストールする方法」
- 298 ページの「Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)」
- 303 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)」
- 322 ページの「署名付きパッケージの追加と削除 (作業マップ)」
- 327 ページの「パッケージコマンドによるソフトウェアパッケージの管理 (作業マップ)」
- 337 ページの「Admintool によるソフトウェアパッケージの追加と削除 (作業マップ)」

ソフトウェアパッケージを管理するためのコマンド

表 23-1 に、Solaris のインストール後に、ソフトウェアパッケージの追加、削除、検査に使用するコマンドを示します。

表 23-1 ソフトウェアパッケージを管理するためのツールまたはコマンド

ツールまたはコマンド	マニュアルページ	説明
admintool	admintool (1M)	グラフィカルツールを使用してソフトウェアパッケージをインストールまたは削除する
installer	installer (1M)	インストーラを使用してソフトウェアパッケージをインストールまたは削除する
pkgadd	pkgadd (1M)	署名付きまたは署名なしのソフトウェアパッケージをインストールする
pkgadm	pkgadm (1M)	署名付きパッケージおよびパッチの管理に使用する鍵と証明書を保守する
pkgchk	pkgchk (1M)	ソフトウェアパッケージのインストールを検査する
pkginfo	pkginfo (1)	ソフトウェアパッケージ情報を表示する
pkgparam	pkgparam (1)	ソフトウェアパッケージのパラメータ値を表示する
pkgrm	pkgrm (1M)	ソフトウェアパッケージを削除する
prodreg	prodreg (1M)	Solaris Product Registry 内のソフトウェアをブラウズ、登録解除、およびアンインストールする
pkgtrans	pkgtrans (1)	インストール可能なパッケージをある形式から別の形式に変換する。-g オプションを指定して pkgtrans コマンドを実行すると、署名が生成され、該当するデータストリーム内に格納される

Solaris Web Start プログラムによるソフトウェアの追加

この節では、Solaris オペレーティング環境がすでにインストールされているシステムに、Solaris Web Start プログラムを使ってソフトウェアを追加する方法を説明します。Solaris Web Start プログラムでは、Solaris オペレーティング環境を最初にインストールしたときに、インストール対象から外したソフトウェアグループのコンポーネントだけをインストールします。インストールまたはアップグレード後に別のソフトウェアグループのアップグレードを行うことはできません。4つのソフトウェアグループについては、『Solaris 9 12/03 インストールガイド』を参照してください。

▼ Solaris Web Start プログラムを使ってソフトウェアをインストールする方法

注 – この手順では、システムがボリューム管理 (vold) を実行しているものとします。システムがボリューム管理を実行していない場合、ボリューム管理を使用しないでリムーバブルメディアにアクセスする方法については、第 18 章を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. インストールを **CD**、**DVD**、またはネットワークのいずれから行うかを決めます。次のいずれかの手順に従います。
 - CD からインストールする場合は、CD-ROM ドライブに CD を挿入する。
Solaris 9 LANGUAGES CD を挿入すると、Solaris Web Start プログラムが自動的に起動されます。手順 6 へ進んでください。
 - DVD からインストールする場合は、DVD-ROM ドライブに DVD を挿入する。
 - ネットワークからインストールする場合は、インストールするソフトウェアのネットイメージを見つける。
3. **Solaris Web Start** インストーラが見つかるまで、ディレクトリを変更します。
Solaris Web Start インストーラは、CD や DVD 上のさまざまなディレクトリにあります。
 - Solaris 9 Software 1 of 2 CD
 - Solaris 9 Software 2 of 2 CD
 - Solaris 9 Documentation CD
 - Solaris 9 Languages CD。この CD を挿入すると、Solaris Web Start プログラムが自動的に起動されるCD や DVD の構成については、『Solaris 9 12/03 インストールガイド』の「Solaris メディアの構成に関するリファレンス情報」を参照してください。
4. 指示に従って、ソフトウェアをインストールします。
 - ファイルマネージャから、「インストール (Installer)」をダブルクリックする。
 - コマンド行から、次のように入力する。

```
% ./installer [options]
```

```
-nodisplay
```

GUI を使用しないでインストーラを実行する。

-noconsole

対話式のテキストコンソールデバイスを使用しないで実行する。ソフトウェアをインストールするための UNIX スクリプトに `installer` コマンドを組み込んだときは、このオプションを `-nodisplay` オプションと合わせて使用する。

指示に従って、ソフトウェアをインストールします。

5. 「インストール (**Installer**)」をダブルクリックします。
「インストール (**Installer**)」ウィンドウが表示され、続いて「Solaris Web Start」ダイアログボックスが表示されます。
6. 画面の指示に従ってソフトウェアをインストールします。
7. ソフトウェアの追加が終わったら、「終了 (**Exit**)」をクリックします。
Solaris Web Start プログラムが終了します。

Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)

次の作業マップに、Solaris Product Registry を使って行うことができるソフトウェアの管理作業を示します。

作業	説明	参照先
インストールまたはアンインストールしたソフトウェアを Solaris Product Registry を使って表示する	インストールまたはアンインストールしたソフトウェアを Solaris Product Registry を使って表示できる	300 ページの「インストールまたはアンインストールしたソフトウェアの情報を Solaris Product Registry の GUI を使って表示する方法」
Solaris Product Registry を使ってソフトウェアをインストールする	Solaris Product Registry を使ってソフトウェアを検索し、Solaris Web Start プログラムを起動できる。このプログラムの指示に従って、ソフトウェアのインストールを行う	301 ページの「Solaris Product Registry の GUI を使ってソフトウェアをインストールする方法」

作業	説明	参照先
Solaris Product Registry を使ってソフトウェアをアンインストールする	Solaris Product Registry を使ってソフトウェアをアンインストールできる	302 ページの「Solaris Product Registry の GUI を使ってソフトウェアをアンインストールする方法」

Solaris Product Registry は、インストールされているソフトウェアをユーザーが管理するためのツールです。ソフトウェアのインストールが終わると、Solaris Product Registry は、Solaris Web Start プログラム 3.0 または Solaris pkgadd コマンドを使用して、インストールされたすべてのソフトウェアの一覧を表示します。

Solaris Product Registry は、GUI またはコマンド行インタフェース (CLI) で使用できます。Solaris Product Registry CLI の使用方法については、303 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)」を参照してください。

Solaris Product Registry の GUI インタフェースでは、次の作業を実行できます。

- インストールおよび登録されているソフトウェアとソフトウェア属性の一覧表示
- System Software Localizations ディレクトリにインストールされている地域対応版のすべての Solaris システム製品の表示
- インストーラの選択と起動
- 追加のソフトウェア製品のインストール
- ソフトウェアや個々のソフトウェアパッケージのアンインストール

Solaris Product Registry の GUI のメインウィンドウは、次の 3 つの情報領域から構成されています。

- インストール、登録、削除されたソフトウェア
- 選択されているソフトウェアの標準属性
- カスタマイズされた属性と登録されているソフトウェアの内部属性

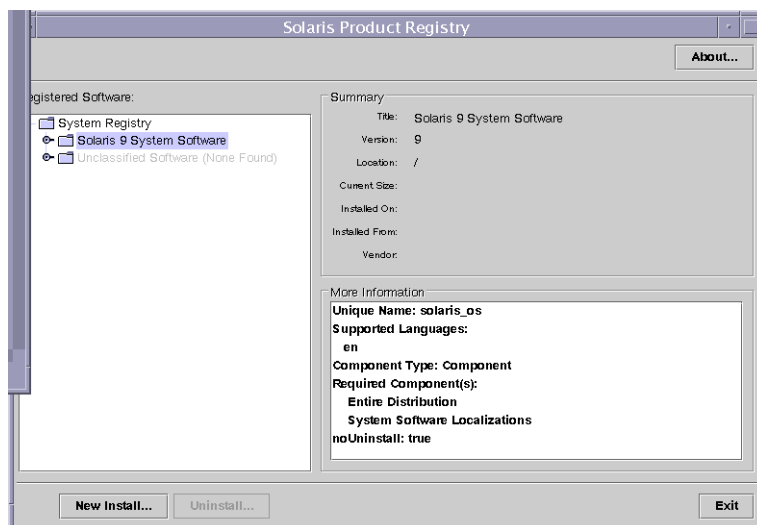


図 23-1 「Solaris Product Registry」 ウィンドウ

▼ インストールまたはアンインストールしたソフトウェアの情報を Solaris Product Registry の GUI を使って表示する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris Product Registry** ツールを起動します。

```
# prodreg &
```

「Solaris Product Registry」メインウィンドウが表示されます。

3. 「登録されているソフトウェア (**Registered Software**)」ボックスの「システムレジストリ (**System registry**)」ディレクトリの左側にあるアイコンをクリックします。

右側を指していたアイコンが下側が変わります。Solaris Product Registry の項目は、左側にテキストファイルアイコンがあるものを除き、どの項目も展開または縮小できます。

「登録されているソフトウェア (Registered Software)」ボックスにインストールされているソフトウェアには、必ず次のものが含まれます。

- Solaris のインストール時に選択した構成ソフトウェアグループ。ソフトウェアグループは、「コアシステムサポート (Core System Support)」、「エンドユーザーシステムサポート (End User System Support)」、「開発者システムサポート (Developer System Support)」、「全体ディストリビューション (Entire

Distribution)」、「全体ディストリビューションと OEM サポート (Entire Distribution plus OEM Support)」のいずれかです。

- 追加のシステムソフトウェア (選択したソフトウェアグループに含まれていない Solaris 製品)。
 - 分類していないソフトウェア (pkgadd コマンドを使ってインストールした Solaris 製品以外のパッケージまたはソフトウェアグループに含まれていないパッケージ)。
4. 表示するソフトウェアアプリケーションが見つかるまでディレクトリを選択します。
ディレクトリを開くと、リストが展開されます。
 5. 属性を表示する場合は、ディレクトリまたはファイルを選択します。
属性情報が「一覧 (Summary)」ボックスと「詳細情報 (more information)」ボックスに表示されます。
 - Solaris Web Start プログラムでインストールしたソフトウェア製品の場合、「タイトル (Title)」、「バージョン (Version)」、「インストール場所 (Location)」、「インストール日時 (Installed on)」の値が必ず表示されます。製品またはソフトウェアグループの下で展開されているリスト内の項目の場合、その製品のバージョン情報が表示されます。
 - pkgrm コマンドを使って製品のすべてまたは一部を削除した場合は、ソフトウェア製品の名前の横にそれを警告するアイコンが表示されます。

▼ Solaris Product Registry の GUI を使ってソフトウェアをインストールする方法

Product Registry を使用すると、ソフトウェアを検索し、ソフトウェアのインストール手順をガイドしてくれる Solaris Web Start プログラムを起動できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris Product Registry** ツールを起動します。

```
# prodreg
```

「Solaris Product Registry」ウィンドウが表示されます。
3. インストールを **CD**、**DVD**、またはネットワークのいずれから行うかを決めます。次のいずれかの手順に従います。
 - CD からインストールする場合は、CD-ROM ドライブに CD を挿入する。
 - DVD からインストールする場合は、DVD-ROM ドライブに DVD を挿入する。
 - ネットワークからインストールする場合は、インストールするソフトウェアのネットイメージを見つける。

4. インストールおよび登録されているソフトウェアの一覧を表示するには、その左側にあるアイコンをクリックします。
5. 「**Solaris Product Registry**」ウィンドウの下にある「新規インストール (**New Install**)」ボタンをクリックします。
「インストーラの選択 (**Select Installer**)」ダイアログボックスが表示され、最初は /cdrom ディレクトリまたは現在使用中のディレクトリが選択されています。
6. **Solaris Web Start** プログラムのインストーラが見つかるまで、ディレクトリを選択します。
Solaris Web Start インストーラは、CD や DVD 上のさまざまなディレクトリにあります。CD や DVD の構成については、『*Solaris 9 12/03 インストールガイド*』の「Solaris メディアの構成に関するリファレンス情報」を参照してください。
 - Solaris 9 Software 1 of 2 と 2 of 2 CD
 - Solaris 9 Software 2 of 2 CD
 - Solaris 9 Documentation CD
 - Solaris 9 Languages CD。この CD を挿入すると、Solaris Web Start プログラムが自動的に起動される
7. 起動するインストーラを見つけたら、「ファイル (**Files**)」ボックスでそのインストーラの名前を選択します。
8. 「了解 (**OK**)」をクリックします。
選択したインストーラが起動されます。
9. インストーラによって表示される手順に従って、ソフトウェアをインストールします。

▼ Solaris Product Registry の GUI を使ってソフトウェアをアンインストールする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Solaris Product Registry** ツールを起動します。

```
# prodreg
```

「Solaris Product Registry」ウィンドウが表示されます。
3. インストールおよび登録されているソフトウェアの一覧を表示するには、その左側にあるアイコンをクリックします。
4. アンインストールするソフトウェアの名前が見つかるまでディレクトリを選択します。

5. ソフトウェア属性を読んで、このソフトウェアがアンインストールするソフトウェアであることを確認します。
6. 「Solaris Product Registry」ウィンドウの下にある「アンインストール (Uninstall)」ボタンをクリックします。
選択したソフトウェアがアンインストールされます。

Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)

次の作業マップに、Solaris Product Registry のコマンド行インタフェースを使って行うことができるソフトウェアの管理作業を示します。

作業	説明	参照先
インストールまたはアンインストールしたソフトウェアを prodreg を使って表示する	browse サブコマンドを使ってソフトウェア情報を表示できる	304 ページの「インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)」
prodreg を使ってソフトウェア属性を表示する	info サブコマンドを使って特定のソフトウェア属性を表示できる	307 ページの「ソフトウェア属性を表示する方法 (prodreg)」
prodreg コマンドを使ってソフトウェアコンポーネント間の依存関係を確認する	info サブコマンドを使って特定のソフトウェアコンポーネントに依存するコンポーネントを表示できる	309 ページの「ソフトウェアコンポーネント間の依存関係を確認する方法 (prodreg)」
損傷を受けたソフトウェア製品を prodreg を使って特定する	適切なアンインストーラを使用しないでインストール済みのソフトウェアファイルやパッケージを削除すると、システム上のソフトウェアが損傷を受ける可能性がある	310 ページの「損傷を受けたソフトウェア製品を特定する方法 (prodreg)」
prodreg を使ってソフトウェアをアンインストールする	uninstall サブコマンドを使ってソフトウェアをシステムから削除できる	313 ページの「ソフトウェアをアンインストールする方法 (prodreg)」

作業	説明	参照先
損傷を受けたソフトウェアを prodreg を使ってアンインストールする	ソフトウェアコンポーネントのアンインストーラをシステムから削除した場合は、損傷を受けたコンポーネントのアンインストールが失敗することがある	317 ページの「損傷を受けたソフトウェアをアンインストールする方法 (prodreg)」
損傷を受けたソフトウェアコンポーネントを prodreg を使って再インストールする	損傷を受けたソフトウェアコンポーネントに依存するソフトウェアがある場合は、損傷を受けたコンポーネントと依存ソフトウェアをアンインストールするのではなく、損傷を受けたコンポーネントを再インストールできる	320 ページの「損傷を受けたソフトウェアコンポーネントを再インストールする方法 (prodreg)」

prodreg コマンドは、Solaris Product Registry のコマンド行インタフェース (CLI) です。prodreg コマンドは、システム上のソフトウェアを管理するための複数のサブコマンドをサポートしています。

端末ウィンドウで prodreg コマンドを実行して、次の作業を行うことができます。

- インストールおよび登録されているソフトウェアとソフトウェア属性の一覧表示
- System Software Localizations ディレクトリにインストールされている地域対応版のすべての Solaris システム製品の表示
- 損傷を受けたソフトウェアの特定
- Solaris Product Registry のソフトウェアエントリの削除
- ソフトウェアや個々のソフトウェアパッケージのアンインストール

コマンド行インタフェースから Solaris Product Registry を管理する方法については、prodreg (1M) のマニュアルページを参照してください。

▼ インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)

Solaris Product Registry 内のソフトウェアに関する情報を確認するには、端末ウィンドウで prodreg コマンドのサブコマンド browse を実行します。

1. 端末ウィンドウを開きます。
2. **Solaris Product Registry** をブラウズします。

```
% prodreg browse
  BROWSE # +/-/. UUID                               # NAME
=====
```



```

1      -      root                                1 System
                                                Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 4/03
                                                System
                                                Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
                                                Software

```

prodreg のサブコマンド browse では、登録済みソフトウェアに関する次の情報を表示できます。

BROWSE #	<p>prodreg browse コマンドを実行すると、Solaris Product Registry により、登録済みの各ソフトウェアコンポーネントのブラウザ番号が生成されます。この番号を prodreg browse コマンドや prodreg info コマンドの引数に指定することにより、登録済みの特定のコンポーネントの下位コンポーネントを表示できます。</p> <p>注 - ブラウズ番号 (BROWSE#) は、システムのリポートや再インストールによって変更される可能性があります。ブラウザ番号をスクリプト内に記述したり、その他のログインセッションで再利用したりすることはできません。</p>
+/-/.	<p>このフィールドは、ソフトウェアコンポーネントが Solaris Product Registry に登録された子コンポーネントを持っているかどうかを示します。このフィールドには、次の文字が表示されます。</p> <ul style="list-style-type: none"> ■ +- ソフトウェアコンポーネントは、現在表示されていないが子コンポーネントを持っていることを示します。 ■ -- ソフトウェアコンポーネントは、現在表示されている子コンポーネントを持っていることを示します。 ■ .- ソフトウェアコンポーネントが子コンポーネントを持っていないことを示します。
UUID	<p>このフィールドには、Solaris Product Registry に格納されているソフトウェア固有の識別子が表示されます。</p>
#	<p>このフィールドには、システム上のソフトウェアコンポーネントのインスタンス番号が表示されます。システム上に複数のソフトウェアコンポーネントインスタンスがある場合、Solaris Product Registry はそれぞれのインスタンスに別々のインスタンス番号を割り当てます。</p>
NAME	<p>このフィールドには、ソフトウェアの地域対応化された名前が表示されます。出力例では、Solaris オペレーティング環境の名前は「Solaris 9 4/03」になっています。</p>

3. Solaris Product Registry 内の任意のソフトウェアコンポーネント 1 個の情報をブラウザします。

```
% prodreg browse -m "name"
```

-m *name* *name* で名前を指定されたソフトウェア
コンポーネントの情報を表示する

システム上に *name* ソフトウェアのインスタンスが複数存在する場合は、次のコマンドを実行して Solaris Product Registry をブラウズします。

```
% prodreg browse -u name-UUID -i instance
```

-u *name-UUID* 一意の識別子 *name-UUID* で指定された
name ソフトウェアコンポーネントの情
報を表示する

-i *instance* インスタンス番号 *instance* で指定された
name ソフトウェアコンポーネントの情
報を表示する

4. ブラウズするそれぞれのソフトウェアコンポーネントに対して、手順 3 を繰り返します。

例 — コンポーネント名によるソフトウェア情報の表示 (prodreg)

次の例では、コンポーネント名を指定してソフトウェア情報を表示します。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  ===== =====  =====
  1      -      root                                     1  System
                                           Registry
  2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                           System
                                           Software
  3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software

% prodreg browse -m "Solaris 9 4/03 System Software"
```

例 — コンポーネントのブラウズ番号によるソフトウェア情報の表示 (prodreg)

次の例では、`prodreg browse` コマンドの `-n` オプションでコンポーネントのブラウズ番号を指定して、ソフトウェアの情報を表示します。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  ===== =====  =====
  1      -      root                                     1  System
```

```

2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Registry
Solaris 9 4/03
System
Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
Software

```

```
% prodreg browse -n 2
```

例 — コンポーネントの UUID によるソフトウェア情報の表示 (prodreg)

次の例では、`prodreg browse` コマンドの `-u` オプションでコンポーネントの UUID を指定して、ソフトウェアの情報を表示します。

```

% prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====  =====  =====
1          -      root                                     1  System
Registry
2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
System
Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
Software

```

```
% prodreg browse -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ ソフトウェア属性を表示する方法 (prodreg)

`prodreg info` コマンドを使って、特定のソフトウェア属性を表示できます。`prodreg info` コマンドは、登録済みソフトウェアに関するさまざまな情報を表示します。次に例を挙げます。

- ソフトウェアコンポーネントの名前
- ソフトウェアコンポーネントの説明
- ソフトウェアの必須コンポーネント
- このソフトウェアを必要とするその他のコンポーネント
- ソフトウェアのベースディレクトリ
- ソフトウェアコンポーネントのパス

1. 端末ウィンドウを開きます。
2. **Solaris Product Registry** をブラウズします。

```

% prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====  =====  =====

```

```

1      -      root                                1 System
                                                Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 4/03
                                                System
                                                Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
                                                Software

```

3. **Solaris Product Registry** 内の任意のソフトウェアコンポーネントの属性を表示します。

prodreg info コマンドで出力された任意のソフトウェアコンポーネントの属性を表示できます。次のコマンドを使用します。

```
% prodreg info -m "name"
```

```
-m name
```

name で名前を指定されたソフトウェアコンポーネントの属性を表示する

4. 表示するそれぞれのソフトウェアコンポーネントに対して、手順 3 を繰り返します。

例 — コンポーネント名によるソフトウェア属性の表示 (prodreg)

次の例では、コンポーネント名を指定してソフトウェア属性を表示します。

```

% prodreg browse
BROWSE # +/-/.  UUID                                #  NAME
===== =====
1      -      root                                1  System
                                                Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9 4/03
                                                System
                                                Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
                                                Software

% prodreg info -m "Solaris 9 4/03 System Software"

```

例 — コンポーネントのブラウザ番号によるソフトウェア属性の表示 (prodreg)

次の例では、prodreg info コマンドの -n オプションでコンポーネントのブラウザ番号を指定して、ソフトウェアの属性を表示します。

```

% prodreg browse
BROWSE # +/-/.  UUID                                #  NAME
===== =====

```

```

1      -      root                                1 System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 4/03
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
                                           Software

```

```
% prodreg info -n 2
```

例 — コンポーネントの UUID によるソフトウェア属性の表示 (prodreg)

次の例では、`prodreg info` コマンドの `-u` オプションでコンポーネントの UUID を指定して、ソフトウェア属性を表示します。

```

% prodreg browse
BROWSE # +/-/.  UUID                                #  NAME
===== =====
1      -      root                                1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9 4/03
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
                                           Software

```

```
% prodreg info -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ ソフトウェアコンポーネント間の依存関係を確認する方法 (prodreg)

特定のソフトウェアコンポーネントに依存するコンポーネントを表示するには、`prodreg info` コマンドを使用します。特定のコンポーネントをアンインストールする前に、ソフトウェア製品間の依存関係を確認できます。

1. 端末ウィンドウを開きます。
2. **Solaris Product Registry** をブラウズします。

```

% prodreg browse
BROWSE # +/-/.  UUID                                #  NAME
===== =====
1      -      root                                1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9 4/03
                                           System
                                           Software

```

```
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
                                     Software
```

Solaris Product Registry に確認するソフトウェアコンポーネントが表示されるまで `prodreg browse` コマンドを繰り返し実行します。 `prodreg browse` コマンドを使って Solaris Product Registry をブラウズする方法については、304 ページの「インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)」を参照してください。

3. 特定のソフトウェアコンポーネントの依存関係を表示します。

```
% prodreg info -m "name" -a "Dependent Components"
```

-m *name* *name* で名前を指定されたソフトウェアコンポーネントの属性を表示する

-a "Dependent Components" Dependent Components 属性の値を表示することにより、*name* ソフトウェアに依存するコンポーネントを表示する

このコマンドは、*name* ソフトウェアに依存するソフトウェアコンポーネントの一覧を出力します。

例 — その他のソフトウェア製品に依存するコンポーネントの表示 (prodreg)

次の例では、ExampleSoft というソフトウェア製品に依存するコンポーネントを一覧表示します。

```
% prodreg -m "ExampleSoft" -a "Dependent Components"
```

```
Dependent Components:
```

Name	UUID	#
-----	-----	-
ExampleSoftA	7f49ecvb-1ii2-11b2-a3f1-0800119u7e8e	1

▼ 損傷を受けたソフトウェア製品を特定する方法 (prodreg)

適切なアンインストーラを使用しないでインストール済みのソフトウェアファイルやパッケージを削除すると、システム上のソフトウェアが損傷を受ける可能性があります。損傷を受けたソフトウェアは、正しく機能しない場合があります。 `prodreg info` コマンドを使って、ソフトウェア製品が損傷を受けていないかどうかを確認できます。

1. 確認するソフトウェアの Solaris Product Registry 情報を表示します。

```
% prodreg browse -m name
```

```
BROWSE # +/-/. UUID # NAME
```

```

=====  =====  =====  =====  =====
1          -          root                               1  System
Registry
2          +          a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9 8/03
System
Software
3          +          8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
Software
4          -          name-UUID                          1  name
233        .          component-a-pkg                    1  component-a
234        .          component-b-pkg                    1

```

`-m name` *name* で名前を指定されたソフトウェアコンポーネントの情報を表示する

`name-UUID` *name* ソフトウェアコンポーネントのUUIDを指定する

`component-a-pkg` *name* ソフトウェアに依存する *component-a* コンポーネントのパッケージ名を指定する

`component-a` *name* ソフトウェアに依存するコンポーネントの名前を指定する

`component-b-pkg` *name* ソフトウェアに依存する *component-b* コンポーネントのパッケージ名を指定する

前出の出力例では、Name 列に *component-b-pkg* エントリの名前が表示されていません。Solaris Product Registry にソフトウェアコンポーネント名が表示されない場合、そのコンポーネントは損傷を受けている可能性があります。

2. ソフトウェアコンポーネントが損傷を受けていないかどうかを検証します。

```

% prodreg info -u name-UUID -i 1 -d
isDamaged=TRUE

```

`-u name-UUID` *name* ソフトウェアコンポーネントの情報を表示する

`-i 1` *name* ソフトウェアコンポーネントの1番目のインスタンスの情報を表示する

`-d` *name* ソフトウェアコンポーネントの *isDamaged* 属性の値を表示する

isDamaged=TRUE と出力された場合、*name* ソフトウェアコンポーネントは損傷を受けています。

3. *name-UUID* ソフトウェアコンポーネントを構成するパッケージを特定します。

```

% prodreg info -u name-UUID -i 1 -a PKGS
pkgs:
component-a-pkg component-b-pkg

```

4. これらのパッケージがシステムにインストールされていることを確認します。

```
% pkginfo component-a-pkg
application component-a-pkg component-a
```

```
% pkginfo component-b-pkg
ERROR: information on "component-b-pkg" was not found
```

pkginfo component-b-pkg コマンドのエラーメッセージは、component-b-pkg パッケージがシステムから削除されたことを示しています。name ソフトウェアコンポーネントは、component-b-pkg パッケージがないと正しく機能しない可能性があります。

例 — 損傷を受けたソフトウェアコンポーネントの特定 (prodreg)

次の例では、ExampleSoft ソフトウェアが損傷を受けていないかどうかを確認します。

```
% prodreg browse -m Examplesoft
BROWSE # +/-/.  UUID                                     #  NAME
=====  =====  =====
1         -      root                                     1  System
                                           Registry
2         +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 8/03
                                           System
                                           Software
3         +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
4         -      95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
233      .      90209809-9785-b89e-c821-0472985982be  1  Example Doc
234      .      EXSOztt                                     1
235      .      EXSOblob                                    1  Example Data
```

Solaris Product Registry の NAME 列で、ExampleSoft の子コンポーネント EXSOztt のエントリがありません。ExampleSoft ソフトウェアは損傷を受けている可能性があります。prodreg info コマンドに -u、-i、および -d オプションを指定して、ExampleSoft ソフトウェアが損傷を受けているかどうかを確認します。

```
% prodreg info -u 95842091-725a-8501-ef29-0472985982be -i 1 -d
isDamaged=TRUE
```

前出のコマンドの出力結果から、ExampleSoft ソフトウェアが損傷を受けていることがわかります。prodreg info コマンドの -a PKGS オプションを実行して、ExampleSoft ソフトウェアパッケージを特定します。

```
% prodreg info
-u 95842091-725a-8501-ef29-0472985982be
-i 1 -a PKGS
pkgs:
EXSOztt EXSOblob
```


pkginfo コマンドを実行して、EXSOztt および EXSOblob パッケージがシステムにインストールされていることを確認します。

```
% pkginfo EXSOztt
ERROR: information for "EXSOztt" was not found
```

```
% pkginfo EXSOblob
application EXSOblob          Example Data
```

pkginfo コマンドの出力結果から、EXSOztt パッケージがシステムにインストールされていないことがわかります。

▼ ソフトウェアをアンインストールする方法 (prodreg)

prodreg uninstall コマンドを使って、システムからソフトウェアを削除できます。prodreg uninstall コマンドを使ってソフトウェアをアンインストールすると、指定のソフトウェアとそのソフトウェアに関連付けられている子コンポーネントがすべて削除されます。アンインストールするソフトウェアに依存するソフトウェアがないことを確認してから、ソフトウェアを削除してください。ソフトウェアの依存関係を確認する方法については、309 ページの「ソフトウェアコンポーネント間の依存関係を確認する方法 (prodreg)」を参照してください。

ソフトウェアコンポーネントのアンインストールが完了したら、prodreg unregister -r コマンドを使って、ソフトウェアそのものと、その子コンポーネント全部を Solaris Product Registry から削除できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. アンインストールするソフトウェアの情報を表示します。

```
# prodreg browse -u name-UUID
BROWSE # +/-/. UUID                                     # NAME
===== =====
1      -      root                                       1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 8/03
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
1423   -      name-UUID                                   1  name
1436   .      component-a-UUID                           1  component-a
1437   -      component-b-UUID                           1  component-b
1462   .      component-c-UUID                           1  component-c

-u name-UUID
```

一意の識別子 *name-UUID* で指定されたソフトウェアコンポーネントの情報を表示する

<i>name</i>	一意の識別子 <i>name-UUID</i> でアンインストールするソフトウェアコンポーネントの名前を指定する
<i>component-a-UUID</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-a</i> の一意の識別子を指定する
<i>component-a</i>	<i>name</i> ソフトウェアによって必要とされているコンポーネントの名前を指定する
- <i>component-b-UUID</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-b</i> の一意の識別子を指定する。ハイフン (-) は、 <i>component-b</i> がその他のソフトウェアコンポーネントを必要としていることを表す
<i>component-b</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネントの名前を指定する
<i>component-c-UUID</i>	<i>component-b</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-c</i> の一意の識別子を指定する
<i>component-c</i>	<i>component-b</i> ソフトウェアによって必要とされているソフトウェアコンポーネントの名前を指定する

3. ソフトウェアをアンインストールします。

```
# prodreg uninstall -u name-UUID
```

4. アンインストールするソフトウェアの依存関係を確認します。

```
# prodreg info -u name-UUID
Title: name
.
.
.
Child Components:
Name                               UUID                               #
-----
component-a                         component-a-UUID                   1
component-b                         component-b-UUID                   1

Required Components:
Name                               UUID                               #
-----
component-a                         component-a-UUID                   1
component-b                         component-b-UUID                   1
```

prodreg info コマンドによって出力された次の情報を確認します。

- 「Child Components」 - *name* ソフトウェアコンポーネントに関連付けられたソフトウェアコンポーネントを一覧表示します。 *name* ソフトウェアの登録を解除すると、その子コンポーネントの登録も解除されます。前出の prodreg info コマンドでは、すべての子コンポーネントが一覧表示されます。これらの子コンポーネントの登録を解除してもよいかどうか確認してください。
- 「Required Components」 - *name* ソフトウェアによって必要とされているソフトウェアコンポーネントを一覧表示します。ソフトウェアコンポーネントは、子コンポーネント以外のコンポーネントを必要とする場合があります。コンポーネントとともにアンインストールされ、登録を解除されるのは、子コンポーネントだけです。
- 「Dependent Components」 - *name* ソフトウェアがないと実行できないコンポーネントを一覧表示します。 *name* ソフトウェアの登録を解除すると、その依存コンポーネントの登録も解除されます。前出の prodreg info コマンドでは、すべての依存コンポーネントが一覧表示されます。これらの依存コンポーネントの登録を解除してもよいかどうか確認してください。

前出の出力例では、 *name* ソフトウェアに依存コンポーネントはありません。

5. *name* ソフトウェアの子コンポーネントの依存関係を確認します。

```
# prodreg info -u component-a-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
name                                name-UUID                          1

# prodreg info -u component-b-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
name                                name-UUID                          1

# prodreg info -u component-c-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
component-b                         component-b-UUID                    1
```

前出の出力例では、 *name* ソフトウェアの子コンポーネントに依存するソフトウェアはありません。

6. ソフトウェアコンポーネントと子コンポーネントの登録を解除します。

```
# prodreg unregister -r -u name-UUID -i 1

-r                               一意の識別子 name-UUID を指定して、
                               ソフトウェアとその子コンポーネント全
                               部の登録を再帰的に解除する

-u name-UUID                     登録を解除するソフトウェアを表す一意
                               の識別子を指定する
```

例 — ソフトウェアコンポーネントのアンインストール (prodreg)

次の例では、ExampleSoft ソフトウェアとその子コンポーネントすべてをアンインストールします。

```
# prodreg browse -m "ExampleSoft"
BROWSE # +/-/. UUID # NAME
===== =====
1 - root 1 System
Registry
2 + a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 8/03
System
Software
3 + 8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
Software
1423 - 95842091-725a-8501-ef29-0472985982be 1 ExampleSoft
1436 . 90209809-9785-b89e-c821-0472985982be 1 Example Doc
1437 - EXSOztt 1 Example Data
1462 . EXSOblob 1 Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1

# prodreg info -u 95842091-725a-8501-ef29-0472985982be
Title: ExampleSoft Software
.
.
Child Components:
Name UUID #
-----
Example Doc 90209809-9785-b89e-c821-0472985982be 1
Example Data EXSOztt 1

Required Components:
Name UUID #
-----
Example Doc 90209809-9785-b89e-c821-0472985982be 1
Example Data EXSOztt 1

# prodreg info -u 90209809-9785-b89e-c821-0472985982be -i 1
-a "Dependent Components"
Dependent Components:
Name UUID #
-----
ExampleSoft 95842091-725a-8501-ef29-0472985982be 1

# prodreg info -u EXSOztt -i 1 -a "Dependent Components"
```

```

Dependent Components:
Name                               UUID                               #
-----
ExampleSoft                        95842091-725a-8501-ef29-0472985982be 1

# prodreg info -u EXSOblob -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example Data                        EXSOztt                            1

# prodreg unregister -r -u 95842091-725a-8501-ef29-0472985982be -i 1

```

▼ 損傷を受けたソフトウェアをアンインストールする方法 (prodreg)

損傷を受けたソフトウェアコンポーネントを `prodreg uninstall` コマンドでアンインストールしようとする、失敗します。これは、ソフトウェアコンポーネントのアンインストールプログラムがシステムから削除されている場合に発生する現象です。

システム上にソフトウェアコンポーネントのアンインストールプログラムがない場合は、次の手順に従ってアンインストールを行なってください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. アンインストールするソフトウェアの情報を表示します。

```

# prodreg browse -m "name"
BROWSE # +/-/.  UUID                               #  NAME
=====  =====  =====
1          -      root                               1  System
Registry
2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9 8/03
System
Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
Software
4          -      UUID                               1  name
1436      .      component-a-UUID                   1  component-a
1437      .      component-b-UUID                   1

```

`-m name`

アンインストールする `name` ソフトウェアコンポーネントの情報を表示する

`UUID`

アンインストールするソフトウェアコンポーネントの `UUID` を指定する

<i>component-a-UUID</i>	<i>component-a</i> ソフトウェアコンポーネントの UUID を指定する
<i>component-a</i>	<i>name</i> ソフトウェアの子ソフトウェアコンポーネントの名前を指定する
<i>component-b-UUID</i>	<i>name</i> ソフトウェアの子ソフトウェアコンポーネントの UUID を指定する

前出の出力例では、*component-b-UUID* エントリにコンポーネント名が関連付けられていません。このように名前がない場合、このコンポーネントは損傷を受けている可能性があります。

3. ソフトウェアをアンインストールします。

```
# prodreg uninstall -u UUID -i 1
The install program requested could not be found
```

-u <i>UUID</i>	アンインストールするソフトウェアコンポーネントの UUID を指定する
-i 1	アンインストールするソフトウェアのインスタンスを指定する

前出のエラーメッセージは、システム上にアンインストールプログラムが存在しないことを示しています。

4. ソフトウェアコンポーネントのアンインストールプログラムを特定します。

```
# prodreg info -m "name" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
uninstaller-location uninstall_name
```

-m <i>name</i>	<i>name</i> ソフトウェアコンポーネントの情報を表示する
-a <i>uninstallprogram</i>	<i>name</i> ソフトウェアコンポーネントに関連付けられたアンインストールプログラムの情報を表示する
<i>uninstaller-location</i>	<i>name</i> ソフトウェアコンポーネントのアンインストールプログラムの登録場所を指定する

5. 登録場所にアンインストーラが存在するかどうか確認します。

```
# ls uninstaller-location
uninstaller-location:
No such file or directory
```

ls(1) コマンドの出力結果から、登録場所にアンインストールプログラムが存在しないことがわかります。

6. システムからソフトウェアを削除します。

次のいずれかの方法で削除できます。

- システムのバックアップを使用できる場合は、次の手順に従います。
 - a. バックアップからアンインストールプログラムをロードします。
 - b. 端末ウィンドウなどのシェルコマンド行インタフェースからアンインストールプログラムを実行します。
- バックアップのアンインストールプログラムにアクセスできない場合は、次の手順に従います。
 - a. ソフトウェアコンポーネントの登録を解除します。


```
# prodreg unregister -u UUID -i 1
```
 - b. 削除するソフトウェアによって必要とされている登録済みコンポーネントをすべて削除します。


```
# pkgrm component-a-UUID
```

例 — 損傷を受けたソフトウェアのアンインストール (prodreg)

次の例では、損傷を受けた ExampleSoft ソフトウェアをアンインストールします。この例では、システムバックアップ上のアンインストールプログラムにはアクセスできないものとして扱います。

```
# prodreg browse -m Examplesoft
BROWSE # +/-/.  UUID                                     # NAME
===== =====
1      -      root                                     1 System
Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 8/03
System
Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
Software
4      -      95842091-725a-8501-ef29-0472985982be 1 ExampleSoft
233    .      90209809-9785-b89e-c821-0472985982be 1 Example Doc
234    .      EXSOzzt                                     1
235    .      EXSOblob                                    1 Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
The install program requested could not be found

# prodreg info -m "ExampleSoft" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleSoft/987573587 uninstall_ExampleSoft

# ls /var/sadm/prod/org.example.ExampleSoft/987573587
/var/sadm/prod/org.example.ExampleSoft/987573587:
No such file or directory

# prodreg unregister -u 95842091-725a-8501-ef29-0472985982be -i 1
```

```
# pkgrm EXSOblob
```

▼ 損傷を受けたソフトウェアコンポーネントを再インストールする方法 (prodreg)

損傷を受けたソフトウェアコンポーネントに依存するソフトウェアがある場合は、損傷を受けたコンポーネントを再インストールできます。損傷を受けたコンポーネントと依存ソフトウェアをアンインストールする必要はありません。prodreg unregister コマンドの -f オプションを実行して、損傷を受けたコンポーネントの登録を強制的に解除し、インストールし直します。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 再インストールするソフトウェアの情報を表示します。

```
# prodreg browse -m "name"
BROWSE # +/-/. UUID # NAME
===== =====
1 - root 1 System
Registry
2 + a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 8/03
System
Software
3 + 8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
Software
4 . UUID 1 name
```

-m *name* 再インストールする *name* ソフトウェア
コンポーネントの情報を表示する

UUID 再インストールするソフトウェアコン
ポーネントの *UUID* を指定する

3. 再インストールするソフトウェアに依存するソフトウェアを特定します。

```
# prodreg info -m "name" -a "Dependent Components"
Dependent Components:
Name UUID #
-----
component-a component-a-UUID 1
```

-m *name* 再インストールするソフトウェアコン
ポーネントの名前を *name* で指定する

-a "Dependent Components" *name* ソフトウェアに依存するコンポー
ネントを表示する

component-a *name* ソフトウェアに依存するソフト
ウェアコンポーネントの名前を指定する

component-a-UUID

component-a ソフトウェアコンポーネントの UUID を指定する

component-a ソフトウェアコンポーネントは、再インストールするソフトウェアに依存しています。*component-a* の登録を解除しないで *name* ソフトウェアを再インストールする場合は、*name* ソフトウェアの登録を強制的に解除したあと、*name* ソフトウェアを再インストールします。

- 再インストールするソフトウェアコンポーネントだけの登録を解除します。

```
# prodreg unregister -f -u UUID
```

- ソフトウェアコンポーネントを再インストールします。

```
# /usr/bin/java -cp /usr/installers/installer
```

installer

name ソフトウェアのインストールプログラムの名前を指定する

例 — 損傷を受けたソフトウェアコンポーネントの再インストール (prodreg)

次の例では、損傷を受けたソフトウェアコンポーネント *ComponentSoft* を、依存コンポーネントである *ExampleSoft* の登録解除やアンインストールを行うことなく再インストールします。

```
# prodreg browse -m "ComponentSoft"
BROWSE # +/-/. UUID # NAME
===== =====
1 - root 1 System Registry
2 + a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 8/03 System Software
3 + 8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified Software
4 . 86758449-554a-6531-fe90-4352678362fe 1 ComponentSoft

# prodreg info -m "ComponentSoft" -a "Dependent Components"
Dependent Components:
Name UUID #
-----
ExampleSoft 95842091-725a-8501-ef29-0472985982be 1

# prodreg unregister -f -u 86758449-554a-6531-fe90-4352678362fe -i 1

# /usr/bin/java -cp /usr/installers/org.example.componentsoft
```

署名付きパッケージの追加と削除 (作業マップ)

次の作業マップは、署名付きパッケージの追加および削除に関する作業を示したものです。

作業	説明	参照先
証明書をインポートする	pkgadm addcert コマンドを使って信頼される証明書をインポートする	322 ページの「信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)」
(省略可能) 1 つ以上の証明書の詳細情報を表示する	pkgadm listcert コマンドを使って証明書の詳細情報を表示する	324 ページの「証明書の情報を表示する方法 (pkgadm listcert)」
(省略可能) 証明書を削除する	pkgadm removecert コマンドを使って証明書を削除する	325 ページの「証明書を削除する方法 (pkgadm removecert)」
(省略可能) プロキシサーバーをセットアップする	プロキシを備えたファイアウォールの背後にシステムが存在している場合、プロキシサーバーを指定する	325 ページの「プロキシサーバーをセットアップする方法」
署名付きパッケージを追加する	ルート証明書がインポートされると、pkgadd コマンドを使って署名付きパッケージを追加できる	326 ページの「署名付きパッケージを追加する方法 (pkgadd)」
(省略可能) 署名付きパッケージを削除する	署名付きパッケージの削除方法は、署名なしパッケージの削除方法と同一である	335 ページの「ソフトウェアパッケージを削除する方法 (pkgrm)」

▼ 信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **Java** キーストア内にルート **CA** 証明書が存在することを確認します。

```
# keytool -storepass storepass -list -keystore certfile
```

<code>keytool</code>	非公開鍵とそれらに対応する公開鍵を認証するための X.509 証明書チェーンの Java キーストア (データベース) を管理する。また、信頼されるエンティティが発行した証明書も管理する。 <code>keytool</code> コマンドの詳細については、『 <i>keytool-Key and Certificate Management Tool</i> 』を参照。
<code>-storepass storepass</code>	Java キーストアの完全性を維持するためのパスワードを指定する。
<code>-list</code>	デフォルトで、証明書の MD5 フィンガープリントを出力する。
<code>-keystore certfile</code>	Java キーストア永続ファイルの名前と場所を指定する。

3. Java キーストア内のルート CA 証明書を一時ファイルへエクスポートします。

```
# keytool -export -storepass storepass -alias gtecybertrustca -keystore
gtecybertrustca -keystore /usr/j2se/jre/lib/security/cacerts -file filename
```

<code>-export</code>	信頼される証明書をエクスポートする。
<code>-storepass storepass</code>	Java キーストアの完全性を維持するためのパスワードを指定する。
<code>-alias gtecybertrustca</code>	信頼される証明書の別名を指定する。
<code>-keystore certfile</code>	キーストアファイルの名前と場所を指定する。
<code>-file filename</code>	エクスポートされた証明書を格納するファイルを指定する。

4. 信頼される証明書をパッケージキーストアにインポートします。

```
# pkgadm addcert -t -f format certfile
```

<code>-t</code>	証明書が信頼される CA 証明書であることを示す。このオプションを指定すると、コマンド出力に証明書の詳細情報が含まれ、ユーザーはその情報の確認を求められる。
<code>-f format</code>	証明書または非公開鍵の形式を指定する。インポートする証明書は、PEM (pem)、バイナリ DER (der) のいずれかの形式を使ってコード化されている必要がある。
<code>certfile</code>	証明書を含むファイルを指定する。

詳細については、`pkgadm` のマニュアルページを参照。

5. 一時ファイルを削除します。

例 — 信頼される証明書をインポートする

次の例は、信頼される証明書をインポートする方法を示しています。この例では、keytool コマンドを使って、Java キーストアからパッケージキーストアへ Sun のルート CA 証明書をインポートしています。

```
# keytool -export -storepass changeit -alias gtecybertrustca -keystore
gtecybertrustca -keystore /usr/j2se/jre/lib/security/cacerts -file
/tmp/root.crt
Certificate stored in file </tmp/root.crt>
# pkgadm addcert -t -f der /tmp/root.crt
Enter Keystore Password: storepass
    Keystore Alias: GTE CyberTrust Root
    Common Name: GTE CyberTrust Root
    Certificate Type: Trusted Certificate
    Issuer Common Name: GTE CyberTrust Root
    Validity Dates:<Feb 23 23:01:00 1996 GMT>-<Feb 23 23:59:00 2006 GMT>
    MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
    SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC...
Trusting certificate <GTE CyberTrust Root>
Type a Keystore protection Password.
Press ENTER for no protection password (not recommended): xxx
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended): xxx
Certificate(s) from </tmp/root.crt> are now trusted
# rm /tmp/root.crt
```

▼ 証明書の情報を表示する方法 (pkgadm listcert)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. パッケージキーストアの内容を表示します。

```
# pkgadm listcert
```

例 — 証明書の情報を表示する (pkgadm listcert)

次の例は、ローカルに格納されている証明書の詳細情報を表示する方法を示しています。

```
# pkgadm listcert -P pass:storepass
    Keystore Alias: GTE CyberTrust Root
    Common Name: GTE CyberTrust Root
    Certificate Type: Trusted Certificate
    Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
    MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
    SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC...
```

▼ 証明書を削除する方法 (pkgadm removecert)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 信頼される証明書をパッケージキーストアから削除します。

```
# pkgadm removecert -n "certfile"
-n "certfile" オプションには、ユーザーの証明書と鍵のペアの別名、信頼される証明書の別名のいずれかを指定します。
```

注 - 証明書の別名を表示するには、`pkgadm listcert` コマンドを使用します。

例 — 証明書を削除する (pkgadm removecert)

次の例は、証明書を削除する方法を示しています。

```
# pkgadm listcert
Enter Keystore Password: storepass
  Keystore Alias: GTE CyberTrust Root
  Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
  Issuer Common Name: GTE CyberTrust Root
  Validity Dates:<Feb 23 23:01:00 1996 GMT>-<Feb 23 23:59:00 2006 GMT>
  MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC...
# pkgadm removecert -n "GTE CyberTrust Root"
Enter Keystore Password: storepass
Successfully removed Certificate(s) with alias <GTE CyberTrust Root>
```

▼ プロキシサーバーをセットアップする方法

プロキシを備えたファイアウォールの背後にシステムが存在している場合、プロキシサーバーをセットアップしてからでないと、HTTP サーバーからパッケージを追加できません。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 次のいずれかの方法を使ってプロキシサーバーを指定します。
 - a. 環境変数 `http_proxy`、`HTTPPROXY`、`HTTPPROXYPORT` のいずれかを使ってプロキシサーバーを指定します。
たとえば、次のようになります。

```
# setenv http_proxy http://mycache.domain:8080
または、次のいずれかを指定します。

# setenv HTTPPROXY mycache.domain
# setenv HTTPPROXYPORT 8080
```

- b. **pkgadd -x** コマンド行でプロキシサーバーを指定します。
たとえば、次のようになります。

```
# pkgadd -x mycache.domain:8080 -d http://myserver.com/pkg SUNWpkg
```

- c. プロキシサーバーの情報を含む管理ファイルを作成します。
たとえば、次のようになります。

```
# cat /tmp/admin
mail=
instance=unique
partial=ask
runlevel=ask
idepend=ask
rdepend=ask
space=ask
setuid=ask
conflict=ask
action=ask
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
basedir=default
proxy=mycache.domain:8080
```

続いて、**pkgadd -a** コマンドでこの管理ファイルを指定します。たとえば、次のようになります。

```
# pkgadd -a /tmp/admin -d http://myserver.com/pkg SUNWpkg
```

▼ 署名付きパッケージを追加する方法 (pkgadd)

次の手順では、Sun のルート CA 証明書がすでにインポートされているものと仮定しています。詳細については、322 ページの「信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)」を参照してください。

プロキシサーバーをセットアップする方法については、325 ページの「プロキシサーバーをセットアップする方法」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 署名付きパッケージを追加します。

```
# pkgadd -d /pathname/package-name
-d device-name オプションには、パッケージのインストール元デバイスを指定します。デバイスとして指定できるのは、ディレクトリ、テープ、フロッピーディスク、リムーバブルディスクのいずれかです。また、pkgtrans コマンドによって作成されたデータストリームを、デバイスとして指定することもできます。
```

例 — 署名付きパッケージを追加する (pkgadd)

次の例は、ダウンロード済みの署名付きパッケージを追加する方法を示しています。

```
# # pkgadd -d /tmp/signed_pppd
The following packages are available:
  1  SUNWpppd      Solaris PPP Device Drivers
                        (sparc) 11.10.0,REV=2003.05.08.12.24

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all
Enter keystore password:
## Verifying signature for signer <User Cert 0>
.
.
.
```

次の例は、デバイス名として HTTP URL を指定して署名付きパッケージをインストールする方法を示しています。指定した URL は、ストリーム形式のパッケージを指している必要があります。

```
# pkgadd -d http://install/signed-video.pkg

## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete
.
.
.
```

パッケージコマンドによるソフトウェア パッケージの管理 (作業マップ)

次の作業マップは、署名付きパッケージと署名なしパッケージの両方に対して、パッケージコマンドで行えるソフトウェア管理作業を示したものです。

作業	説明	参照先
ソフトウェアパッケージをローカルシステムに追加する	pkgadd コマンドを使ってソフトウェアパッケージをローカルシステムに追加できる	328 ページの「ソフトウェアパッケージを追加する方法 (pkgadd)」

作業	説明	参照先
ソフトウェアパッケージをスプールディレクトリに追加する	ソフトウェアを実際にインストールしないで、ソフトウェアパッケージをスプールディレクトリに追加できる	331 ページの「ソフトウェアパッケージをスプールディレクトリに追加する」
インストールされたすべてのソフトウェアパッケージに関する情報を表示する	pkginfo コマンドを使用して、インストールされたパッケージに関する情報を表示できる	333 ページの「インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)」
インストールされたソフトウェアパッケージの整合性を検査する	pkgchk コマンドを使って、インストールされたソフトウェアパッケージの整合性を検査できる	334 ページの「インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)」
ソフトウェアパッケージを削除する	pkgrm コマンドを使って不要になったソフトウェアパッケージを削除できる	335 ページの「ソフトウェアパッケージを削除する方法 (pkgrm)」

▼ ソフトウェアパッケージを追加する方法 (pkgadd)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. インストール済みのパッケージの中で、追加しようとしているものと同じ名前のものがあれば、すべて削除します。

この手順によって、追加および削除されたソフトウェアの正しい記録が取れます。同じアプリケーションの複数のバージョンをシステムで管理する場合があります。ソフトウェアの複数のコピーを管理する方法については、292 ページの「パッケージの削除に関するガイドライン (pkgrm)」を、その作業手順については、335 ページの「ソフトウェアパッケージを削除する方法 (pkgrm)」を参照してください。

3. ソフトウェアパッケージをシステムに追加します。

```
# pkgadd -a admin-file -d device-name pkgid ...
```

`-a admin-file`

(省略可能) インストール時に pkgadd コマンドが参照する管理ファイルを示す。管理ファイルの使用方法については、293 ページの「管理ファイルの使用」を参照。

`-d device-name`

ソフトウェアパッケージへの絶対パスを指定する。`device-name` には、デバイス、ディレクトリ、またはスプールディレクトリへのパスを指定できる。パッケージのあるパスを指定しないと、pkgadd コマンドはデフォルトのスプールディレクトリ (/var/spool/pkg) をチェックする。パッケージがそこ不在の場合、パッケージのインストールは失敗する。

`pkgid`

(省略可能) インストールする1つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、使用可能なすべてのパッケージがインストールされる。

パッケージのインストール中に `pkgadd` コマンドによって問題が検出されると、その問題に関連するメッセージに続いて、次のプロンプトが表示されます。

```
Do you want to continue with this installation?
```

このプロンプトには、`yes`、`no`、または `quit` のいずれかで応答します。複数のパッケージを指定した場合は、`no` と入力して、インストール中のパッケージのインストールを中止してください。他のパッケージのインストールが続行されます。インストールを中止する場合は、`quit` と入力してください。

4. 指定したパッケージが正常にインストールされていることを確認します。

```
# pkgchk -v pkgid
```

エラーがない場合は、インストールされたファイルのリストが返されます。エラーがある場合は、`pkgchk` コマンドによってエラーが報告されます。

例 — マウント済みの CD からソフトウェアパッケージを追加する

次の例は、マウント済みの Solaris 9 CD から `SUNWp15u` パッケージをインストールする方法を示しています。また、パッケージファイルが正しくインストールされているかどうかを確認する方法も示します。

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product SUNWp15u
.
.
.
Installation of <SUNWp15u> was successful.
# pkgchk -v SUNWp15u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.00503
.
.
.
```

例 — リモートパッケージサーバーからソフトウェアパッケージをインストールする

インストールするパッケージがリモートシステムから利用できる場合は、そのパッケージが (パッケージ形式で) 格納されているディレクトリを手動でマウントして、ローカルシステムにパッケージをインストールすることができます。

次の例は、リモートシステムからソフトウェアパッケージをインストールする方法を示しています。この例では、package-server というリモートシステムの /latest-packages ディレクトリにソフトウェアパッケージが格納されているものとします。まず mount コマンドによってパッケージが /mnt にローカルにマウントされ、次に pkgadd コマンドによって SUNWpl5u パッケージがインストールされます。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWpl5u
```

```
·
·
·
```

Installation of <SUNWpl5u> was successful.

オートマウンタがサイトで実行されている場合は、リモートパッケージサーバーを手動でマウントする必要はありません。代わりに、オートマウンタパス (この場合は /net/package-server/latest-packages) を、-d オプションの引数として使用します。

```
# pkgadd -d /net/package-server/latest-packages SUNWpl5u
```

```
·
·
·
```

Installation of <SUNWpl5u> was successful.

次の例は、前の例に似ていますが、-a オプションを使用して noask-pkgadd という管理ファイルを指定している点が異なります。この管理ファイルについては、293 ページの「パッケージ追加時のユーザーの対話操作を省略する (pkgadd)」に説明されています。この例では、noask-pkgadd 管理ファイルがデフォルトの格納場所である /var/sadm/install/admin にあるものとします。

```
# pkgadd -a noask-pkgadd -d /net/package-server/latest-packages SUNWpl5u
```

```
·
·
·
```

Installation of <SUNWpl5u> was successful.

例 — ソフトウェアパッケージを特定の HTTP URL からインストールする

次の例は、特定の HTTP URL をデバイス名として使ってパッケージをインストールする方法を示しています。指定した URL は、ストリーム形式のパッケージを指している必要があります。

```
# pkgadd -d http://install/xf86-4.3.0-video.pkg
```

```
## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete
```

The following packages are available:

```

1 SUNWxf86r      XFree86 Driver Porting Kit (Root)
                  (i386) 4.3.0,REV=0.2003.02.28
2 SUNWxf86u      XFree86 Driver Porting Kit (User)
                  (i386) 4.3.0,REV=0.2003.02.28
.
.
.

```

ソフトウェアパッケージをスプールディレクトリに追加する

使用頻度の高いインストール済みパッケージを、スプールディレクトリにコピーしておくとお便利です。パッケージをデフォルトのスプールディレクトリ `/var/spool/pkg` にコピーする場合、`pkgadd` コマンドを使用するとき、コピー元のパッケージの位置 (`-d device-name` 引数) を指定する必要はありません。`pkgadd` コマンドは、デフォルトで `/var/spool/pkg` ディレクトリを調べて、コマンド行に指定されたパッケージをすべて見つけます。パッケージをスプールディレクトリにコピーすることと、パッケージをシステム上にインストールすることとは異なることに注意してください。

▼ ソフトウェアパッケージをスプールディレクトリに追加する方法 (`pkgadd`)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. スプール済みのパッケージの中で、追加しようとしているものと同じ名前のものがあれば、すべて削除します。
スプール済みのパッケージを削除する方法については、336 ページの「例 — スプールされたソフトウェアパッケージを削除する」を参照してください。
3. ソフトウェアパッケージをスプールディレクトリに追加します。

```
# pkgadd -d device-name -s spooldir pkgid...
```

`-d device-name`

ソフトウェアパッケージへの絶対パスを指定する。
`device-name` には、デバイス、ディレクトリ、またはスプールディレクトリへのパスを指定できる。

`-s spooldir`

パッケージがスプールされるスプールディレクトリの名前を指定する。`spooldir` を指定する必要がある。

`pkgid` (省略可能) スプールディレクトリに追加する1つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、使用可能なすべてのパッケージがコピーされる。

4. 指定したパッケージがスプールディレクトリに正常にコピーされていることを確認します。

```
$ pkginfo -d spooldir | grep pkgid
```

`pkgid` が正常にコピーされると、`pkginfo` コマンドによって、`pkgid` に関する1行の情報が返されます。正常にコピーされなかった場合は、`pkginfo` コマンドによってシステムプロンプトが返されます。

例 — マウント済みの CD からスプールディレクトリを設定する

次の例は、SUNWman パッケージをマウント済みの SPARC Solaris 9 CD からデフォルトのスプールディレクトリ (`/var/spool/pkg`) に転送する方法を示しています。

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

例 — リモートソフトウェアパッケージサーバーからスプールディレクトリを設定する

コピーするパッケージがリモートシステムから利用できる場合は、そのパッケージが(パッケージ形式で)格納されているディレクトリを手動でマウントして、ローカルスプールディレクトリにパッケージをコピーすることができます。

次の例は、このためのコマンドを示しています。この例では、`package-server` というリモートシステムの `/latest-packages` ディレクトリにソフトウェアパッケージが格納されているものとします。まず `mount` コマンドによってパッケージが `/mnt` にローカルにマウントされ、次に `pkgadd` コマンドによって SUNWp15p パッケージが `/mnt` からデフォルトのスプールディレクトリ (`/var/spool/pkg`) にコピーされます。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt -s /var/spool/pkg SUNWp15p
Transferring <SUNWp15p> package instance
```

オートマウントが実行されている場合は、リモートパッケージサーバーを手動でマウントする必要はありません。代わりに、オートマウントパス (この場合は、`/net/package-server/latest-packages`) を、`-d` オプションの引数として使用してください。

```
# pkgadd -d /net/package-server/latest-packages -s /var/spool/pkg SUNWp15p
Transferring <SUNWp15p> package instance
```

例 — デフォルトのプールディレクトリからソフトウェアパッケージをインストールする

次の例は、デフォルトのプールディレクトリから SUNWp15p パッケージをインストールする方法を示しています。オプションを何も指定しないと、pkgadd コマンドは /var/spool/pkg ディレクトリを検索して名前付きのパッケージを見つけます。

```
# pkgadd SUNWp15p
.
.
.
Installation of <SUNWp15p> was successful.
```

インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)

pkginfo コマンドを使用して、インストール済みのパッケージに関する情報を表示します。

```
$ pkginfo
```

例 — インストール済みのすべてのパッケージを表示する

次は、システムがスタンドアロンまたはサーバーのどちらであるかにかかわらず、pkginfo コマンドが、ローカルシステムにインストールされているすべてのパッケージを表示する例です。基本カテゴリ、パッケージ名、およびパッケージの説明が表示されています。

```
$ pkginfo
system      SUNWaccr      System Accounting, (Root)
system      SUNWaccu      System Accounting, (Usr)
system      SUNWadmap     System administration applications
system      SUNWadmc     System administration core libraries
.
.
.
```

例 — ソフトウェアパッケージに関する詳細情報を表示する

```
$ pkginfo -l SUNWcar
PKGINST:  SUNWcar
NAME:     Core Architecture, (Root)
CATEGORY: system
ARCH:    sparc.sun4u
VERSION:  11.9.0,REV=2002.04.06.15.27
```

```

BASEDIR: /
VENDOR: Sun Microsystems, Inc.
DESC: core software for a specific hardware platform group
PSTAMP: crash20020406153633
INSTDATE: Nov 19 2002 14:49
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES:      111 installed pathnames
           36 shared pathnames
           40 directories
           56 executables
           18843 blocks used (approx)

```

▼ インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. インストール済みのパッケージの状態を調べます。

```

# pkgchk -a| -c -v pkgid ...
# pkgchk -d spooldir pkgid ...

```

-a	ファイルの属性と内容 (デフォルト) ではなく、ファイルの属性 (つまりアクセス権) だけを検査することを指定する。
-c	ファイルの内容と属性 (デフォルト) ではなく、ファイルの内容だけを検査することを指定する。
-v	冗長モードを指定する。このモードでは、処理されるたびにファイル名が表示される。
-d spooldir	スプールディレクトリへの絶対パスを指定する。
pkgid	(省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定する。pkgid を指定しない場合は、システムにインストールされているすべてのソフトウェアパッケージが検査される。

例 — インストール済みのソフトウェアパッケージの内容を検査する

次の例は、パッケージの内容を検査する方法を示しています。

```
# pkgchk -c SUNWbash
```

エラーがない場合は、システムプロンプトが返されます。エラーがある場合は、pkgchk コマンドによってエラーが報告されます。

例 — インストール済みのソフトウェアパッケージのファイル属性を検査する

次の例は、パッケージのファイル属性を検査する方法を示しています。

```
# pkgchk -a SUNWbash
```

エラーがない場合は、システムプロンプトが返されます。エラーがある場合は、pkgchk コマンドによってエラーが報告されます。

例 — スプールディレクトリにインストールされているソフトウェアパッケージを検査する

次の例は、スプールディレクトリ (/export/install/packages) にコピーされたソフトウェアパッケージを検査する方法を示しています。

```
# pkgchk -d /export/install/packages
## checking spooled package <SUNWadmap>
## checking spooled package <SUNWadmfw>
## checking spooled package <SUNWadmc>
## checking spooled package <SUNWsadml>
```

注 - スプールされたパッケージの検査には限界があります。これは、パッケージがインストールされないとすべての情報を検査できないからです。

ソフトウェアパッケージを削除する

ソフトウェアパッケージを削除 (アンインストール) するときは、このパッケージの追加 (インストール) 時に使用した関連ツールを使用します。たとえば、Web Start インストーラを使ってインストールしたソフトウェアは、Web Start アンインストーラでアンインストールします。



注意 - ソフトウェアパッケージを削除するのに rm コマンドを使用しないでください。

▼ ソフトウェアパッケージを削除する方法 (pkgrm)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. インストール済みのパッケージを削除します。

```
# pkgrm pkgid...
```

`pkgid` は、削除する 1 つまたは複数のパッケージの名前 (空白で区切られている) を識別します。この引数を省略すると、使用可能なパッケージがすべて削除されます。

例 — ソフトウェアパッケージを削除する

次の例は、パッケージの削除方法を示しています。

```
# pkgrm SUNWctu

The following package is currently installed:
  SUNWctu          Netra ct usr/platform links (64-bit)
                   (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53

Do you want to remove this package? y

## Removing installed package instance <SUNWctu>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
.
.
.
```

例 — スプールされたソフトウェアパッケージを削除する

次の例は、スプールされたパッケージの削除方法を示しています。

```
# pkgrm -s /export/pkg SUNWdmfex.u

The following package is currently spooled:
  SUNWdmfex.u      Sun Davicom 10/100Mb Ethernet Driver (64-bit)
                   (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53

Do you want to remove this package? y
Removing spooled package instance <SUNWdmfex.u>
```

Admintool によるソフトウェアパッケージの追加と削除 (作業マップ)

次の作業マップに、Admintool を使って行うことができるソフトウェアの管理作業を示します。

作業	説明	参照先
Admintool を使ってソフトウェアパッケージを追加する	ソフトウェアパッケージを表示または追加できる	337 ページの「Admintool を使ってソフトウェアパッケージを追加する方法」
Admintool を使ってソフトウェアパッケージを削除する	ソフトウェアパッケージを表示または削除できる	338 ページの「Admintool を使ってソフトウェアパッケージを削除する方法」

Solaris オペレーティングシステムには Admintool が組み込まれています。このツールは、ソフトウェアパッケージの追加と削除など、いくつかの管理作業を実行するためのグラフィカルユーザーインターフェースです。特に、Admintool を使用すると次の作業を行うことができます。

- ソフトウェアパッケージをローカルシステムに追加する
- ソフトウェアパッケージをローカルシステムから削除する
- ローカルシステムにすでにインストールされているソフトウェアを表示する
- インストールされるソフトウェアパッケージをカスタマイズする
- ソフトウェアパッケージの代替インストール用ディレクトリを指定する

▼ Admintool を使ってソフトウェアパッケージを追加する方法

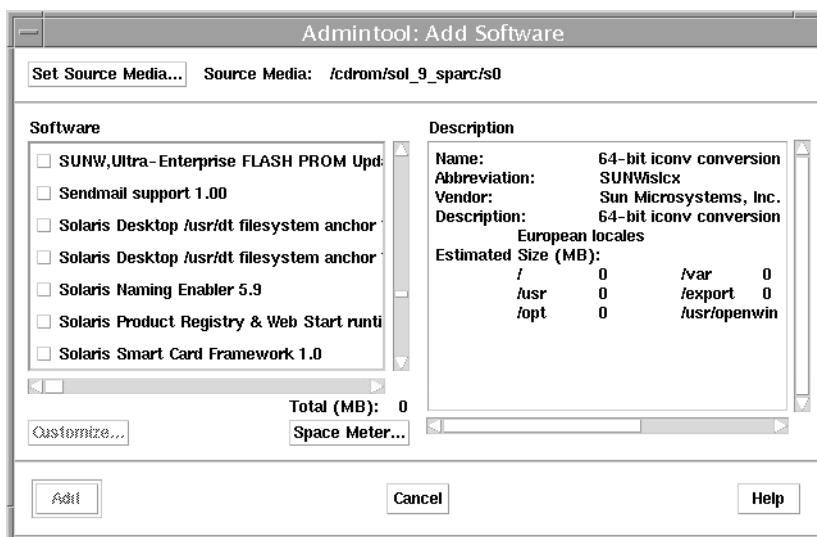
1. スーパーユーザーになります。

Admintool を使ってソフトウェアパッケージを追加または削除する場合、`sysadmin` グループ (グループ 14) のメンバー以外のユーザーは、スーパーユーザーになるか、同等の役割を引き受ける必要があります。

2. **Solaris 9 Software CD** または **DVD** をドライブに挿入します。
ボリュームマネージャが自動的に CD をマウントします。
3. **Admintool** を起動します。

```
# admintool &  
「ユーザー (Users)」ウィンドウが表示されます。
```

4. 「ブラウズ (Browse)」メニューから「ソフトウェア (Software)」を選択します。
「ソフトウェア (Software)」ウィンドウが表示されます。
5. 「編集 (Edit)」メニューから「追加 (Add)」を選択します。
「ソースメディアの設定 (Set Source Media)」ウィンドウが表示されます。必要であれば、インストールメディアへのパスを指定して、「了解 (OK)」をクリックします。デフォルトのパスは、マウント済みの Solaris CD です。
「ソフトウェアを追加 (Add Software)」ウィンドウが表示されます。



6. ローカルシステムにインストールするソフトウェアを選択します。
ウィンドウの「ソフトウェア (Software)」部分で、インストールするソフトウェアに対応するチェックボックスをクリックします。
7. 「追加 (Add)」をクリックします。
パッケージがインストールされるごとに、コマンドツールウィンドウが表示され、インストールの出力が示されます。
「ソフトウェア (Software)」ウィンドウがリフレッシュされて、追加されたばかりのパッケージが表示されます。

▼ Admintool を使ってソフトウェアパッケージを削除する方法

1. スーパーユーザーになります。

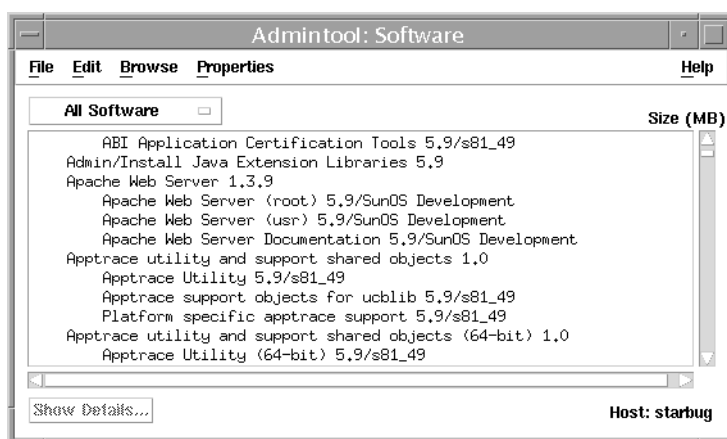
Admintool を使ってソフトウェアパッケージを追加または削除する場合、`sysadmin` グループ (グループ 14) のメンバー以外のユーザーは、スーパーユーザーになるか、同等の役割を引き受ける必要があります。

2. Admintool を起動します。

```
# admintool &
```

「ユーザー (Users)」ウィンドウが表示されます。

3. 「ブラウズ (Browse)」メニューから「ソフトウェア (Software)」を選択します。
「ソフトウェア (Software)」ウィンドウが表示されます。



4. ローカルシステムから削除するソフトウェアを選択します。
5. 「編集 (Edit)」メニューから「削除 (Delete)」を選択します。
本当にソフトウェアを削除するのかどうかを確認するための警告ポップアップウィンドウが表示されます。
6. 「削除 (Delete)」をクリックして、ソフトウェアの削除を確定します。
パッケージが削除されるごとに、コマンドツールウィンドウが表示され、もう一度ソフトウェアを削除するのかどうか確認されます。y、n、またはqを入力します。ソフトウェアを削除することを選択した場合、削除プロセスからの出力が表示されます。

第 24 章

Solaris パッチの管理 (概要)

パッチの管理には、Solaris を実行しているシステムでの Solaris パッチの表示または追加が含まれます。また、不要なパッチや障害の発生したパッチの削除が含まれる場合もあります。パッチの削除は、パッチの「バックアウト」とも呼ばれます。

この章の内容は次のとおりです。

- 341 ページの「パッチについて」
- 342 ページの「署名付きパッチについて」
- 342 ページの「Solaris パッチへのアクセス」
- 344 ページの「Solaris パッチの管理用ツール」

システムにパッチを追加する手順については、349 ページの「Solaris 環境でのパッチ管理 (作業マップ)」を参照してください。

パッチをディスクレスクライアントシステムに追加する方法については、145 ページの「ディスクレスクライアント OS サービスにパッチを適用する」を参照してください。

パッチについて

パッチは、既存のソフトウェアの正常な実行の妨げとなっているファイルとディレクトリを置換または更新するためのファイルとディレクトリの集まりです。既存のソフトウェアと同様に、パッチはアプリケーションバイナリインタフェースに準拠している指定の「パッケージ」形式を元に作成されます。パッケージの詳細については、第 22 章を参照してください。

署名付きパッチについて

「署名付き」パッチとは、デジタル署名の付いたパッチのことです。有効なデジタル署名付きのパッチは、署名が適用された以降はパッチの変更が行われていないことを保証します。署名付きパッチには、パッチをシステムに追加する前に確認できるデジタル署名が含まれています。このため、パッチのダウンロードやパッチの追加をより安全に行うことができます。

Solaris 2.6、7、8、および9 リリースのパッチには、デジタル署名が付属しています。デジタル署名のないパッチ(「署名なしパッチ」)もありますが、最終的にはすべてのパッチが「署名付きパッチ」になる予定です。有効なデジタル署名は、署名が適用された以降にパッチの変更が行われていないことを保証します。

署名付きパッチは Java アーカイブ形式 (JAR) ファイルに格納され、SunSolve OnlineSM から入手できます。

以前の Solaris リリースでは、`smpatch` コマンドと PatchPro を併用することで、署名付きパッチをシステムに追加できました。`smpatch` コマンドの使用手順については、355 ページの「`smpatch` コマンドで署名付きパッチを管理するための準備 (作業マップ)」を参照してください。

この Solaris リリースでは、`patchadd` コマンドを使って署名付きパッチをシステムに追加できます。`patchadd` コマンドの使用手順については、351 ページの「`patchadd` コマンドによる署名付きパッチの追加 (作業マップ)」を参照してください。

署名付きパッチの概要については、285 ページの「署名付きパッケージおよびパッチ」を参照してください。

Solaris パッチへのアクセス

Sun のすべてのユーザーは、SunSolve OnlineSM の Web サイトからパッチにアクセスできます。次の表に、Solaris パッチにアクセスするためのさまざまな方法を示します。

表 24-1 パッチにアクセスする方法

ユーザーの種類	説明
SunSpectrum 契約ユーザー	パッチ情報を格納した SunSolve データベースにアクセスできる。これらのパッチ情報は、SunSolve Online サイトから入手できる。匿名 ftp サービスを利用して入手することもできる。 これらのパッチは、夜間に更新される。
SunSpectrum 契約ユーザー以外	セキュリティパッチの一般セットなどの推奨されるパッチにアクセスできる。これらのパッチ情報は、SunSolve Online から入手できる。

Solaris パッチには、Web サイトから、または匿名 ftp サービスを利用してアクセスできます。

Web サイトからパッチにアクセスする場合は、システムが次の要件を満たしている必要があります。

- インターネットに接続されている
- Netscape™ などの Web ブラウザを実行できる

匿名 ftp を使用してパッチにアクセスする場合は、システムが次の要件を満たしている必要があります。

- インターネットに接続されている
- ftp プログラムを実行できる

次の URL を使って、SunSolve OnlineSM の Web サイトからパッチにアクセスします。

`http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`

推奨されるいくつかのパッチからなるパッチクラスタをインストールすることも、自由に使用できる個々のパッチをインストールすることもできます。パッチレポートも入手できます。

Solaris パッチの番号付け

パッチは、固有の英数字文字列によって識別されます。これは、パッチのベース番号、ハイフン (-)、パッチの改訂バージョン番号の順で構成されています。たとえば、パッチ 108528-10 は SunOS 5.8 カーネル更新用パッチの「パッチ ID」です。

Solaris パッチの管理用ツール

次の表に、Solaris パッチの管理機能の概要を示します。

機能	patchadd/patchrm コマンド	Solaris 2.6、7、および 8 のパッチ管理ツール	Solaris 9 のパッチ管理ツール	PatchPro Interactive または PatchPro Expert
入手方法	Solaris リリースに付属 (SUNWswmt)	http://www.sun.com/PatchPro からダウンロードする必要あり	http://www.sun.com/PatchPro からダウンロードする必要あり	http://www.sun.com/PatchPro からツールを実行
利用できる Solaris リリース	Solaris 2.6、7、8、および 9	Solaris 2.6、7、および 8	Solaris 9	Solaris 2.6、7、8、および 9
署名付きパッチの追加	可。署名付きパッチはダウンロード時に自動的に検証される	可。署名付きパッチはダウンロード時に自動的に検証される	可。署名付きパッチはダウンロード時に自動的に検証される	不可
署名なしパッチの追加	可	不可	可	可
GUI	なし	なし	あり	なし
システムごとに必要なパッチを分析し、署名付きまたは署名なしパッチをダウンロードする機能	なし	あり。署名付きパッチ、署名なしパッチの両方をダウンロード	あり。署名付きパッチ、署名なしパッチの両方をダウンロード	あり。署名なしパッチのみダウンロード
ローカル / リモートのシステムパッチサポート	ローカル	ローカル	ローカルおよびリモート	なし
RBAC サポート	あり	なし	あり	なし

パッチのインストール方法とバックアウト方法の詳細については、patchadd(1M) および patchrm(1M) のマニュアルページを参照してください。各パッチには、パッチ情報が記載されている README ファイルも含まれています。

Solaris パッチ管理ツール

以前の Solaris リリースでは、smpatch コマンドを使って署名付きパッチをシステムに追加できました。

Solaris 2.6、7、および 8 を実行するシステム上の署名付きパッチを管理するには、Solaris Patch Manager Base バージョン 1.0 (smpatch コマンド) を使用します。Solaris 9 を実行するシステム上の署名付きパッチを管理するには、smpatch コマンドと PatchPro 2.1 を併用します。

どちらのパッチツールにも、次の機能があります。

- 必要なパッチを調べ、ローカルシステムに署名付きパッチをダウンロードする。PatchPro Expert と同様に、`/etc/patchpro_hdw.conf` ファイルの内容からインストールされているハードウェアを特定する。どちらのツールも、この機能以外は完全に独立している。
- JAR 形式の署名付きパッチを 1 つ以上適用することにより、追加するパッチ (1 つまたは複数) を認証する。
- 1 つ以上のパッチを削除することにより、パッチ (1 つまたは複数) を削除する前にその依存関係を検査する。
- デフォルトパッチポリシーを設定することにより、`clientroot`、`clientusr`、`rebootafter`、`standard` をはじめとする多種多様なパッチをインストールできる。
- Solaris 9 リリースにアップグレードすると、自動的に最新バージョンの smpatch コマンドを使用できるようになる。

patchadd コマンドは、Solaris 2.6、7、8、および 9 を実行するシステムに署名なしパッチを追加する場合にも使用します。Patch Manager Base バージョン 1.0 は、この目的では使用できません。

Solaris 2.6、7、または 8 の署名付きパッチツールを使用する際の制限事項

Solaris 2.6、7、および 8 の署名付きパッチツールには、次の制限が課されています。

- 代替ブート環境やディスクレスクライアントには、署名付きパッチをインストールできない。
- デジタル署名のないパッチはインストールできない。
- `rebootimmediate`、`reconfigimmediate`、または `nonconforming` 属性を持つパッチはインストールできない。

Solaris パッチ管理ツール (smpatch) のパッケージ要件

パッチ管理ツール (smpatch コマンド) をインストールすると、ツールの実行に必要な複数の Solaris パッケージ (Java パッケージなど) がシステムに追加されます。パッチツールのインストール前にインストールしておかなければならないパッケージもあります。該当するパッケージは、次のとおりです。

- **Solaris 2.6** リリース - Core クラスタと、SUNWmfrun、SUNWlibc、および SUNWxcu4 パッケージ

- **Solaris 7** および **8** リリース – Core クラスタと、SUNWmfrun および SUNWlibc パッケージ
- **Solaris 9** リリース – Solaris 管理コンソールのパッチツールと PatchPro 2.1 を使用する場合は開発者クラスタ (SUNWprog) が必須

システムに必要な Solaris パッケージがインストールされているかどうかを検証する方法については、357 ページの「署名付きパッチツールのパッケージ要件を確認する方法 (smpatch)」を参照してください。

Solaris パッチ管理ツール (smpatch) のダウンロード

Solaris パッチ管理ツールは、次の URL からダウンロードできます。

<https://www.sun.com/PatchPro>

使用している Solaris リリースのリンクをたどり、適切な tar ファイルを選択してください。

署名付きパッチの追加方法の選択

パッチ管理ツールのインストールが完了したら、システムに署名付きまたは署名なしパッチをダウンロードし、追加できます。これには、複数の方法が存在します。次の表を参考にして、現在の要件にもっとも適した方法を選択してください。

コマンドまたはツール	説明	参照先
patchadd	Solaris 9 12/03 リリースのみ -- パッケージキーストアをセットアップした後、このコマンドを使って署名付きパッチをシステムに追加する	patchadd (1M)
smpatch update	必要なパッチを特定する。これらのパッチのダウンロードと追加は自動的に行われる	smpatch (1M)
smpatch analyze	現在のシステムに必要なパッチを特定し、パッチ ID を一覧表示する。これらのパッチをシステムにダウンロードし、追加するには、smpatch download および smpatch add コマンドを使用する	smpatch (1M)
smpatch download および smpatch add	システムにパッチ (1 つまたは複数) をダウンロードし、追加する。これらのコマンドで、必須パッチのダウンロードおよび追加も可能	363 ページの「例 — 署名付きパッチをダウンロードし、追加する (smpatch)」
ftp および smpatch add	ftp コマンドで、システムにパッチ (1 つまたは複数) を転送する。次に、smpatch add コマンドで、システムにパッチを追加する	363 ページの「例 — 署名付きパッチをダウンロードし、追加する (smpatch)」

コマンドまたはツール	説明	参照先
Solaris 管理コンソールのパッチツール	Solaris 9 システムのみ – GUI 機能を使って署名付きパッチを管理したい場合にのみ、このツールを使用する	Solaris 管理コンソールのオンラインヘルプ

第 25 章

Solaris パッチの管理 (手順)

この章では、Solaris が動作するシステム上でパッチを管理する手順について説明します。

この章では、次の内容について説明します。

- 349 ページの「Solaris 環境でのパッチ管理 (作業マップ)」
- 351 ページの「patchadd コマンドによる署名付きパッチの追加 (作業マップ)」
- 355 ページの「smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)」
- 362 ページの「smpatch コマンドによる署名付きパッチの管理 (作業マップ)」
- 369 ページの「署名のない Solaris パッチの管理 (作業マップ)」

Solaris 環境での基本的なパッチの管理方法については、第 24 章を参照してください。

smpatch コマンドに関する問題の障害追跡については、<http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq> を参照してください。

Solaris 環境でのパッチ管理 (作業マップ)

Solaris 環境でのすべてのパッチ管理作業を次のマップに一覧表示します。各作業は、署名付きパッチや署名のないパッチの管理をはじめとする一連の追加作業を示しています。

作業	説明	参照先
パッチのディスク容量の要件を特定する	システムのディスク容量が十分であるかどうかを確認し、パッチをダウンロードまたはスプールする。	350 ページの「パッチの追加に必要なディスク容量があるかどうかを特定する」
署名付きパッチと署名のないパッチのどちらを追加するかを判断する	現在の環境にとって署名付きパッチと署名のないパッチのどちらが適しているかを判断する。	351 ページの「署名付きパッチまたは署名のないパッチの選択」
システムに署名付きパッチまたは署名のないパッチを追加する	署名付きパッチを追加するには、次のいずれかのコマンドを使用する。	
	Solaris 9 12/03 リリースでは patchadd コマンドを使用する。	354 ページの「署名付き Solaris パッチのダウンロードと追加を自動で行う方法 (patchadd)」
	Solaris 2.6、7、8、または 9 リリースでは smpatch コマンドを使用する。	
	smpatch コマンド – この方法が使えるようにシステムを準備する。	355 ページの「smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)」
	smpatch コマンド – システムに署名付きパッチを追加する。	362 ページの「smpatch コマンドによる署名付きパッチの管理 (作業マップ)」
	システムに署名のないパッチを追加する。	369 ページの「署名のない Solaris パッチの管理 (作業マップ)」

パッチの追加に必要なディスク容量があるかどうかを特定する

パッチのダウンロードまたはスプールを開始する前に、ディスク容量に関する次の要件を確認してください。

- 署名付きパッチのデフォルトのダウンロードディレクトリが `/var/sadm/spool` である。署名のないパッチも `/var/sadm/spool` に格納される。
- パッチのダウンロード時には、予想以上に多くのディスク容量が消費される可能性がある。たとえば、ダウンロードしようとするパッチがその他の必須パッチを要求した結果、複数のパッチがダウンロードされる場合など。
- 署名付きパッチは、`/var/sadm/spool` ディレクトリ内に展開されたあと、インストールされる。このため、`/var` ディレクトリには、十分なディスク容量を確保しておく必要がある。

- /var ディレクトリのディスク容量が署名付きパッチのダウンロードおよび展開に必要な容量に満たない場合は、`smpatch -d` コマンドを使って代替パッチダウンロードディレクトリを指定できる。
- パッチをダウンロードし、システムに追加したあと、これらを /var/sadm/spool ディレクトリから削除することにより、/var ディレクトリ内のディスク容量に空きを作ることができる。

署名付きパッチまたは署名のないパッチの選択

署名付きパッチと署名のないパッチのどちらを追加するかは、パッチのダウンロード時にセキュリティ保護を適用する必要があるかどうかによって決定します。パッチのダウンロード時にセキュリティ保護を適用する必要がある場合は、署名付きパッチを追加します。

patchadd コマンドによる署名付きパッチの追加 (作業マップ)

作業	説明	参照先
1. パッケージキーストアをセットアップする	Sun のルート CA 証明書をパッケージキーストアにインポートする	352 ページの「信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)」
2. 署名付きパッチをダウンロードし、追加する	署名付きパッチをダウンロードし、それを patchadd コマンドを使ってシステムに追加する際に、次のいずれかの方法を選択する	
	署名付き Solaris パッチのダウンロードと追加を手動で行う	353 ページの「署名付き Solaris パッチのダウンロードと追加を手動で行う方法 (patchadd)」
	署名付き Solaris パッチのダウンロードと追加を自動で行う	354 ページの「署名付き Solaris パッチのダウンロードと追加を自動で行う方法 (patchadd)」
3. 署名付きパッチを追加する	patchadd コマンドを使って署名付きパッチを追加する	359 ページの「Java キーストアに Sun の証明書をインポートする方法」

信頼される証明書をパッケージキーストアにインポートする方法 (pkgadm addcert)

patchadd コマンドを使って署名付きパッチをシステムに追加するには、署名付きパッチ上の署名を検証できるように、Sun のルート CA 証明書を事前に追加しておく必要があります。この証明書は、Java キーストアからパッケージキーストアへインポートできます。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. Java キーストア内のルート CA 証明書を一時ファイルへエクスポートします。たとえば、次のようになります。

```
# keytool -export -storepass changeit -alias gtecybertrustca -keystore
gtecybertrustca -keystore /usr/j2se/jre/lib/security/cacerts -file
/tmp/root.crt
Certificate stored in file </tmp/root.crt>
```

--export	信頼される証明書をエクスポートする。
-storepass <i>storepass</i>	Java キーストアの完全性を維持するためのパスワードを指定する。
-alias gtecybertrustca	信頼される証明書の別名を指定する。
-keystore <i>certfile</i>	キーストアファイルの名前と場所を指定する。
-file <i>filename</i>	エクスポートされた証明書を格納するファイルを指定する。

3. 一時ファイル内のルート CA 証明書をパッケージキーストアにインポートします。たとえば、次のようになります。

```
# pkgadm addcert -t -f der /tmp/root.crt
Enter Keystore Password: storepass
  Keystore Alias: GTE CyberTrust Root
  Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
  Issuer Common Name: GTE CyberTrust Root
  Validity Dates: <Feb 23 23:01:00 1996 GMT>-<Feb 23 23:59:00 ...
  MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91...

Are you sure you want to trust this certificate? yes
Trusting certificate <GTE CyberTrust Root>
Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
Certificate(s) from </tmp/root.crt> are now trusted
```


<code>--t</code>	証明書が信頼される CA 証明書であることを示す。このオプションを指定すると、コマンド出力に証明書の詳細情報が含まれ、ユーザーはその情報の確認を求められる。
<code>-f format</code>	証明書または非公開鍵の形式を指定する。インポートする証明書は、PEM (<code>pem</code>)、バイナリ DER (<code>der</code>) のいずれかの形式を使ってコード化されている必要がある。
<code>certfile</code>	証明書を含むファイルを指定する。

4. 証明書の情報を表示します。
たとえば、次のようになります。

```
# pkgadm listcert -P pass:storepass
Keystore Alias: GTE CyberTrust Root
Common Name: GTE CyberTrust Root
Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
Validity Dates: <Feb 23 23:01:00 1996 GMT>-<Feb 23 23:59:00 2006 GMT>
MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:
BC:65:A6:89:64
```

5. 一時ファイルを削除します。
たとえば、次のようになります。

```
# rm /tmp/root.crt
```

▼ 署名付き Solaris パッチのダウンロードと追加を手動で行う方法 (`patchadd`)

次の手順は、署名付き Solaris パッチのダウンロードと追加を別個に手動で行いたい場合に使用できます。

この手順では、パッケージキーストアのセットアップが完了しているものと仮定しています。

1. (省略可能) パッチを適用するシステムにログインします。
あるいは、パッチをダウンロードし、`ftp` コマンドを使用してそのパッチをターゲットのシステムにコピーすることもできます。
2. **Web** ブラウザを開き、**SunSolve Online** のサイトに移動します。
`http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`
3. 特定のパッチだけをダウンロードするか、パッチクラスタをダウンロードするかを決めます。次のいずれかの手順に従います。

- a. 「パッチ検索 (**Find Patch**)」検索フィールドにパッチ番号 (*patch-ID*) を入力し、「パッチ検索 (**Find Path**)」をクリックします。

patch-ID を入力すると、最新バージョンのパッチがダウンロードされます。

自由に使用できるパッチの場合、そのパッチの README が表示されます。自由に使用できないパッチの場合、ACCESS DENIED メッセージが表示されます。

SPARC システムと x86 システムではパッチ番号が異なります。これについては、表示されるパッチの README に記載されています。使用しているシステムアーキテクチャに適合したパッチをインストールしてください。

- b. パッチを適用するシステムで実行している **Solaris** リリースに基づいて、推奨されるパッチクラスタをクリックします。

4. 「この署名付きパッチ (*n bytes*) をダウンロードする **HTTPS FTP (Download Signed Patch HTTPS or FTP)**」ボタンをクリックします。

署名付きパッチ (またはパッチクラスタ) が正常にダウンロードされたら、Web ブラウザを閉じます。

5. 必要に応じて、ダウンロードされたパッチパッケージが含まれているディレクトリに移動します。

6. スーパーユーザーになるか、同等の役割を引き受けます。

7. 署名付きパッチを追加します。
たとえば、次のようになります。

```
# patchadd /tmp/114861-01.jar
```

▼ 署名付き Solaris パッチのダウンロードと追加を自動で行う方法 (patchadd)

次の手順は、署名付き Solaris パッチのダウンロードと追加を、自動的に一度で行いたい場合に使用できます。

この手順では、パッケージキースタアのセットアップが完了しているものと仮定しています。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 署名付きのパッチを **SunOnline** サイトからダウンロードし、システムに追加します。
たとえば、次のようになります。

```
# patchadd "http://sunsolve.central.sun.com/cgi/patchDownload.pl?target=114684&method=hs"
```

```
.  
. .
```

```

Downloading patch from ...
+ dwnld_file http://sunsolve.central.sun.com/cgi/patchDownload.pl?target=
114684&method=hs /tmp/patchadd-dwnld /var/sadm/security console patchadd
.....20%.....40%.....60%.....80%.....100%
## Downloading...
## Download Complete
.
.
.
Enter keystore password: xxx
.
.
.

```

smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)

smpatch コマンドを使って署名付きパッチをシステムに追加する場合は、次のマップに示す準備作業をすべて完了しておく必要があります。

作業	説明	参照先
1. Solaris パッケージの要件を確認する	パッチツールをサポートするために必要な Solaris パッケージがシステムにインストールされていることを確認する	357 ページの「署名付きパッチツールのパッケージ要件を確認する方法 (smpatch)」
2. Solaris パッチ管理ツールをダウンロードし、インストールする	使用している Solaris リリースに対応した Solaris パッチ管理ツールを選択する	357 ページの「Solaris パッチ管理ツールのダウンロードおよびインストール方法 (smpatch)」
3. キーストアに Sun の証明書をインポートする	パッチの署名の検証に使用する Sun の証明書をインポートし、受け入れる SUNWcert パッケージは、署名付きパッチツールのインストール時に自動的にインストールされる。すでに署名付きパッチツールがインストールされている場合は、SUNWcert パッケージを個別にインストールしてはならない	359 ページの「Java キーストアに Sun の証明書をインポートする方法」
4. (省略可能) キーストアのパスワードを変更する	キーストアの安全性を確保するため、パスワードを変更する	360 ページの「Java キーストアのパスワードの変更方法」
5. パッチ環境を設定する	署名付きパッチを追加できるようにシステムを設定する	360 ページの「パッチ環境の設定方法 (smpatch)」

Solaris パッチ管理ツールの使用 (smpatch)

Solaris パッチ管理ツールの使用時には、次の点に注意してください。

- 適切なパッチ (カーネルアップデートパッチ、Java パッチ、推奨パッチクラスタなど) により、システムが最新の状態になっていることを確認します。
- Solaris パッチ管理ツールのインストール後、パッチの署名の検証に使用する Sun の証明書を手動でインポートする必要があります。
- Solaris 2.6、7、または 8 のみ – システムにインストールされている以前のバージョンの PatchPro ソフトウェアは、Solaris Patch Manager Base バージョン 1.0 のインストール時にアップグレードされます。
- パッチを quiet システム (ログインしているユーザーが存在しないシステム) にインストールする場合は、シングルユーザーモードを選択することをお勧めします。
- 署名付きパッチは、smpatch download コマンドでダウンロードされるときに検証されます。

しかし、Solaris 9 システムでは、パッチのダウンロード時に署名検証メッセージが表示されません。これは、パッチが正常に検証された場合も同様です。署名の検証に失敗した場合、パッチはシステムにダウンロードされません。

- Solaris 9 のみ – smpatch コマンド行に認証情報を指定しなかった場合、認証情報の入力を求めるプロンプトが表示されます。

たとえば、smpatch コマンドに次のような構文で認証情報を指定します。

```
# smpatch add -p mypassword -u root -- -i patch-ID
```

smpatch のサブコマンド (add, analyze, download, または remove) を指定する場合は、サブコマンドの引数と認証オプションおよび引数を -- で区切りま
す。smpatch コマンドに認証情報を要求するプロンプトを表示させることもでき
ます。たとえば、次のようになります。

```
# /usr/sadm/bin/smpatch add -i patch-ID
```

Authenticating as user: root

ヘルプを参照するには /? を入力してください。Enter キーを押すと、
[] で囲まれたデフォルトが選択されます。

文字列の値を入力してください: password ::

ツール com.sun.admin.patchmgr.cli.PatchMgrCli を starbag から読み込み中
ユーザー root として starbag にログインしました。
starbag から com.sun.admin.patchmgr.cli.PatchMgrCli
がダウンロードされました。

- PatchPro 2.1 をアンインストールする必要がある場合は、
/opt/SUNWppro/bin/uninstallpatchpro スクリプトを使用します。現在の
ディレクトリが /opt/SUNWppro/bin である場合は、このスクリプトで
PatchPro2.1 を削除してはなりません。360 ページの「パッチ環境の設定方法
(smpatch)」の説明に従ってパスを設定し、適切なディレクトリ、たとえばルート
ディレクトリ (/) から uninstallpatchpro スクリプトを実行します。

署名付きパッチツールのパッケージ要件を確認する方法 (smpatch)

システムに署名付きパッチツールをインストールする前に、必要な Solaris パッケージがインストールされていることを確認します。Solaris 2.6、7、または 8 を実行している場合は、最小のシステム構成にいくつかのパッケージを追加する必要があります。Solaris 9 を実行している場合は、署名付きパッチツールを使用するためには開発者クラスタ (SUNWCprog) をインストールする必要があります。

1. 実行している **Solaris** リリースを確認し、次のいずれかの作業を行います。

- a. **Solaris 2.6** を実行している場合は、次のようにして、システムに必要なパッケージがインストールされているかどうかを確認します。

```
# pkginfo | grep SUNWmfrun
system      SUNWmfrun      Motif RunTime Kit
# pkginfo | grep SUNWlibC
system      SUNWlibC        Sun Workshop Compilers Bundled libC
# pkginfo | grep SUNWxcu4
system      SUNWxcu4        XCU4 Utilities
```

- b. **Solaris 7** または **8** を実行している場合は、次のようにして、システムに必要なパッケージがインストールされているかどうかを確認します。

```
# pkginfo | grep SUNWmfrm
system      SUNWmfrun      Motif RunTime Kit
# pkginfo | grep SUNWlibC
system      SUNWlibC        Sun Workshop Compilers Bundled libC
```

- c. **Solaris 9** を実行している場合は、次のようにして、システムに必要な開発者クラスタがインストールされていることを確認します。

```
# cat /var/sadm/system/admin/CLUSTER
CLUSTER=SUNWCprog
```

2. **pkginfo** コマンドを実行しても何も出力されない場合は、必要なパッケージをインストールしてください。

▼ Solaris パッチ管理ツールのダウンロードおよびインストール方法 (smpatch)

1. スーパーユーザーになります。

2. 次のリンクをたどって、使用している **Solaris** リリースに適した **tar** ファイルをダウンロードします。

<http://www.sun.com/PatchPro>

3. 次のいずれかの方法で、パッチツールパッケージを展開します。

- a. **Solaris 2.6** または **7** を実行している場合は、次のコマンドを使ってパッケージの圧縮を解除し、展開します。

```
# uncompress SUNWpkg-name.tar.Z
# tar xvf SUNWpkg-name.tar
```

- b. **Solaris 8** または **9** を実行している場合は、次のコマンドを使ってパッケージを展開します。

```
# gunzip -dc SUNWpkg-name.tar.gz | tar xvf -
```

4. インストールスクリプトを実行します。

```
# cd unzipped-pkg-dir
# ./setup
```

インストールスクリプトの実行時のエラーについては、366 ページの「署名付きパッチに関する問題の障害追跡 (smpatch)」を参照してください。

例 — Solaris パッチ管理ツールのダウンロードおよびインストール (smpatch)

次の例では、Solaris 2.6 パッチ管理ツールをダウンロードし、インストールします。

```
# uncompress pproSunOSsparc5.6jre2.1.tar.Z
# tar xvf pproSunOSsparc5.6jre2.1.tar
.
.
.
# cd pproSunOSsparc5.6jre2.1
# ./setup
.
.
.
```

次の例では、Solaris 9 パッチ管理ツールをダウンロードし、インストールします。

```
# gunzip -dc pproSunOSsparc5.9jre2.1.tar.gz | tar xvf -
.
.
# cd pproSunOSsparc5.9jre2.1
# ./setup
.
.
.
```

▼ Java キーストアに Sun の証明書をインポートする方法

keytool コマンドを使って、システムに追加する署名付きパッチの検証に使用する Sun の証明書をインポートし、検証します。証明書を以前のインストールからインポートした場合も、この作業を実行する必要があります。

注 - SUNWcert パッケージは、署名付きパッチツールのインストール時に自動的にインストールされます。すでに署名付きパッチツールがインストールされている場合は、SUNWcert パッケージを個別にインストールしないでください。

1. **Solaris** パッチ管理ツールをダウンロードするために必要な準備作業が完了していることを確認します。
2. スーパーユーザーになります。
3. **Sun** のルート証明書と **Sun** の **Class B** 証明書のフィンガープリントを確認します。

```
# /usr/j2se/bin/keytool -printcert -file /etc/certs/SUNW/
smirootcacert.b64
# /usr/j2se/bin/keytool -printcert -file /etc/certs/SUNW/smicacert.b64
```

4. 次のサイトで、コマンドの出力結果と **Sun** のルート証明書および **Class B** 証明書のフィンガープリントが一致していることを確認します。

<https://www.sun.com/pki/ca/>

5. **Sun** の **Class B** 証明書をシステムにインポートし、受け入れます。

```
# /usr/j2se/bin/keytool -import -alias smiacacert -file /etc/certs/SUNW/
smiacacert.b64 -keystore /usr/j2se/jre/lib/security/cacerts
Enter keystore password: changeit
Owner: O=Sun Microsystems Inc, CN=Sun Microsystems Inc CA (Class B)
Issuer: CN=Sun Microsystems Inc Root CA, O=Sun Microsystems Inc, C=US
Serial number: 1000006
Valid from: Mon Nov 13 12:23:10 MST 2000 until: Fri Nov 13 12:23:10 ...
Certificate fingerprints:
    MD5:  B4:1F:E1:0D:80:7D:B1:AB:15:5C:78:CB:C8:8F:CE:37
    SHA1: 1E:38:11:02:F0:5D:A3:27:5C:F9:6E:B1:1F:C4:79:95:E9:6E:D6:DF
Trust this certificate? [no]: yes
Certificate was added to keystore
```

6. **Sun** のルート証明書をシステムにインポートし、受け入れます。

```
# /usr/j2se/bin/keytool -import -alias smirootcacert -file /etc/certs/
SUNW/
smirootcacert.b64 -keystore /usr/j2se/jre/lib/security/cacerts
Enter keystore password: changeit
Owner: CN=Sun Microsystems Inc Root CA, O=Sun Microsystems Inc, C=US
```

```
Issuer: CN=GTE CyberTrust Root, O=GTE Corporation, C=US
Serial number: 200014a
Valid from: Tue Nov 07 15:39:00 MST 2000 until: Thu Nov 07 16:59:00 ...
Certificate fingerprints:
    MD5: D8:B6:68:D4:6B:04:B9:5A:EB:34:23:54:B8:F3:97:8C
    SHA1: BD:D9:0B:DA:AE:91:5F:33:C4:3D:10:E3:77:F0:45:09:4A:E8:A2:98
Trust this certificate? [no]: yes
Certificate was added to keystore
```

- 署名付きパッチ証明書をシステムにインポートし、受け入れます。

```
# /usr/j2se/bin/keytool -import -alias patchsigning -file /opt/SUNWppro/
etc/certs/patchsigningcert.b64 -keystore /usr/j2se/jre/lib/security/
cacerts
Enter keystore password: changeit
Owner: CN=Enterprise Services Patch Management, O=Sun Microsystems Inc
Issuer: O=Sun Microsystems Inc, CN=Sun Microsystems Inc CA (Class B)
Serial number: 1400007b
Valid from: Mon Sep 24 14:38:53 MDT 2001 until: Sun Sep 24 14:38:53 ...
Certificate fingerprints:
    MD5: 6F:63:51:C4:3D:92:C5:B9:A7:90:2F:FB:C0:68:66:16
    SHA1: D0:8D:7B:2D:06:AF:1F:37:5C:0D:1B:A0:B3:CB:A0:2E:90:D6:45:0C
Trust this certificate? [no]: yes
Certificate was added to keystore
```

▼ Java キーストアのパスワードの変更方法

- スーパーユーザーになります。
- キーストアのパスワードを変更します。

```
# /usr/j2se/bin/keytool -keystore /usr/j2se/jre/lib/security/
cacerts
Enter keystore password: changeit
New keystore password: new-password
Re-enter new keystore password: new-password
```

▼ パッチ環境の設定方法 (smpatch)

- スーパーユーザーになります。
- パッチツールのディレクトリをパスに追加します。
- (省略可能) システムのハードウェア構成を確認します。 **smpatch analyze** コマンドは、このハードウェア構成に基づいて必要なパッチを特定します。

```
# pprosetup -H

Change Hardware Configuration.
```



```
Analyzing this computer.
```

```
.....
```

このコマンドで特定されるのは、Sun のネットワークストレージ製品だけです。

4. システムに追加するパッチの種類を特定します。

```
# pprosetup -i standard:singleuser:rebootafter:reconfigafter
```

システムのデフォルトパッチポリシーが確立されます。

5. (省略可能) システムに規約署名付きパッチを追加したい場合は、次の手順に従って SunSolve のユーザー名とパスワードを定義します。

- a. SunSolve ユーザー名を定義します。

```
# pprosetup -u username
```

- b. SunSolve パスワードを定義します。次のファイルにパスワードを追加してください。

```
/opt/SUNWppro/lib/.sunsolvepw
```

6. パッチツールがシステムにパッチをダウンロードできるように、プロキシサーバーを特定します。

- a. システムがファイアウォールで保護されている場合は、**patchpro.sun.com** サーバーと、次のいずれかの Sun パッチサーバー (パッチのダウンロード用サーバー) にアクセスできるプロキシサーバーを定義する必要があります。

- `americas.patchmanager.sun.com` (デフォルト)
- `emea.patchmanager.sun.com`
- `japan.patchmanager.sun.com`

- b. 次のコマンドで、プロキシサーバーを特定します。

```
# pprosetup -x proxy-server:proxy-port
```

たとえば、`webaccess.corp.net.com` をプロキシサーバーに指定する場合、次のように `pprosetup` コマンドを入力します。

```
# pprosetup -x webaccess.corp.net.com:8080
```

次に進む手順

署名付きパッチの準備作業がすべて完了したら、パッチ管理ツールを使って署名付きパッチを追加します。

smpatch コマンドによる署名付きパッチの管理 (作業マップ)

作業	説明	参照先
1. 署名付きパッチの準備作業を行う	署名付きパッチの準備作業をすべて完了する ■ 署名付きパッチツールのパッケージ要件を確認する ■ Solaris パッチ管理ツールをダウンロードし、インストールする ■ キーストアに Sun の証明書をインポートする ■ キーストアのパスワードを変更する ■ 署名付きパッチを追加できるようにシステムを設定する	355 ページの「smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)」
2. 署名付きパッチ (1 つまたは複数) をダウンロードし、追加する	smpatch コマンドを使って、署名付きパッチをダウンロードし、追加する	362 ページの「署名付きパッチをダウンロードし、追加する方法 (smpatch)」
3. (省略可能) 署名付きパッチを削除する	必要に応じて、署名のないパッチをシステムから削除する	365 ページの「署名付きパッチを削除する方法 (smpatch)」

▼ 署名付きパッチをダウンロードし、追加する方法 (smpatch)

1. システムに署名付きパッチをダウンロードし、追加する前に、準備作業が完了していることを確認します。詳細については、355 ページの「smpatch コマンドで署名付きパッチを管理するための準備 (作業マップ)」を参照してください。

2. スーパーユーザーになります。

3. Solaris 9 システムのみ – Solaris 管理コンソールサーバーに、システムに PatchPro パッケージが追加されていることを通知します。

```
# /etc/init.d/init.wbem stop  
# /etc/init.d/init.wbem start
```

4. SunSolve の Web サイトから署名付きパッチ (1 つまたは複数) をダウンロードします。

```
# smpatch download -i patch-ID
```

```
Requested patches:
```

patch-ID

Downloading the requested patches

/var/sadm/spool/patch-ID.jar has been validated.

For downloaded patch(es) see */var/sadm/spool*

5. 署名付きパッチを追加します。

```
# smpatch add -i patch-ID
```

例 — 署名付きパッチをダウンロードし、追加する (smpatch)

次の例では、smpatch コマンドを使って、Solaris 9 システムに署名付きパッチをダウンロードし、追加します。

```
# /usr/sadm/bin/smpatch download -i 111711-01  
Authenticating as user: root
```

ヘルプを参照するには */?* を入力してください。Enter キーを押すと、
[] で囲まれたデフォルトが選択されます。

文字列の値を入力してください: password ::

ツール *com.sun.admin.patchmgr.cli.PatchMgrCli* を *starbag* から読み込み中
ユーザー *root* として *starbag* にログインしました。

starbag から *com.sun.admin.patchmgr.cli.PatchMgrCli*
がダウンロードされました。

```
Requested patches:  
111711-01
```

```
Downloading the requested patches ...
```

For downloaded patch(es) see */var/sadm/spool*.

```
# smpatch add -i 111711-01  
Authenticating as user: root
```

ヘルプを参照するには */?* を入力してください。Enter キーを押すと、
[] で囲まれたデフォルトが選択されます。

文字列の値を入力してください: password ::

ツール *com.sun.admin.patchmgr.cli.PatchMgrCli* を *starbag* から読み込み中
ユーザー *root* として *starbag* にログインしました。

starbag から *com.sun.admin.patchmgr.cli.PatchMgrCli*
がダウンロードされました。

```
On machine starbug ...  
Installing patch 111711-01
```

```
#
```

次の例では、`smpatch` コマンドを使って、Solaris 2.6 システムにパッチ 105081-45 をダウンロードし、追加します。

```
# smpatch download -i 105407-01

Requested patches:

    105407-01

Downloading the requested patches

/var/sadm/spool/105407-01.jar has been validated.

For downloaded patch(es) see /var/sadm/spool
# smpatch add -i 105407-01

On machine "earth/172.20.27.27" ...
```

```
Installing patch 105407-01 ...
Purging /var/sadm/spool/105407-01
/var/sadm/spool/README.txt has been moved to
/var/sadm/spool/patchproSequester
```

次の例では、`smpatch` コマンドを使って、Solaris 7 システムにパッチ 107081-45 をダウンロードし、追加します。このパッチには、依存パッチが2つあります。依存パッチは自動的にダウンロードされ、検証されます。

```
# smpatch download -i 107081-45

Requested patches:

    107081-45

Downloading the requested patches

The following patches were added due to patch dependencies:
    108376-37
    107656-09

/var/sadm/spool/108376-37.jar has been validated.

/var/sadm/spool/107656-09.jar has been validated.

/var/sadm/spool/107081-45.jar has been validated.

For downloaded patch(es) see /var/sadm/spool
# smpatch add -i 108376-37 -i 107656-09 -i 107081-45

On machine "venus/172.20.27.26" ...

Installing patch 108376-37 ...
Installing patch 107656-09 ...
```

```
Installing patch 107081-45 ...
Purging /var/sadm/spool/108376-37
Purging /var/sadm/spool/107656-09
Purging /var/sadm/spool/107081-45
```

次の例では、ftp コマンドを使って、SunSolve Online の Web サイトから署名付きの Solaris 8 パッチをダウンロードします。その後、smpatch add コマンドを使って、このパッチをシステムに追加します。

```
# ftp sunsolve.sun.com
Connected to sunsolve.sun.com.
220-
220-Welcome to the SunSolve Online FTP server.
220-
220-Public users may log in as anonymous.
.
.
.
Name (sunsolve.sun.com:root): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: xxx
230-
230-SUN MICROSYSTEMS, INC.
.
.
.230 Guest login ok, access restrictions apply.
ftp> cd signed_patches
250 CWD command successful.
ftp> get 112846-01.jar /var/sadm/spool/112846-01.jar
200 PORT command successful.
150 Opening ASCII mode data connection for 112846-01.jar (22524 bytes).
226 Transfer complete.
local: /var/sadm/spool/112846-01 remote: 112846-01.jar
22613 bytes received in 0.065 seconds (341.70 Kbytes/s)
ftp> quit
# smpatch add -i 112846-01
On machine "earth/172.20.27.27" ...
```

```
Installing patch 112846-01 ...
Purging /var/sadm/spool/112846-01
```

▼ 署名付きパッチを削除する方法 (smpatch)

1. スーパーユーザーになります。
2. 署名付きパッチを削除します。

```
# smpatch remove -i patch-ID
```

複数のパッチを1回のコマンドでまとめて削除することはできません。

例 — 署名付きパッチを削除する (smpatch)

次の例では、Solaris 9 を実行しているシステムから署名付きパッチを削除します。

```
# /usr/sadm/bin/smpatch remove -- -i 111711-01
Authenticating as user: root
```

ヘルプを参照するには /? を入力してください。Enter キーを押すと、
[] で囲まれたデフォルトが選択されます。
文字列の値を入力してください: password :
ツール com.sun.admin.patchmgr.cli.PatchMgrCli を starbag から読み込み中
ユーザー root として starbag にログインしました。
starbag から com.sun.admin.patchmgr.cli.PatchMgrCli
がダウンロードされました。

```
On machine starbug ...
    Removing patch 111711-01
```

次の例では、Solaris 2.6 を実行しているシステムから署名付きパッチを削除します。

```
# smpatch remove -i 105407-01
```

```
On machine "earth/172.20.27.27" ...
```

```
Removing patch 105407-01
```

```
Checking installed patches...
```

```
Backing out patch 105407-01...
```

```
Patch 105407-01 has been backed out.
```

署名付きパッチに関する問題の障害追跡 (smpatch)

署名付きパッチに関する問題の障害追跡やエラーメッセージの最新情報については、
<http://jp.sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq>
を参照してください。

パッチツールのログファイルの表示

パッチ管理ツールのインストールや署名付きパッチの追加に関する問題は、システム上のさまざまなログファイルから特定できます。

デフォルトでは、PatchPro はシステムログファイルにメッセージを書き込みます。システムログファイルの場所は、syslog 構成ファイル `/etc/syslog.conf` で確認できます。PatchPro 構成ファイル `/opt/SUNWppro/etc/patchpro.conf` の `patchpro.log.file` プロパティを更新することにより、PatchPro がローカルファイルシステム上の別のファイルにメッセージを書き込むように設定できます。

たとえば、`/var/tmp/patchpro.log` ファイルにメッセージを書き込むためには、`patchpro.log.file` プロパティの値を `/var/tmp/patchpro.log` に設定します。

次の表を使って、パッチ管理ツールや署名付きパッチのインストール時の障害情報を格納するログファイルを特定します。

ログファイル	説明
<code>/var/tmp/ppro_install_log.nmm</code>	PatchPro パッケージやパッチのインストールが成功したかどうかを示す
<code>/var/tmp/log/patchpro.log</code>	さまざまなパッチツールで署名付きパッチを追加したときに発生した問題を示す
<code>/var/adm/messages</code>	さまざまなパッチツールで署名付きパッチを追加したとき、またはパッチツールが正しく初期化されなかったときに発生した問題を示す
Solaris 9 システム上の Solaris 管理コンソールのログビューア	Solaris 管理コンソールのパッチツールを使って署名付きパッチを正しく追加できたかどうかを示す

▼ 隔離されたパッチの解決方法

必須パッチを要求するパッチや、システムのリブートを必要とするパッチは、正しくインストールされないことがあります。PatchPro でインストールできないパッチは、`/var/spool/pkg/patchproSequester` ディレクトリに隔離されます。

パッチの README ファイルの REQUIRED PATCHES セクションに必須パッチが記載されているかどうかを確認してください。

パッチの README ファイルのコピーは、SunSolve Online の Web サイトから表示できます。JAR アーカイブから README ファイルを抽出することもできます。デジタル署名が改ざんされる恐れがあるため、JAR アーカイブは展開しないでください。パッチの README ファイルは、次の手順で安全に抽出できます。

パッチが正しくインストールされなかった原因を特定するため、`/var/tmp/log/patchpro.log` ファイルの内容も確認してください。

1. `/var/spool/pkg/patchproSequester` ディレクトリの内容を表示して、パッチ (1 つまたは複数) がインストールされなかったことを確認します。

```
# cd /var/spool/pkg/patchproSequester; ls
```

2. JAR アーカイブから **README** ファイルを抽出します。

a. まず、**README** ファイルの名前を特定します。次に例を示します。

```
# /usr/j2se/bin/jar tvf 107058-01.jar | grep README
```

b. 次に、**README** ファイルを抽出します。次に例を示します。

```
# /usr/j2se/bin/jar xvf 107058-01.jar 107058-01/README.107058-01
extracted: 107058-01/README.107058-01
```

3. **README** ファイルを表示します。

次に例を示します。

```
# more 107058-01/README.107058-01
```

▼ Java キーストアからインポート済みの証明書を削除する方法

PatchPro のインストール時に問題が発生した場合は、証明書をいったん削除してから、もう一度インポートすることをお勧めします。

1. スーパーユーザーになります。
2. 以前にインポートした証明書を削除します。

```
#/usr/j2se/bin/keytool -delete -alias smicacert -keystore
/usr/j2se/jre/lib/security/cacerts
キーストアのパスワードを入力してください : changeit
# /usr/j2se/bin/keytool -delete -alias smirootcacert -keystore
/usr/j2se/jre/lib/security/cacerts
キーストアのパスワードを入力してください : changeit
# /usr/j2se/bin/keytool -delete -alias patchsigning -keystore
/usr/j2se/jre/lib/security/cacerts
キーストアのパスワードを入力してください : changeit
```


署名のない Solaris パッチの管理 (作業マップ)

作業	説明	参照先
1. (省略可能) 署名のないパッチの情報を表示する	システムにインストール済みの署名のないパッチに関する情報を表示する	369 ページの「Solaris パッチの情報を表示する方法」
2. 署名のないパッチをダウンロードする	署名のないパッチをシステムにダウンロードする	370 ページの「署名のない Solaris パッチのダウンロード方法」
3. 署名のないパッチを追加する	署名のないパッチをシステムに追加する	371 ページの「署名のない Solaris パッチの追加方法」
4. (省略可能) 署名のないパッチを削除する	必要に応じて、署名のないパッチをシステムから削除する	362 ページの「 <code>smpatch</code> コマンドによる署名付きパッチの管理 (作業マップ)」

署名のない Solaris パッチの情報を表示する

パッチをインストールするにあたっては、以前にインストールされたパッチの詳細を知る必要がある場合もあります。次の表に、すでにシステムにインストール済みのパッチについての情報を表示するためのコマンドを示します。

表 25-1 Solaris パッチの管理用コマンド

コマンド	説明
<code>patchadd -p, showrev -p</code>	システムに適用されたすべてのパッチを表示する
<code>pkgparam pkgid PATCHLIST</code>	<code>pkgid</code> (パッケージの名前) によって識別されるパッケージに適用されたすべてのパッチを表示する。たとえば、 <code>SUNWadmap</code>
<code>patchadd -S Solaris-OS -p</code>	OS サーバーにインストールされているすべての <code>/usr</code> パッチを表示する

Solaris パッチの情報を表示する方法

`patchadd -p` コマンドを使用して、システムにインストールされているパッチに関する情報を表示します。

```
$ patchadd -p
```

次のコマンドを使用して、特定のパッチがシステムにインストールされているかどうかを確認します。たとえば、次のようになります。

```
$ patchadd -p | grep 111879
```

署名のない Solaris パッチの追加

署名のないパッチをサーバーやスタンドアロンシステムに追加するには、`patchadd` コマンドを使用します。ディスクレスクライアントシステムにパッチを追加する必要がある場合は、145 ページの「ディスクレスクライアント OS サービスにパッチを適用する」を参照してください。

パッチを追加すると、`patchadd` コマンドは `pkgadd` コマンドを呼び出して、パッチディレクトリからローカルシステムのディスクへパッチパッケージをインストールします。厳密に言えば、`patchadd` コマンドは次の処理を行います。

- 管理ホストとターゲットホストの Solaris バージョン番号を確認する。
- インストール中のパッチによって置換されるパッチ、このパッチに必要な他のパッチ、およびこのパッチと互換性を持たないパッチに関する情報を使用して、パッチパッケージの `pkginfo` ファイルを更新する。

パッチのインストール時に、`patchadd` コマンドはインストールプロセスの記録を現在の Solaris バージョンの `/var/sadm/patch/patch-ID/log` ファイルに保存します。

この `patchadd` コマンドは、次の条件ではパッチをインストールしません。

- パッケージがホストに完全にインストールされていない。
- パッチパッケージのアーキテクチャが、システムのアーキテクチャと異なる。
- パッチパッケージのバージョンが、インストールされているパッケージのバージョンと一致しない。
- 同じベースコードと上位のバージョン番号を持つパッチがすでにインストールされている。
- パッチが、すでにインストールされている他のパッチと互換性を持たない。インストール済みの各パッチは、この情報を `pkginfo` ファイルに格納している。
- インストールしようとしているパッチには他のパッチが必要だが、そのパッチがインストールされていない。

▼ 署名のない Solaris パッチのダウンロード方法

1. (省略可能) パッチを適用するシステムにログインします。
あるいは、パッチをダウンロードし、`ftp` コマンドを使用してそのパッチをターゲットのシステムにコピーすることもできます。
2. **Web** ブラウザを開き、**SunSolve Online** のサイトに移動します。

`http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`

3. 特定のパッチだけをダウンロードするか、パッチクラスタをダウンロードするかを決めます。次のいずれかの手順に従います。
 - a. 「パッチ検索 (**Find Patch**)」検索フィールドにパッチ番号 (*patch-ID*) を入力し、「パッチ検索 (**Find Path**)」をクリックします。

patch-ID を入力すると、最新バージョンのパッチがダウンロードされます。自由に使用できるパッチの場合、そのパッチの README が表示されます。自由に使用できないパッチの場合、ACCESS DENIED メッセージが表示されます。SPARC システムと x86 システムではパッチ番号が異なります。これについては、表示されるパッチの README に記載されています。使用しているシステムアーキテクチャに適合したパッチをインストールしてください。
 - b. パッチを適用するシステムで実行している Solaris に基づいて、推奨されるパッチクラスタをクリックします。
4. 「このパッチ (*n bytes*) をダウンロードする **HTTP FTP (Download Patch HTTP or FTP)**」ボタンをクリックします。

パッチ (またはパッチクラスタ) が正常にダウンロードされたら、Web ブラウザを閉じます。
5. 必要に応じて、ダウンロードされたパッチパッケージが含まれているディレクトリに移動します。
6. パッチパッケージを解凍します。

```
% unzip patch-ID-revision
```

▼ 署名のない Solaris パッチの追加方法

1. スーパーユーザーになります。
2. パッチ (または複数のパッチ) を追加します。

```
# patchadd patch-ID-revision
```
3. パッチが正常に削除されたことを確認します。

```
# patchadd -p | grep patch-ID-revision
```

例 — 署名のない Solaris パッチの追加

次の例では、システムに Solaris 8 パッチ (パッチ番号 111879-01) を追加します。パッチのダウンロードは完了しているものとします。

```
# patchadd /export/Sol8patch/111879-01  
Checking installed patches...
```

```
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 111879-01 has been successfully installed.
See /var/sadm/patch/111879-01/log for details

Patch packages installed:
  SUNWwsr
# patchadd -p | grep 111879-01
Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
```

署名のない Solaris パッチの削除

パッチをバックアウトすると、`patchrm` コマンドは、そのパッチによって修正されたすべてのファイルを復元します。ただし、次の場合は除きます。

- パッチが `patchadd -d` オプションを使用してインストールされた場合 (`-d` オプションは、更新中または置換中のファイルのコピーを保管しないように `patchadd` に指示する)
- パッチが最新のパッチによって置換されている場合
- パッチが他のパッチによって必要とされている場合

`patchrm` コマンドは、`pkgadd` を呼び出して、最初のパッチインストールで保管されたパッケージを復元します。

パッチを削除している間、`patchrm` は、`/tmp/backoutlog.process_id` にバックアウトプロセスのログを格納します。パッチが正常にバックアウトされた場合、このログファイルは削除されます。

▼ 署名のない Solaris パッチを削除する方法

署名のない Solaris パッチを削除するには、`patchrm` コマンドを使用します。

1. スーパーユーザーになります。
2. パッチを削除します。

```
# patchrm patch-ID-revision
```
3. パッチが削除されたことを確認します。

```
# patchadd -p | grep patch-ID-revision
```

例 — 署名のない Solaris パッチの削除

次の例では、Solaris 8 パッチ (パッチ番号 111879-01) を削除します。

```
# patchrm 111879-01
Checking installed patches...
Backing out patch 111879-01...
Patch 111874-02 has been backed out.
# showrev -p | grep 111879-01
#
```


第 26 章

デバイスの管理

以下の各章では、デバイスの管理方法について説明します。

第 27 章	デバイス構成の概要およびシステムのデバイス情報を表示する手順について説明します。
第 28 章	デバイスの設定手順について説明します。
第 29 章	USB デバイスの概要および USB デバイスの使用手順について説明します。
第 30 章	デバイス命名規則の概要とデバイスにアクセスするための手順について説明します。

第 27 章

デバイスの管理 (手順)

この章では、Solaris 環境のディスク、CD-ROM、テープデバイスなどの周辺デバイスを管理する方法について説明します。

この章で説明する手順は次のとおりです。

- 382 ページの「システム構成情報を表示する方法」
- 383 ページの「デバイス情報を表示する方法」
- 386 ページの「デバイスドライバを追加する方法」
- 385 ページの「周辺デバイスを追加する方法」

この章の内容は次のとおりです。

- 377 ページの「デバイス管理作業についての参照先」
- 378 ページの「デバイスドライバについて」
- 378 ページの「デバイスの自動構成」
- 380 ページの「デバイス構成情報の表示」

周辺デバイスへのアクセスについては、第 30 章を参照してください。

Solaris 環境のデバイス管理には、通常、システムでの周辺デバイスの追加と削除、デバイスをサポートするための Sun 以外のデバイスドライバの追加、システム構成情報の表示が含まれます。

デバイス管理作業についての参照先

次の表に、デバイスをホットプラグしたり、シリアルデバイス (プリンタやモデムなど) や周辺デバイス (ディスク、CD-ROM、テープデバイスなど) を追加したりする手順を説明している参照先を示します。

表 27-1 デバイスを追加する場合の参照先

デバイス管理作業	参照先
ホットプラグ不可ディスクの追加	第 34 章または第 35 章
SCSI または PCI デバイスのホットプラグ	392 ページの「cfgadm コマンドによる SCSI ホットプラグ」または 402 ページの「x86: cfgadm コマンドによる PCI ホットプラグ」
USB デバイスのホットプラグ	424 ページの「USB デバイスのホットプラグ」
CD-ROM またはテープデバイスの追加	385 ページの「周辺デバイスを追加する方法」
モデムの追加	『Solaris のシステム管理 (上級編)』の「端末とモデムの管理 (概要)」
プリンタの追加	『Solaris のシステム管理 (上級編)』の「印刷サービスの管理 (概要)」

デバイスドライバについて

コンピュータは通常、広範囲の周辺デバイスと大量記憶デバイスを使用します。たとえば、各システムには、SCSI ディスクドライブ、キーボードとマウス、磁気バックアップメディアなどがあります。これ以外に一般に使用されるデバイスには、CD-ROM ドライブ、プリンタとプロッタ、ライトペン、タッチセンサー式画面、デジタルタイザ、タブレットとスタイラスのペアがあります。

Solaris ソフトウェアは、これらのデバイスと直接には通信を行いません。各タイプのデバイスに異なるデータ形式、プロトコル、および転送速度が必要になります。

「デバイスドライバ」は、オペレーティングシステムが特定のハードウェアと通信できるようにする低レベルのプログラムです。このドライバは、そのハードウェアに対するオペレーティングシステムの「インタプリタ」として機能します。

デバイスの自動構成

プラットフォーム固有の構成要素を備えた汎用コアと、一連のモジュールからなるカーネルは、Solaris 環境で自動的に構成されます。

カーネルモジュールとは、システムで固有の作業を実行するために使用されるハードウェアまたはソフトウェアの構成要素のことです。「ロード可能」なカーネルモジュールの例としては、デバイスのアクセス時にロードされるデバイスドライバがあげられます。

プラットフォームに依存しないカーネルは `/kernel/genunix` です。プラットフォーム固有の構成要素は、`/platform/`uname -m`/kernel/unix` です。

カーネルモジュールについては、次の表で説明します。

表 27-2 カーネルモジュール

保存場所	内容
<code>/platform/`uname -m`/kernel</code>	プラットフォーム固有のカーネル構成要素
<code>/kernel</code>	システムのブートに必要なすべてのプラットフォームに共通のカーネル構成要素
<code>/usr/kernel</code>	特定の命令セット内にあるすべてのプラットフォームに共通のカーネル構成要素

システムは、ブート時にどのデバイスが接続されるかを判断します。さらに、カーネルは、それ自体を動的に構成して、必要なモジュールだけをメモリーにロードします。ディスクデバイスやテープデバイスなどのデバイスが初めてアクセスされると、対応するデバイスドライバがロードされます。このプロセスは、「自動構成」と呼ばれます。これは、すべてのカーネルモジュールが、必要に応じて自動的にロードされるためです。

`/etc/system` ファイルを修正することによって、カーネルモジュールがロードされる方法をカスタマイズできます。このファイルを修正する方法については、`system` (4) のマニュアルページを参照してください。

自動構成の機能と利点

自動構成の利点は次のとおりです。

- モジュールが必要に応じてロードされるため、主メモリーをより効率的に使用できる。
- 新しいデバイスがシステムに追加されるときに、カーネルを再構成する必要がない。
- カーネルを再構成しないでドライバをロード、テストして、システムをリブートすることができる。

自動構成プロセスは、システム管理者が新しいデバイス (およびドライバ) をシステムに追加するとき使用されます。これは、再構成ブートを実行することによって行われるため、システムは新しいデバイスを認識することができます。

標準サポートされていないデバイスを使用する場合

Solaris 環境には、各種の標準デバイスをサポートするために必要なデバイスドライバが組み込まれています。これらのドライバは、`/kernel/drv` および `/platform/`uname -m`/kernel/drv` ディレクトリにあります。

ただし Solaris で標準にサポートされていないデバイスを購入した場合は、そのメーカーから、デバイスを正しくインストール、保守、管理するために必要なソフトウェアを提供してもらう必要があります。

そのようなデバイス用ソフトウェアには、少なくともデバイスドライバとその関連設定 (`.conf`) ファイルが含まれます。`.conf` ファイルは、`drv` ディレクトリにもあります。また、サポートされていないデバイスは、Solaris で提供されるユーティリティと互換性を持たないので、保守および管理用のユーティリティが必要になる場合があります。

詳細は、デバイスのご購入先にお問い合わせください。

デバイス構成情報の表示

システムとデバイスの構成情報を表示するには、次の 3 つのコマンドを使用します。

コマンド	マニュアルページ	説明
<code>prtconf</code>	<code>prtconf (1M)</code>	メモリーの総量、システムのデバイス階層によって記述されたデバイス構成を含む、システム構成情報を表示します。このコマンドによる出力は、システムのタイプによって異なります。
<code>sysdef</code>	<code>sysdef (1M)</code>	システムハードウェア、疑似デバイス、ロード可能なモジュール、および指定のカーネルパラメータを含む、デバイス構成情報を表示します。
<code>dmesg</code>	<code>dmesg (1M)</code>	最後のリブート以降にシステムに接続されたデバイスのリストと、システム診断情報を表示します。

システムのデバイスの識別に使用されるデバイス名については、446 ページの「デバイス名の命名規則」を参照してください。

driver not attached メッセージ

次のドライバ関連メッセージが、`prtconf` コマンドと `sysdef` コマンドによって表示されることがあります。

```
device, instance #number (driver not attached)
```

このメッセージは、このデバイスのドライバが使用できないことをいつも示すわけではありません。このメッセージは、ノードにデバイスがないか、あるいはデバイスが使用中ではないために、デバイスインスタンスに「現在」接続されているドライバがないことを示します。ドライバは、デバイスがアクセスされると自動的にロードされ、デバイスが使用されなくなると自動的にアンロードされます。

システムデバイスの識別

`prtconf` および `sysdef` コマンドの出力から、システムに接続されているディスク、テープ、および CD-ROM デバイスを識別できます。デバイスインスタンスの出力の横に `driver not attached` メッセージが表示されます。これらのデバイスは、何らかのシステムプロセスによって常に監視されているため、「`driver not attached`」メッセージは通常、そのデバイスインスタンスにデバイスがないことを示す良い標識になります。

たとえば、次の `prtconf` 出力は、`instance #3` と `instance #6` のデバイスを識別しています。これは、最初の SCSI ホストアダプタ (`esp, instance #0`) のターゲット 3 のディスクデバイスと、ターゲット 6 の CD-ROM デバイスを示しています。

```
$ /usr/sbin/prtconf
.
.
.
esp, instance #0
    sd (driver not attached)
    st (driver not attached)
    sd, instance #0 (driver not attached)
    sd, instance #1 (driver not attached)
    sd, instance #2 (driver not attached)
    sd, instance #3
    sd, instance #4 (driver not attached)
    sd, instance #5 (driver not attached)
    sd, instance #6
.
.
.
```

次のコマンドを使うと、システムに接続されたデバイスのみを表示できます。

```
$ prtconf | grep -v not
```

デバイス情報は、`sysdef` 出力からも得られます。

システム構成情報を表示する方法

システム構成情報を表示するには、`prtconf` コマンドを使用してください。

```
# /usr/sbin/prtconf
```

疑似デバイス、ロード可能なモジュール、および指定のカーネルパラメータを含むシステム構成情報を表示するには、`sysdef` コマンドを使用してください。

```
# /usr/sbin/sysdef
```

例 — システム構成情報を表示する

SPARC システムでは、次の `prtconf` 出力が表示されます。

```
# prtconf
System Configuration: Sun Microsystems sun4u
Memory size: 128 Megabytes
System Peripherals (Software Nodes):
SUNW,Ultra-5_10
  packages (driver not attached)
    terminal-emulator (driver not attached)
    deblocker (driver not attached)
    obp-tftp (driver not attached)
    disk-label (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    sun-keyboard (driver not attached)
    ufs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  pci, instance #0
    pci, instance #0
      ebus, instance #0
        auxio (driver not attached)
        power, instance #0
        SUNW,pll (driver not attached)
        se, instance #0
        su, instance #0
        su, instance #1
        ecpp (driver not attached)
        fdthree, instance #0
  .
  .
  .
```

x86 システムからは、次の `sysdef` 出力が表示されます。

```
# sysdef
* Hostid
```

```

*
 29f10b4d
*
* i86pc Configuration
*
*
* Devices
*
+boot (driver not attached)
  memory (driver not attached)
aliases (driver not attached)
chosen (driver not attached)
i86pc-memory (driver not attached)
i86pc-mmio (driver not attached)
openprom (driver not attached)
options, instance #0
packages (driver not attached)
delayed-writes (driver not attached)
itu-props (driver not attached)
isa, instance #0
  motherboard (driver not attached)
  pnpADP,1542, instance #0
  asy, instance #0
  asy, instance #1
  lp, instance #0 (driver not attached)
  fdc, instance #0
    fd, instance #0
    fd, instance #1 (driver not attached)
  kd (driver not attached)
  kdmouse (driver not attached)
.
.
.

```

デバイス情報を表示する方法

デバイス情報は、`dmesg` コマンドを使用して表示してください。

```
# /usr/sbin/dmesg
```

この `dmesg` 出力は、システムコンソール上のメッセージとして表示され、最後のリブート以降に接続されたデバイスを表示します。

例 — デバイス情報を表示する

SPARC システムからは、次の `dmesg` 出力が表示されます。

```
# dmesg
Jan  3 08:44:41 starbug genunix: [ID 540533 kern.notice] SunOS Release 5.9 ...
Jan  3 08:44:41 starbug genunix: [ID 913631 kern.notice] Copyright 1983-2002 ...
```

```
Jan 3 08:44:41 starbug genunix: [ID 678236 kern.info] Ethernet address = ...
Jan 3 08:44:41 starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
Jan 3 08:44:41 starbug unix: [ID 930857 kern.info] avail mem = 121888768
Jan 3 08:44:41 starbug rootnex: [ID 466748 kern.info] root nexus =
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
.
.
.
#
```

x86 システムからは、次の dmesg 出力が表示されます。

```
# dmesg
Jan 2 07:21:46 naboo genunix: [ID 540533 kern.notice] SunOS Release 5.9 Version ...
Jan 2 07:21:46 naboo genunix: [ID 913631 kern.notice] Copyright 1983-2002 ...
Jan 2 07:21:46 naboo genunix: [ID 897550 kern.info] Using default device ...
Jan 2 07:21:46 naboo unix: [ID 168242 kern.info] mem = 130684K (0x7f9f000)
Jan 2 07:21:46 naboo unix: [ID 930857 kern.info] avail mem = 116547584
Jan 2 07:21:46 naboo rootnex: [ID 466748 kern.info] root nexus = i86pc
Jan 2 07:21:46 naboo rootnex: [ID 349649 kern.info] pci0 at root: ...
Jan 2 07:21:46 naboo genunix: [ID 936769 kern.info] pci0 is /pci@0,0
Jan 2 07:21:46 naboo genunix: [ID 678236 kern.info] Ethernet address = ...
.
.
.
```

システムへ周辺デバイスを追加する

新しい (ホットプラグイン不可) の周辺デバイスを追加する場合、通常、次の作業が必要になります。

- システムのシャットダウン
- システムへのデバイスの接続
- システムのリブート

システムにホットプラグインできない次のデバイスを追加する場合は、385 ページの「周辺デバイスを追加する方法」の手順に従ってください。

- CD-ROM
- 二次ディスクドライブ
- テープドライブ
- SBUS カード

場合によっては、新しいデバイスをサポートするために、Sun 以外のデバイスドライバを追加しなければなりません。

ホットプラグインデバイスについては、第 28 章を参照してください。

▼ 周辺デバイスを追加する方法

1. スーパーユーザーになります。
2. デバイスをサポートするためにデバイスドライバを追加する必要がある場合は、**386** ページの「デバイスドライバを追加する方法」の手順 **2** と **3** を実行します。
3. **/reconfigure** ファイルを作成します。

```
# touch /reconfigure
```

この **/reconfigure** ファイルがあると、Solaris ソフトウェアは、次にシステムに電源を入れたときまたはブートしたときに、新しくインストールされたデバイスがないかどうかをチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -g30 -y
```

-i0 システムを **init 0** 状態に戻す。システムの電源を落としてデバイスの追加、削除を行うのに適した状態になる。

-g30 システムを 30 秒以内にシャットダウンする。デフォルト値は 60 秒。

-y ユーザーの介入なしに、システムのシャットダウンを続ける。このオプションを指定しないと、シャットダウンプロセスを続けるかどうか、プロンプトでたずねられる。

5. システムがシャットダウンしたら、次のいずれかを選択して電源を落とします。
 - a. **SPARC** プラットフォームでは、**ok** プロンプトが表示されたら電源を落としても安全です。
 - b. **x86** プラットフォームでは、**type any key to continue** プロンプトが表示されたら電源を落としても安全です。電源スイッチの位置については、各システムに添付のハードウェアマニュアルを参照してください。
6. すべての外部デバイスの電源を落とします。
周辺デバイスの電源スイッチの位置については、各自の周辺デバイスに添付のハードウェアマニュアルを参照してください。
7. 周辺デバイスをインストールして、追加するデバイスのターゲット番号がシステム上の他のデバイスとは異なることを確認します。
ターゲット番号を選択するために、ディスクの裏側にある小さいスイッチを見つけてください。
デバイスの設置と接続については、周辺デバイスに添付のハードウェアマニュアルを参照してください。

8. システムの電源を入れます。
システムがブートされてマルチユーザーモードになり、ログインプロンプトが表示されます。
9. 周辺デバイスにアクセスし、そのデバイスが追加されたことを確認してください。
デバイスにアクセスする方法については、第 30 章を参照してください。

▼ デバイスドライバを追加する方法

この手順では、デバイスがすでにシステムに追加されていることを前提としています。追加されていない場合は、380 ページの「標準サポートされていないデバイスを使用する場合」を参照してください。

1. スーパーユーザーになります。
2. テープ、フロッピーディスク、または **CD-ROM** をドライブに入れます。
3. ドライバをインストールします。

```
# pkgadd -d device package-name
```

<code>-d device</code>	パッケージを含むデバイスのパス名を指定する。
<code>package-name</code>	デバイスドライバを含むパッケージ名を指定する。

4. パッケージが正常に追加されたことを確認します。

```
# pkgchk package-name  
#
```

パッケージが正しくインストールされている場合は、何も表示されません。

例 — デバイスドライバを追加する

次の例では、XYZdrv というパッケージをインストールして確認します。

```
# pkgadd XYZdrv  
(ライセンス関連のメッセージが表示される)  
.  
.  
Installing XYZ Company driver as <XYZdrv>  
.  
.  
Installation of <XYZdrv> was successful.  
# pkgchk XYZdrv  
#
```

第 28 章

デバイスの動的構成 (手順)

この章では、Solaris 環境でデバイスを動的に構成する手順について説明します。Solaris 環境でシステム構成要素がホットプラグ機能をサポートする場合、システムが動作しているときにも、デバイスを追加、削除、または交換することができます。システムの構成要素がホットプラグをサポートしていない場合は、システムをリブートしてデバイスを再構成します。

デバイスを動的に構成する手順については、次の項目を参照してください。

- 391 ページの「cfgadm コマンドによる SCSI ホットプラグ (作業マップ)」
- 401 ページの「cfgadm コマンドによる PCI ホットプラグ (作業マップ)」
- 408 ページの「アプリケーション開発者 RCM スクリプト (作業マップ)」
- 409 ページの「システム管理者 RCM スクリプト (作業マップ)」

cfgadm コマンドを使用して USB デバイスをホットプラグする方法については、438 ページの「cfgadm コマンドを使った USB デバイスのホットプラグ」を参照してください。

デバイスへのアクセス方法については、第 30 章を参照してください。

動的再構成とホットプラグ機能

「ホットプラグ機能」とは、システムの動作中に、システム構成要素を物理的に取り付け、取り外し、または交換できる機能のことです。「動的再構成」とは、システム構成要素をホットプラグできる機能のことです。また動的再構成は、システムリソースをシステムから物理的に取り外さなくても (なんらかの方法で) システムリソース (ハードウェアとソフトウェアの両方) をシステム内で移動したり、無効にできる機能のことです。

cfgadm コマンドを使用すると、次のデバイスをホットプラグできます。

- SPARC および x86 プラットフォームの USB デバイス

- SPARC および x86 プラットフォームの SCSI デバイス
- x86 プラットフォームの PCI デバイス

cfgadm コマンドには、次のような機能があります。

- システム構成要素の状態の表示
- システム構成要素の検査
- システム構成要素の構成の変更
- 構成ヘルプメッセージの表示

cfgadm コマンドでシステム構成要素を再構成する利点は、システムが動作しているときでも、システム構成要素を取り付け、取り外し、または交換できることです。さらに、cfgadm コマンドでは、システム構成要素を取り付け、取り外し、または交換するために必要な手順が示されます。

SCSI 構成要素をホットプラグする手順については、cfgadm(1M) のマニュアルページと 392 ページの「cfgadm コマンドによる SCSI ホットプラグ」を参照してください。x86 のシステムにおいて PCI アダプタカードをホットプラグする手順については、402 ページの「x86: cfgadm コマンドによる PCI ホットプラグ」を参照してください。

注 - すべての SCSI と PCI のコントローラが cfgadm コマンドによるホットプラグ機能をサポートしているわけではありません。

Sun が提供する高可用性の一部として、動的再構成は他の階層化製品 (代替パス指定やフェイルオーバーソフトウェアなど) とともに使用することをお勧めします。両方の製品ともデバイス障害に対する耐性を提供します。

高可用性ソフトウェアがなくても、障害が発生したデバイスを交換できます。この場合、適切なアプリケーションを手動で停止し、重要でないファイルシステムのマウントを手動で解除し、デバイスを取り付けまたは取り外します。

注 - 企業レベルのシステムなど、特定のハードウェア構成に対するデバイスのホットプラグについては、お使いのハードウェア構成のマニュアルを参照してください。

接続点

cfgadm コマンドは接続点についての情報を表示します。「接続点」とは、動的再構成を行うことができるシステム内の特定の場所のことです。

接続点は、次の要素から構成されています。

- 占有装置 (*occupant*): システムに構成できるハードウェア構成要素のことです。
- 受容体 (*receptacle*): 占有装置を受け入れる場所のことです。

接続点は、論理と物理の両方の接続点 ID (Ap_Id) で表現されます。物理 Ap_Id は接続点の物理的なパス名です。論理 Ap_Id は物理 Ap_Id に代わるユーザーに理解しやすい ID です。Ap_Id の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

通常、SCSI HBA (Host Bus Adapter)、つまり、SCSI コントローラの論理 Ap_Id はコントローラ番号 (c0 など) で表現されます。

コントローラ番号が SCSI HBA に割り当てられていない場合、内部的に生成された固有の識別子が提供されます。たとえば、SCSI コントローラの固有な識別子は次のようになります。

```
fas1:scsi
```

通常、SCSI デバイスの論理 Ap_Id は次のようになります。

```
HBA-logical-apid::device-identifier
```

次の例において、c0 は SCSI HBA の論理 Ap_Id です。

```
c0::dsk/c0t3d0
```

通常、デバイス識別子は /dev ディレクトリ内にある論理デバイス名から導き出されます。たとえば、論理デバイス名が /dev/rmt/1 のテープデバイスの論理 Ap_Id は次のようになります。

```
c0::rmt/1
```

SCSI デバイスの論理 Ap_Id を /dev ディレクトリ内にある論理デバイス名から導き出すことができない場合、内部的に生成された固有の識別子が提供されます。たとえば、/dev/rmt/1 テープデバイスの識別子は、次のようになります。

```
c0::st4
```

SCSI Ap_Id の詳細については、`cfgadm_scsi(1M)` のマニュアルページを参照してください。

`cfgadm` コマンドはすべてのリソースと動的再構成の操作を、一般的な状態 (`configured`、`unconfigured` など) や操作 (`connect`、`configure`、`unconfigure` など) を示す用語で表現します。一般的な状態や操作の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

次に、SCSI HBA 接続点の受容体と占有装置の状態を説明します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI HBA には利用できない	configured	1 つまたは複数のデバイスがバス上で構成されている

受容体の状態	説明	占有装置の状態	説明
disconnected	バスは休止している	unconfigured	デバイスは構成されていない
connected	バスはアクティブである		

次に、SCSI デバイス接続点の受容体と占有装置の状態を説明します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI デバイスには利用できない	configured	デバイスが構成されている
disconnected	バスは休止している	unconfigured	デバイスは構成されていない
connected	バスはアクティブである		

SCSI 接続点の状態は特別なハードウェアによって示されない限り未知です。SCSI 構成要素の構成情報を表示する手順については、392 ページの「SCSI デバイスに関する情報を表示する方法」を参照してください。

x86: PCI アダプタカードの取り外し

デバイスドライバがホットプラグ機能をサポートしている場合、必須でないシステムリソースとして機能している PCI アダプタカードは取り外すことができます。重要なシステムリソースとして機能している PCI アダプタカードは取り外すことができません。PCI アダプタカードが取り外し可能であるためには、次の条件が必要です。

- デバイスドライバはホットプラグ機能をサポートしていなければなりません。
- 重要なリソースには代替パスでアクセスできなければなりません。

たとえば、システムにインストールされているイーサネットカードが 1 つしかない場合、ネットワーク接続を切断せずにこのイーサネットカードを取り外すことは不可能です。このような環境でネットワーク接続をアクティブに保ったままイーサネットカードを取り外すには、別の階層化ソフトウェアサポートが必要です。

x86: PCI アダプタカードの取り付け

PCI アダプタカードをシステムに取り付けるには、次の条件が必要です。

- スロットが利用できなければなりません。
- デバイスドライバが当該アダプタカードのホットプラグ機能をサポートしていなければなりません。

PCI アダプタカードの取り付けまたは取り外しの手順については、402 ページの「x86: cfgadm コマンドによる PCI ホットプラグ」を参照してください。

cfgadm コマンドによる SCSI ホットプラグ (作業マップ)

作業	説明	参照先
1. SCSI デバイスについての情報の表示	SCSI コントローラおよびデバイスについての情報を表示する	392 ページの「SCSI デバイスに関する情報を表示する方法」
2. SCSI コントローラの構成解除	SCSI コントローラの構成を解除する	393 ページの「SCSI コントローラの構成を解除する方法」
3. SCSI コントローラの構成	構成を解除された SCSI コントローラを構成する	393 ページの「SCSI コントローラを構成する方法」
4. SCSI デバイスの構成	SCSI デバイスを構成する	394 ページの「SCSI デバイスを構成する方法」
5. SCSI コントローラを切り離す	特定の SCSI コントローラを切り離す	395 ページの「SCSI コントローラを切り離す方法」
6. SCSI コントローラの接続	切り離された特定の SCSI コントローラを接続する	396 ページの「SPARC: SCSI コントローラを接続する方法」
7. SCSI デバイスを SCSI バスに取り付ける	特定の SCSI デバイスを SCSI バスに取り付ける	396 ページの「SPARC: SCSI デバイスを SCSI バスに取り付ける方法」
8. SCSI コントローラ上の同一デバイスと交換	SCSI バス上のデバイスを、同じタイプの別のデバイスに交換する	397 ページの「SPARC: SCSI コントローラ上の同一デバイスを交換する方法」
9. SCSI デバイスを取り外す	SCSI デバイスをシステムから取り外す	398 ページの「SPARC: SCSI デバイスを取り外す方法」
10. SCSI 構成に関する問題の障害追跡	失敗した SCSI 構成解除操作を解決する	401 ページの「失敗した SCSI 構成解除操作の解決方法」

cfgadm コマンドによる SCSI ホットプラグ

この節では、`cfgadm` コマンドを使用してさまざまな SCSI ホットプラグ処理を実行する方法について説明します。

この節で説明する手順では、特定のデバイスを使用して、`cfgadm` コマンドで SCSI 構成要素をホットプラグする例を示します。`cfgadm` コマンドで提供されるデバイス情報や表示されるデバイス情報は、システム構成によって異なります。

▼ SCSI デバイスに関する情報を表示する方法

次の手順では、SCSI コントローラ `c0` と `c1`、およびコントローラに接続されたデバイスを例として使用して、`cfgadm` コマンドを使用して表示可能なデバイス構成情報を示します。

注 - SCSI デバイスが `cfgadm` コマンドでサポートされていない場合、その SCSI デバイスは `cfgadm` コマンドの出力には表示されません。

1. スーパーユーザーになります。
2. システムの接続点についての情報を表示します。

```
# cfgadm -l
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
```

この例では、`c0` と `c1` は 2 つの SCSI コントローラを表しています。

3. システムの **SCSI** コントローラとこれらに接続されているデバイスについての情報を表示します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::disk/c0t0d0  disk         connected   configured  unknown
c0::rmt/0       tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::disk/c1t3d0  disk         connected   configured  unknown
c1::disk/c1t4d0  unavailable  connected   unconfigured unknown
```

注 - `cfgadm -l` コマンドは、SCSI デバイスではなく、SCSI HBA についての情報を表示します。ディスクやテープなどの SCSI デバイスについての情報を表示するには、`cfgadm -al` コマンドを使用してください。

次の手順では、SCSI 接続点だけが表示されます。画面に表示される接続点はシステムによって異なります。

▼ SCSI コントローラの構成を解除する方法

次の手順では、SCSI コントローラ `c1` を使用して、SCSI デバイスの構成を解除する例を示します。

1. スーパーユーザーになります。
2. SCSI コントローラの構成を解除します。

```
# cfgadm -c unconfigure c1
```

3. SCSI コントローラの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus           connected   configured unknown
c0::dsk/c0t0d0       disk               connected   configured unknown
c0::rmt/0            tape               connected   configured unknown
c1                   scsi-bus           connected   unconfigured unknown
```

`c1` の `Occupant` の列に `unconfigured` と表示されていることに注目してください。これは、SCSI バスに占有装置が構成されていないことを示します。

構成解除処理が失敗した場合は、401 ページの「失敗した SCSI 構成解除操作の解決方法」を参照してください。

▼ SCSI コントローラを構成する方法

次の手順では、SCSI コントローラ `c1` を使用して、SCSI コントローラを構成する例を示します。

1. スーパーユーザーになります。
2. SCSI コントローラを構成します。

```
# cfgadm -c configure c1
```

3. SCSI コントローラが構成されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
```

c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	unavailable	connected	unconfigured	unknown

前述の構成を解除する例では、SCSI バス上のすべてのデバイスを削除しました。この例では、すべてのデバイスをシステムに構成し直します。

▼ SCSI デバイスを構成する方法

次の手順では、SCSI ディスク c1t4d0 を使用して、SCSI デバイスを構成する例を示します。

1. スーパーユーザーになります。
2. 構成するデバイスを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 unavailable  connected   unconfigured unknown
```

3. SCSI デバイスを構成します。

```
# cfgadm -c configure c1::dsk/c1t4d0
```

4. SCSI デバイスが構成されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 disk         connected   configured  unknown
```

▼ SCSI コントローラを切り離す方法



注意 - SCSI デバイスを切り離すときは十分に注意してください。特に、ルート (/)、usr、var、swap パーティションなどの重要なファイルシステムが入っているディスクのコントローラを扱うときは細心の注意を払ってください。動的再構成ソフトウェアは、システムがハングする原因をすべて発見できるわけではありません。この手順は、十分注意して実行してください。

次の手順では、SCSI コントローラ c1 を使用して、SCSI デバイスを切り離す例を示します。

1. スーパーユーザーになります。
2. デバイスを切り離す前に、デバイスが接続されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 disk         connected   configured  unknown
```

3. SCSI コントローラを切り離します。

```
# cfgadm -c disconnect c1
WARNING: Disconnecting critical partitions may cause system hang.
Continue (yes/no)? y
```



注意 - このコマンドは、cfgadm -c connect コマンドを使用するまで、SCSI バス上のすべての入出力動作を中断します。cfgadm コマンドは基本的な検査を行い、重要なパーティションが切り離されるのを防ぎます。しかし、すべての場合を発見できるわけではありません。このコマンドの使い方が不適切な場合、システムがハングし、システムをリブートしなければならなくなる可能性もあります。

4. SCSI バスが切り離されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             unavailable  disconnected  configured  unknown
c1::dsk/c1t10d0 unavailable  disconnected  configured  unknown
c1::dsk/c1t4d0 unavailable  disconnected  configured  unknown
```

コントローラとそれに接続されていたすべてのデバイスがシステムから切り離されました。

▼ SPARC: SCSI コントローラを接続する方法

次の手順は、SCSI コントローラ `c1` を使用して、SCSI コントローラを接続する例を示します。

1. スーパーユーザーになります。
2. デバイスを接続する前に、デバイスが切り離されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::disk/c0t0d0  disk         connected   configured  unknown
c0::rmt/0       tape         connected   configured  unknown
c1             unavailable  disconnected  configured  unknown
c1::disk/c1t10d0  unavailable  disconnected  configured  unknown
c1::disk/c1t4d0  unavailable  disconnected  configured  unknown
```

3. SCSI コントローラを接続します。

```
# cfgadm -c connect c1
```

4. SCSI コントローラが接続されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::disk/c0t0d0  disk         connected   configured  unknown
c0::rmt/0       tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::disk/c1t3d0  disk         connected   configured  unknown
c1::disk/c1t4d0  disk         connected   configured  unknown
```

▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法

SCSI コントローラ `c1` を使用して、SCSI デバイスを SCSI バスに取り付ける方法を説明します。

注 - デバイスを取り付けるときは、デバイス自身の `Ap_Id` ではなく、デバイスを取り付ける SCSI HBA (コントローラ) の `Ap_Id` を指定します。

1. スーパーユーザーになります。

- 現在の SCSI 構成を確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus           connected   configured unknown
c0::dsk/c0t0d0      disk               connected   configured unknown
c0::rmt/0            tape               connected   configured unknown
c1                   scsi-bus           connected   configured unknown
c1::dsk/c1t3d0      disk               connected   configured unknown
```

- SCSI デバイスを SCSI バスに取り付けます。

```
# cfgadm -x insert_device c1
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000
This operation will suspend activity on SCSI bus: c1
```

- Continue (yes/no)? というプロンプトに **y** と入力して、次に進みます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

- デバイスを接続して、電源を入れます。

- Enter y if operation is complete or n to abort (yes/no)? というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

- デバイスが取り付けられていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus           connected   configured unknown
c0::dsk/c0t0d0      disk               connected   configured unknown
c0::rmt/0            tape               connected   configured unknown
c1                   scsi-bus           connected   configured unknown
c1::dsk/c1t3d0      disk               connected   configured unknown
c1::dsk/c1t4d0      disk               connected   configured unknown
```

コントローラ c1 に新しいディスクが取り付けられました。

▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法

次の手順では、SCSI ディスク c1t4d0 を使用して、SCSI コントローラ上の同一デバイスを交換する例を示します。

- スーパーユーザーになります。
- 現在の SCSI 構成を確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant      Condition
c0                   scsi-bus            connected    configured    unknown
c0::dsk/c0t0d0       disk                 connected    configured    unknown
c0::rmt/0             tape                 connected    configured    unknown
c1                   scsi-bus            connected    configured    unknown
c1::dsk/c1t3d0       disk                 connected    configured    unknown
c1::dsk/c1t4d0       disk                 connected    configured    unknown
```

3. SCSI バス上のデバイスを、同じタイプの別のデバイスに交換します。

```
# cfgadm -x replace_device c1::dsk/c1t4d0
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
```

- a. **Continue (yes/no)?** というプロンプトに **y** と入力して、次に進みます。
ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

- b. デバイスの電源を切ってから、そのデバイスを取り外します。
c. 交換用のデバイスを取り付けます。そして、取り付けたデバイスの電源を入れます。
交換用のデバイスは取り外したデバイスと同じタイプであり、同じアドレス (ターゲットと論理ユニット番号) でなければなりません。

- d. **Enter y if operation is complete or n to abort (yes/no)?** というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. デバイスが交換されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant      Condition
c0                   scsi-bus            connected    configured    unknown
c0::dsk/c0t0d0       disk                 connected    configured    unknown
c0::rmt/0             tape                 connected    configured    unknown
c1                   scsi-bus            connected    configured    unknown
c1::dsk/c1t3d0       disk                 connected    configured    unknown
c1::dsk/c1t4d0       disk                 connected    configured    unknown
```

▼ SPARC: SCSI デバイスを取り外す方法

次に、SCSI ディスク c1t4d0 を使用して、SCSI コントローラ上のデバイスを取り外す例を示します。

1. スーパーユーザーになります。

2. 現在の SCSI 構成を確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected    configured unknown
c0::dsk/c0t0d0       disk                 connected    configured unknown
c0::rmt/0             tape                 connected    configured unknown
c1                   scsi-bus            connected    configured unknown
c1::dsk/c1t3d0       disk                 connected    configured unknown
c1::dsk/c1t4d0       disk                 connected    configured unknown
```

3. SCSI デバイスをシステムから取り外します。

```
# cfgadm -x remove_device c1::dsk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
```

a. Continue (yes/no)? というプロンプトに **y** と入力して、次に進みます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。
```

b. デバイスの電源を切ってから、そのデバイスを取り外します。

c. Enter y if operation is complete or n to abort (yes/no)? というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. デバイスがシステムから取り外されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected    configured unknown
c0::dsk/c0t0d0       disk                 connected    configured unknown
c0::rmt/0             tape                 connected    configured unknown
c1                   scsi-bus            connected    configured unknown
c1::dsk/c1t3d0       disk                 connected    configured unknown
```

SPARC: SCSI 構成に関する問題の障害追跡

この節では、SCSI 構成に関する問題の障害追跡を行うために、エラーメッセージとその解決策について説明します。SCSI 構成に関する問題の障害追跡の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource                Information
-----
```

```
/dev/dsk/c1t0d0s0  mounted filesystem "/file-system"
```

原因

ファイルシステムがマウントされているデバイスを取り外しまたは交換しようとしてしました。

解決法

エラーメッセージのリストにあるファイルシステムのマウントを解除してから、もう一度 `cfgadm` コマンドを実行します。

`cfgadm` コマンドを使用して、スワップデバイス、専用のダンプデバイスなどのシステムリソースを取り外すと、システムリソースがアクティブな場合、次のようなエラーメッセージが表示されます。

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource              Information
-----
/dev/dsk/device-name  swap area
```

原因

1 つまたは複数の構成されているスワップ領域を削除または置き換えようとしてしました。

解決法

指定されたデバイス上のスワップ領域の構成を解除してから、再度 `cfgadm` を実行します。

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource              Information
-----
/dev/dsk/device-name  dump device (swap)
```

原因

スワップ領域上に構成されているダンプデバイスを取り外そうとしたか、置き換えようとしてしました。

解決法

スワップ領域に構成されているダンプデバイスの構成を解除してから、再度 `cfgadm` を実行します。

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource              Information
-----
/dev/dsk/device-name  dump device (dedicated)
```

原因

専用ダンプデバイスを取り外そうとしたか、置き換えようとしてしました。

解決法

専用ダンプデバイスの構成を解除し、`cfgadm` 処理を再実行します。

▼ 失敗した SCSI 構成解除操作の解決方法

1 つ以上のターゲットデバイスが使用中である場合、および SCSI 構成解除操作が失敗した場合、次の手順を使用します。この手順を使用しないと、将来、このコントローラおよびターゲットデバイスの動的再構成操作が失敗し、`dr in progress` メッセージが表示されます。

1. スーパーユーザーになります (まだなっていない場合)。
2. 次のコマンドを入力してコントローラを再構成します。

```
# cfgadm -c configure device-name
```

cfgadm コマンドによる PCI ホットプラグ (作業マップ)

作業	説明	参照先
1. PCI スロット構成情報の表示	システムの PCI ホットプラグ可能なデバイスとスロットの状態を表示する	402 ページの「x86: PCI スロット構成情報を表示する方法」
2. PCI アダプタカードを取り外す	カードを構成解除してからスロットから電源を外し、システムからカードを取り外す	403 ページの「x86: PCI アダプタカードを取り外す方法」
3. PCI アダプタカードを取り付ける	アダプタカードをホットプラグ可能なスロットに挿入する。スロットを電源に接続し、カードを構成する	403 ページの「x86: PCI アダプタカードを取り付ける方法」
4. PCI 構成に関する問題の障害追跡	PCI 構成障害に対処するために、エラーメッセージと解決策を確認する	404 ページの「x86: PCI 構成に関する問題の障害追跡」

x86: cfgadm コマンドによる PCI ホットプラグ

この節では、x86 システム上で PCI アダプタカードをホットプラグする手順について説明します。

次の例では、簡潔にするため、PCI 接続点だけを表示しています。画面に表示される接続点はシステムによって異なります。

▼ x86: PCI スロット構成情報を表示する方法

cfgadmin コマンドは、システム上の PCI ホットプラグ可能なデバイスとスロットの状態を表示します。詳細については `cfgadm(1M)` のマニュアルページを参照してください。

1. スーパーユーザーになります。
2. PCI スロット構成情報を表示します。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant    Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        connected   configured  ok
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

具体的な PCI デバイス情報を表示します。

```
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                Type                Information
pci1:hpc0_slot0     unknown            Slot 7
pci1:hpc0_slot1     unknown            Slot 8
pci1:hpc0_slot2     unknown            Slot 9
pci1:hpc0_slot3     ethernet/hp        Slot 10
pci1:hpc0_slot4     unknown            Slot 11
```

論理 Ap_Id の `pci1:hpc0_slot0` は、ホットプラグ可能なスロット Slot 7 の論理 Ap_Id です。構成要素 `hpc0` はこのスロットのホットプラグ可能なアダプタカードを示し、`pci1` は PCI バスのインスタンスを示します。Type フィールドは、スロット中の PCI アダプタカードのタイプを示します。

▼ x86: PCI アダプタカードを取り外す方法

1. スーパーユーザーになります。
2. PCI アダプタカードが入っているスロットを確認します。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        connected   configured ok
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

3. デバイスを開いているアプリケーションを停止します。
たとえば、イーサネットカードの場合、ifconfig コマンドでインタフェースを無効にしてから、カードを引き抜きます。
4. デバイスの構成を解除します。

```
# cfgadm -c unconfigure pci1:hpc0_slot3
```

5. デバイスの構成が解除されていることを確認します。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        connected   unconfigured unknown
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

6. スロットへの電源を切り離します。
- ```
cfgadm -c disconnect pci1:hpc0_slot3
```
7. デバイスが切り離されていることを確認します。

```
cfgadm
Ap_Id Type Receptacle Occupant Condition
pci1:hpc0_slot0 unknown empty unconfigured unknown
pci1:hpc0_slot1 unknown empty unconfigured unknown
pci1:hpc0_slot2 unknown empty unconfigured unknown
pci1:hpc0_slot3 ethernet/hp disconnected unconfigured unknown
pci1:hpc0_slot4 unknown empty unconfigured unknown
```

8. スロットのラッチを開いて PCI アダプタカードを取り外します。

## ▼ x86: PCI アダプタカードを取り付ける方法

1. スーパーユーザーになります。
2. ホットプラグ可能なスロットを確認して、ラッチを開きます。

- ホットプラグ可能なスロットに **PCI** アダプタカードを挿入します。
- PCI** アダプタカードを挿入した後に、どのスロットに **PCI** アダプタカードが入っているかを確認します。ラッチを閉じます。

```
cfgadm
Ap_Id Type Receptacle Occupant Condition
pci1:hpc0_slot0 unknown empty unconfigured unknown
pci1:hpc0_slot1 unknown empty unconfigured unknown
pci1:hpc0_slot2 unknown empty unconfigured unknown
pci1:hpc0_slot3 ethernet/hp disconnected unconfigured unknown
pci1:hpc0_slot4 unknown empty unconfigured unknown
```

- スロットへの電源を接続します。

```
cfgadm -c connect pci1:hpc0_slot3
```

- スロットが接続されていることを確認します。

```
cfgadm
Ap_Id Type Receptacle Occupant Condition
pci1:hpc0_slot0 unknown empty unconfigured unknown
pci1:hpc0_slot1 unknown empty unconfigured unknown
pci1:hpc0_slot2 unknown empty unconfigured unknown
pci1:hpc0_slot3 ethernet/hp connected unconfigured unknown
pci1:hpc0_slot4 unknown empty unconfigured unknown
```

- PCI** アダプタカードを構成します。

```
cfgadm -c configure pci1:hpc0_slot3
```

- スロット中の **PCI** アダプタカードの構成を確認します。

```
cfgadm
Ap_Id Type Receptacle Occupant Condition
pci1:hpc0_slot0 unknown empty unconfigured unknown
pci1:hpc0_slot1 unknown empty unconfigured unknown
pci1:hpc0_slot2 unknown empty unconfigured unknown
pci1:hpc0_slot3 ethernet/hp connected configured unknown
pci1:hpc0_slot4 unknown empty unconfigured unknown
```

- 新しいデバイスの場合、サポートソフトウェアを構成します。

たとえば、デバイスがイーサネットカードの場合、ifconfig コマンドでインタフェースを設定します。

## x86: PCI 構成に関する問題の障害追跡

エラーメッセージ

```
cfgadm: Configuration operation invalid: invalid transition
```

原因

無効な移行を行いました。

#### 解決法

`cfgadm -c` コマンドが適切に発行されているかどうかを確認します。`cfgadm` コマンドで現在の受容体と占有装置の状態を確認し、`Ap_id` が正しいことを確認します。

#### エラーメッセージ

```
cfgadm: Attachment point not found
```

#### 原因

指定した接続点は見つかりません。

#### 解決法

接続点が正しいかどうかを確認します。`cfgadm` コマンドを使用して、利用可能な接続点のリストを表示します。また、物理パスを調べて、接続点が現在も残っているかどうかを確認します。

---

注 - `cfgadm` コマンド以外にも、ホットプラグ処理中に便利なコマンドがいくつかあります。`prtconf` コマンドは、Solaris がハードウェアを認識するかどうかを表示します。ハードウェアの挿入後に `prtconf` コマンドを使用して、ハードウェアが認識されているかどうかを確認します。構成後に `prtconf -D` コマンドを使用して、新たにインストールしたハードウェアデバイスにドライバが設定されているかどうかを確認します。

---

## Reconfiguration Coordination Manager (RCM) スクリプトの概要

Reconfiguration Coordination Manager (RCM) は、システムコンポーネントの動的な除去を管理するフレームワークです。RCM を使用すると、システムリソースを順番に登録および解放することができます。

新しい RCM スクリプト機能を使用すると、アプリケーションを停止したり、動的な再構成の間にアプリケーションからデバイスを手際良く解放したりする独自のスクリプトを記述できます。スクリプトによって登録されたリソースに要求が影響を与える場合、RCM フレームワークは再構成要求に応じてスクリプトを自動的に起動します。

リソースを動的に除去する場合は、アプリケーションからリソースを手動で解放しておく必要があります。あるいは、`-f` オプションを指定して `cfgadm` コマンドを使用して再構成オペレーションを強制することも可能でした。ただし、このオプションはアプリケーションを認識不能な状態のままにする可能性があります。また、アプリケーションからリソースを手動で解放すると、一般にエラーが発生します。

RCM スクリプト機能を使うと、動的再構成処理を簡単かつ効果的に実行できます。RCM スクリプトを作成すると、次の操作を実行できます。

- 動的にデバイスを取り外したときにデバイスが自動的に解放される。デバイスがアプリケーションによって起動した場合は、この処理によって、デバイスも終了する
- システムからデバイスを動的に取り外すときに、サイト固有のタスクを実行する

## RCM スクリプトについて

RCM スクリプトは、次のとおりです。

- 実行可能シェルスクリプト (Perl、sh、csh、または ksh) または RCM デーモンが実行するバイナリプログラム。推奨言語は、Perl
- スクリプトファイル所有者のユーザー ID を使用することにより、自分のアドレス領域で実行されるスクリプト
- `cfgadm` コマンドを使ってシステムリソースを動的に再構成するときに、RCM デーモンによって実行されるスクリプト

## RCM スクリプトで実行できること

RCM スクリプトを使用した場合、デバイスを動的に取り外すと、デバイスがアプリケーションから解放されます。デバイスが開いている場合には、RCM スクリプトによって閉じられます。

たとえば、テープバックアップアプリケーションで RCM スクリプトを使用して、テープドライブを終了させたり、テープバックアップアプリケーションをシャットダウンしたりすることができます。

## RCM スクリプト処理の動作方法

次のようにしてスクリプトを起動します。

```
$ script-name command [args ...]
```

スクリプトにより、次の基本的な手順が実行されます。

1. コマンド行引数から RCM コマンドを取得する
2. コマンドを実行する
3. 結果を名前と値のペアで `stdout` に記述する
4. 適切な終了ステータスで終了する

RCM デーモンは、スクリプトのインスタンスを同時に 1 つ実行します。たとえば、RCM デーモンは、スクリプトの実行中には、そのスクリプトが終了するまで同じスクリプトを実行しません。

## RCM スクリプトコマンド

次の RCM コマンドを RCM スクリプトに含める必要があります。

- `scriptinfo` - スクリプト情報を収集する
- `register` - リソースに処理対象を登録する
- `resourceinfo` - リソース情報を収集する

次の RCM コマンドの一部またはすべてを RCM スクリプトに含めることができます。

- `queryremove` - リソースが解放されたかどうかを問い合わせる
- `preremove` - リソースを解放する
- `postremove` - リソースの削除後に通知する
- `undoremove` - `preremove` で実行された動作を元に戻す

これらの RCM コマンドの詳細については、`rcmscript(4)` のマニュアルページを参照してください。

## RCM スクリプト処理環境

デバイスを動的に取り外すと、RCM デーモンにより次のコマンドが実行されます。

- スクリプトで識別されたリソースリスト (デバイス名) を収集するための、スクリプトの `register` コマンドが実行される
- スクリプトの登録されたリソースが動的な取り外し操作によって影響を受ける場合、リソースを取り外す前にスクリプトの `queryremove/preremove` コマンドが実行される
- 取り外し操作が成功した場合に、スクリプトの `postremove` コマンドが実行される。ただし、取り外し操作に失敗した場合、RCM デーモンによりスクリプトの `undoremove` コマンドが実行される

## RCM スクリプトでの作業

次の節では、アプリケーション開発者およびシステム管理者のために RCM スクリプト作業について説明します。

## アプリケーション開発者 RCM スクリプト (作業マップ)

次の作業マップでは、RCM スクリプトを作成するアプリケーション開発者の作業について説明します。

| 作業                            | 説明                                                     | 参照先                               |
|-------------------------------|--------------------------------------------------------|-----------------------------------|
| 1. アプリケーションが使用するリソースを特定する     | アプリケーションが使用するリソース (デバイス名) を特定する。このデバイスは動的に取り外される可能性がある | cfgadm (1M)                       |
| 2. リソースを解放するコマンドを特定する         | アプリケーションからリソースを完全に解放するようにアプリケーションに通知するコマンドを特定する        | アプリケーションのマニュアル                    |
| 3. リソースを取り外した後に使用するコマンドを特定する  | リソースを取り外したことをアプリケーションに通知するコマンドを含める                     | rcmscript (4)                     |
| 4. リソースの取り外しに失敗した場合のコマンドを特定する | 使用可能なリソースについてアプリケーションに通知するコマンドを含める                     | rcmscript (4)                     |
| 5. RCM スクリプトを記述する             | 前の作業で特定した情報に基づいて RCM スクリプトを記述する                        | 411 ページの「テープバックアップ用の RCM スクリプトの例」 |
| 6. RCM スクリプトをインストールする         | 適切なスクリプトディレクトリにスクリプトを追加する                              | 410 ページの「RCM スクリプトのインストール方法」      |
| 7. RCM スクリプトをテストする            | 手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストする               | 410 ページの「RCM スクリプトのテスト方法」         |



## システム管理者 RCM スクリプト (作業マップ)

ここでは、サイトをカスタマイズするために RCM スクリプトを作成するシステム管理者の作業について説明します。

| 作業                              | 説明                                             | 参照先                               |
|---------------------------------|------------------------------------------------|-----------------------------------|
| 1. 動的に移動するリソースを特定する             | cfgadm -l コマンドを使って移動する可能性があるリソース (デバイス名) を特定する | cfgadm (1M)                       |
| 2. 停止するアプリケーションを特定する            | アプリケーションを完全に停止させるコマンドを特定する                     | アプリケーションのマニュアル                    |
| 3. リソースの取り外し前および取り外し後のコマンドを特定する | リソースを取り外す前後の動作を特定する                            | rcmscript (4)                     |
| 4. RCM スクリプトを記述する               | 前の作業で特定した情報に基づいて RCM スクリプトを記述する                | 411 ページの「テープバックアップ用の RCM スクリプトの例」 |
| 5. RCM スクリプトをインストールする           | 適切なスクリプトディレクトリにスクリプトを追加する                      | 410 ページの「RCM スクリプトのインストール方法」      |
| 6. RCM スクリプトをテストする              | 手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストする       | 410 ページの「RCM スクリプトのテスト方法」         |

## RCM スクリプトに名前を付ける

次の規則に従って、スクリプトに *vendor*、*service* という名前を付ける必要があります。

|                |                                         |
|----------------|-----------------------------------------|
| <i>vendor</i>  | スクリプトを提供するベンダーのストックシンボル、またはベンダーを識別する固有名 |
| <i>service</i> | スクリプトが表すサービス名                           |

## RCM スクリプトのインストールまたは削除

RCM スクリプトのインストールまたは削除を行うには、スーパーユーザー (*root*) の権限が必要です。この表を使用して、RCM スクリプトをインストールするディレクトリを判断してください。

表 28-1 RCM スクリプトディレクトリ

| ディレクトリの位置                                | スクリプトタイプ            |
|------------------------------------------|---------------------|
| /etc/rcm/scripts                         | 特定のシステム用のスクリプト      |
| /usr/platform/`uname -i`/lib/rcm/scripts | 特定のハードウェア実装用のスクリプト  |
| /usr/platform/`uname -m`/lib/rcm/scripts | 特定のハードウェアクラス用のスクリプト |
| /usr/lib/rcm/scripts                     | 任意のハードウェア用のスクリプト    |

## ▼ RCM スクリプトのインストール方法

1. スーパーユーザーになります。
2. スクリプトを表 28-1 で説明されている適切なディレクトリにコピーします。  
たとえば、次のようになります。

```
cp SUNW, sample.pl /usr/lib/rcm/scripts
```

3. スクリプトのユーザー ID およびグループ ID を希望の値に変更します。

```
chown user:group /usr/lib/rcm/scripts/SUNW, sample.pl
```

4. SIGHUP を RCM デーモンに送信します。

```
pkill -HUP -x -u root rcm_daemon
```

## ▼ RCM スクリプトの削除方法

1. スーパーユーザーになります。
2. RCM スクリプトディレクトリからスクリプトを削除します。  
たとえば、次のようになります。

```
rm /usr/lib/rcm/scripts/SUNW, sample.pl
```

3. SIGHUP を RCM デーモンに送信します。

```
pkill -HUP -x -u root rcm_daemon
```

## ▼ RCM スクリプトのテスト方法

1. スクリプトを実行する前にコマンド行シェルに RCM\_ENV\_FORCE などの環境変数を設定します。  
たとえば、Korn シェルで次のように設定します。

```
$ export RCM_ENV_FORCE=TRUE
```

2. コマンド行から手動でスクリプトコマンドを実行してスクリプトをテストします。  
たとえば、次のようになります。

```
$ script-name scriptinfo
$ script-name register
$ script-name preremove resource-name
$ script-name postremove resource-name
```

3. スクリプトの各 RCM スクリプトコマンドにより、適切な出力結果が **stdout** に印刷されるかどうかを確認します。
4. 適切なスクリプトディレクトリにスクリプトをインストールします。  
詳細については、410 ページの「RCM スクリプトのインストール方法」を参照してください。
5. 動的な削除操作を実行してスクリプトをテストします。  
たとえば、スクリプトによってデバイス /dev/dsk/c1t0d0s0 が登録されたとします。次のコマンドを実行してください。

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```



---

注意 – 上記のコマンドは、システムの状態を変化させたり、システム障害を招くおそれもあるため、これらのコマンドを十分理解しておくことは大切です。

---

## テープバックアップ用の RCM スクリプトの例

ここでは、RCM スクリプトを使用したテープバックアップの例を示します。

### テープバックアップ用の RCM スクリプトの役割

テープバックアップ用の RCM スクリプトは、次の手順を実行します。

1. RCM コマンドのディスパッチテーブルを設定します。
2. 指定した RCM コマンドに対応するディスパッチルーチンを呼び出し、未実装の RCM コマンドのステータス 2 で終了させます。
3. `scriptinfo` セクションを設定します。

```
rcm_script_func_info=Tape backup appl script for DR
```

4. すべてのテープドライブのデバイス名を **stdout** に印刷して、すべてのテープドライブをシステムに登録します。

```
rcm_resource_name=/dev/rmt/$f
エラーが発生した場合、エラー情報を stdout に印刷します。
```

```
rcm_failure_reason=$errmsg
```

5. テープデバイスのリソース情報を設定します。

```
rcm_resource_usage_info=Backup Tape Unit Number $unit
```

6. バックアップアプリケーションがそのデバイスを使用しているかどうか確認して、`preremove` 情報を設定します。バックアップアプリケーションがそのデバイスを使用していない場合、動的再構成操作が続行されます。バックアップアプリケーションがそのデバイスを使用している場合、スクリプトにより `RCM_ENV_FORCE` が検査されます。`RCM_ENV_FORCE` が `FALSE` に設定されている場合、スクリプトにより動的再構成操作が拒否され、次のメッセージが印刷されます。

```
rcm_failure_reason=tape backup in progress pid=...
```

`RCM_ENV_FORCE` が `TRUE` に設定されている場合、バックアップアプリケーションが停止し、再構成操作が続行されます。

## テープバックアップ再構成シナリオの結果

RCM スクリプトを使わずに `cfgadm` コマンドを使ってテープドライブを取り外した場合、次のような結果になります。

- バックアップアプリケーションがテープドライブを使用していない場合に `cfgadm` コマンドを使用すると、操作は正常に実行されます。
- バックアップアプリケーションがテープドライブを使用中に `cfgadm` コマンドを使用すると、操作が失敗します。

RCM スクリプトと `cfgadm` コマンドを使ってテープドライブを取り外した場合、次のような結果になります。

- バックアップアプリケーションがテープドライブを使用していない場合に `cfgadm` コマンドを使用すると、操作は正常に実行されます。
- バックアップアプリケーションがテープドライブを使用しているときに、`-f` オプションを指定せずに `cfgadm` コマンドを使用すると、次のようなメッセージが表示され、操作が失敗します。

```
tape backup in progress pid=...
```

- バックアップアプリケーションがテープドライブを使用しているときに、`-f` オプションを指定して `cfgadm` コマンドを使用すると、スクリプトによってバックアップアプリケーションが停止され、`cfgadm` 操作が正常に実行されます。

## 例 — テープバックアップ用の RCM スクリプト

```
#!/usr/bin/perl -w

サイト用にカスタマイズされた RCM スクリプトの例
```

```

#
RCM_ENV_FORCE が FALSE の場合、
RCM はテープドライブがバックアップのために
使用されているときは、ドライブを解放できない
#
RCM_ENV_FORCE が TRUE の場合、
DR はテープドライブをバックアップのために
使用しているバックアップアプリケーションを終了してテープドライブを
取り外すことができる。
#

use strict;

my ($cmd, %dispatch);
$cmd = shift(@ARGV);
RCM コマンドのテーブルをディスパッチする
%dispatch = (
 "scriptinfo" => \&do_scriptinfo,
 "register" => \&do_register,
 "resourceinfo" => \&do_resourceinfo,
 "queryremove" => \&do_preremove,
 "preremove" => \&do_preremove
);

if (defined(%dispatch{$cmd})) {
 &{%dispatch{$cmd}};
} else {
 exit (2);
}

sub do_scriptinfo
{
 print "rcm_script_version=1\n";
 print "rcm_script_func_info=Tape backup appl script for DR\n";
 exit (0);
}

sub do_register
{
 my ($dir, $f, $errmsg);

 $dir = opendir(RMT, "/dev/rmt");
 if (!$dir) {
 $errmsg = "Unable to open /dev/rmt directory: $!";
 print "rcm_failure_reason=$errmsg\n";
 exit (1);
 }

 while ($f = readdir(RMT)) {
 # 非表示のファイルや同一デバイスの複数の名前を無視する
 if (($f !~ /\^\./) && ($f =~ /^[0-9]+$)) {
 print "rcm_resource_name=/dev/rmt/$f\n";
 }
 }
}

```

```

 }

 closedir(RMT);
 exit (0);
}
sub do_resourceinfo
{
 my ($rsrc, $unit);

 $rsrc = shift(@ARGV);
 if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
 $unit = $1;
 print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
 exit (0);
 } else {
 print "rcm_failure_reason=Unknown tape device!\n";
 exit (1);
 }
}

sub do_preremove
{
 my ($rsrc);

 $rsrc = shift(@ARGV);

 # このリソースをバックアップアプリケーションが
 # 使用している場合にチェックする
 # if ($rsrc 上でバックアップアプリケーション
 # が動作していない場合) {
 # # DR を続行させる
 # exit (0);
 # }
 #
 # RCM_ENV_FORCE が FALSE の場合、処理を拒否する
 # RCM_ENV_FORCE が TRUE の場合、
 # バックアップアプリケーションを終了して
 # DR が処理を続行できるようにする
 #
 if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
 if ($cmd eq 'preremove') {
 # kill the tape backup application
 }
 exit (0);
 } else {
 #
 # テープバックアップアプリケーションによって
 # デバイスが使用されていたため、テープドライブを
 # 解放できなかったことを示す
 #
 print "rcm_failure_reason=tape backup in progress pid=...\n"
;

 exit (3);
 }
}
}

```

## 第 29 章

# USB デバイスの使用 (概要/手順)

---

この章では、USB (Universal Serial Bus) デバイスの概要と、Solaris 環境で USB デバイスを使用する手順について説明します。

USB デバイスを使った手順については、次の節を参照してください。

- 423 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」
- 431 ページの「USB オーディオデバイスの使用 (作業マップ)」
- 438 ページの「`cfgadm` コマンドを使用した USB デバイスのホットプラグ (作業マップ)」

この章の内容は次のとおりです。

- 415 ページの「USB デバイスの概要」
- 420 ページの「Solaris 環境における USB について」
- 424 ページの「USB 大容量ストレージデバイスの使用」
- 424 ページの「USB デバイスのホットプラグ」
- 431 ページの「USB オーディオデバイスの使用」
- 437 ページの「USB オーディオデバイスに関する問題の障害追跡」
- 438 ページの「`cfgadm` コマンドを使った USB デバイスのホットプラグ」

動的再構成およびホットプラグについての一般的な情報については、第 28 章を参照してください。

USB プリンタの構成情報については、『Solaris のシステム管理 (上級編)』の「印刷の新機能」を参照してください。

---

## USB デバイスの概要

Universal Serial Bus (USB) は PC 業界で開発された、周辺機器 (キーボード、マウス、プリンタなど) をシステムに接続するための低コストのソリューションです。

USB コネクタは 1 方向 1 種類のケーブルだけに適合するように設計されています。USB が設計された主な目的は、デバイスごとに異なる何種類ものコネクタを減らすことです。USB の設計により、システムの背面パネルの混雑を軽減できます。

デバイスは、外部 USB ハブ上の USB ポートか、コンピュータ本体に設置されたルートハブ上の USB ポートのいずれかに接続されます。ハブには複数のポートがあるため、1 つのハブからデバイスツリーの複数の枝が伸びることがあります。

ほかにも、USB デバイスを使用すると次のような利点があります。

- USB デバイスはホットプラグ可能です。詳細については、424 ページの「USB デバイスのホットプラグ」を参照してください。
- Solaris 9 環境では、USB 1.1 デバイスをサポートします。
- Solaris 環境で 1 ホストコントローラあたり、最大 126 台のデバイスをサポートします。
- 最大 12M ビット/秒のデータ転送速度をサポートします。
- ロースピード (1.5 M ビット/秒) からフルスピード (12 M ビット/秒) までをサポートします。
- Solaris Ready USB PCI コントローラを利用できます。詳細については、<http://www.sun.com/io> を参照してください。
- 低コストの外部ハブを追加するだけで簡単にバスを拡張することが可能です。また、ハブとハブを接続してツリートポロジを構築できます。

Sun のプラットフォームで提供される USB デバイスサポートは、次のとおりです。

- USB 1.1 をサポートする OHCI ホストコントローラを搭載した次の SPARC システムは、ロースピードおよびフルスピードのデバイスをサポートします。
  - Solaris 8 または Solaris 9 で動作する Sun Blade™ システム
  - Solaris 9 で動作する Netra™ X1/T1 およびいくつかの Sun Fire™ システム
- 次の SPARC システムは、ロースピードとフルスピードの USB 1.1 デバイスをサポートします。

Solaris 8 または Solaris 9 で動作する、上記を含む任意の PCI ベースの sun4u システム

- Solaris 8 または Solaris 9 の x86 プラットフォーム版で動作する、OHCI コントローラと UHCI コントローラを搭載した x86 システムは、USB 1.1 サポートを提供します。
- Solaris 8 または Solaris 9 の x86 プラットフォーム版で動作する、OHCI ホストコントローラを搭載した x86 システムは、USB 1.1 サポートを提供します。

次の表に、Solaris 環境でサポートされる USB デバイスを示します。

| USB デバイス               | USB デバイスがサポートされるシステム |
|------------------------|----------------------|
| オーディオデバイス上の HID コントロール | SPARC および x86 システム   |



| USB デバイス   | USB デバイスがサポートされるシステム                                                                                  |
|------------|-------------------------------------------------------------------------------------------------------|
| ハブ         | SPARC および x86 システム                                                                                    |
| キーボードとマウス  | SPARC および x86 システム                                                                                    |
| 大容量ストレージ   | SPARC および x86 システム<br>サポートされる構成では、キーボードとマウスは1つずつしか含めることができない。これらのデバイスは、オンボードの USB ホストコントローラに接続する必要がある |
| プリンタ       | SPARC および x86 システム                                                                                    |
| スピーカおよびマイク | SPARC および x86 システム                                                                                    |

## よく使用される USB 関連の略語

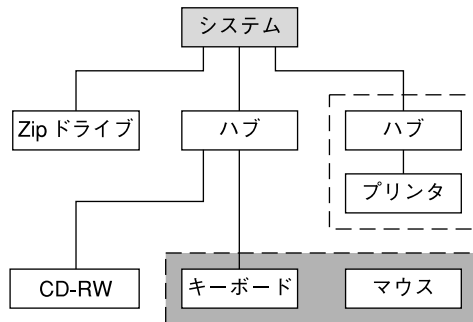
次の表に、Solaris 環境で使用される USB の略語について説明します。USB の構成要素と略語についての詳細は <http://www.usb.org> を参照してください。

| 略語    | 定義                               |
|-------|----------------------------------|
| USB   | Universal Serial Bus             |
| USBA  | USB アーキテクチャ (Solaris)            |
| USBAI | USBA クライアントドライバインタフェース (Solaris) |
| HCD   | USB ホストコントローラドライバ                |
| EHCI  | 拡張オープンコントローラインタフェース              |
| OHCI  | オープンホストコントローラインタフェース             |
| UHCI  | ユニバーサルホストコントローラインタフェース           |

## USB バスの説明

USB 仕様は、ライセンス料を払わずに入手することができます。USB 仕様は、バスとコネクタの電気的および機械的なインタフェースを定義します。

USB が採用するトポロジでは、ハブが USB デバイスに接続点を提供します。ホストコントローラには、システム内のすべての USB ポートの起点となるルートハブが含まれます。ハブの詳細については、421 ページの「USB ホストコントローラとルートハブ」を参照してください。



- USB ホストコントローラーおよびルートハブ
- 合成デバイス
- 複合デバイス

図 29-1 USB 物理デバイスの階層

図 29-1 は、有効な USB ポートが 3 つ搭載されたシステムを示しています。1 番目の USB ポートは Zip ドライブに接続されています。2 番目の USB ポートは外部ハブに接続されており、このハブには CD-RW デバイスと、キーボードとマウスの複合デバイスが接続されています。この「複合デバイス」としてのキーボードには USB コントローラーが組み込まれており、このコントローラーによって、キーボードとキーボードに接続されたマウスの両方が制御されます。キーボードとマウスは、同じ USB コントローラーによって制御されるため、同一の USB バスアドレスを共有します。

図 29-1 は、ハブとプリンタの「合成デバイス」の例も示しています。このハブは外部ハブで、プリンタと同じケースに入っています。プリンタはこのハブに固定接続されます。ただし、このハブとプリンタは、それぞれ異なる USB バスアドレスを持ちます。また、図 29-1 は「複合デバイス」の例も示しています。

次の表に、図 29-1 に示したデバイスの一部について、デバイスツリーパス名を示します。

|            |                                              |
|------------|----------------------------------------------|
| Zip ドライブ   | /pci@1f,4000/usb@5/storage@1                 |
| キーボード      | /pci@1f,4000/usb@5/hub@2/device@1/keyboard@0 |
| マウス        | /pci@1f,4000/usb@5/hub@2/device@1/mouse@1    |
| CD-RW デバイス | /pci@1f,4000/usb@5/hub@2/storage@3           |
| プリンタ       | /pci@1f,4000/usb@5/hub@3/printer@1           |

## USB デバイスとドライバ

USB デバイスは複数のデバイスクラスに分類されます。各デバイスクラスは対応するドライバを持っています。1つのクラス内のデバイスは同じデバイスドライバで管理されます。ただし USB 仕様では、特定のクラスに属さない、ベンダー固有のデバイスも許可しています。類似した属性とサービスを持つデバイスはグループ化されます。

Human Interface Device (HID) クラスには、ユーザーが制御するデバイス (キーボード、マウス、ジョイスティックなど) が含まれます。Communication Device クラスには、電話に接続するデバイス (モデムや ISDN インタフェースなど) が含まれます。その他にも、Audio Device、Monitor Device、Printer Device、Storage Device などのデバイスクラスがあります。各 USB デバイスはデバイスのクラスを表す記述子を持っています。デバイスクラスは、そのメンバーが構成とデータ転送についてどのように動作するかを指定します。クラス情報の詳細については、<http://www.usb.org> を参照してください。

## Solaris USB アーキテクチャ (USBA)

USB デバイスは、2つのレベルのデバイスツリーノードとして表現されます。デバイスノードは、USB デバイス全体を表します。1つまたは複数の子インタフェースノードはデバイス上にある個々のUSB インタフェースを表します。特殊なケースとして、デバイスノードとインタフェースノードが1つのノードに「結合される」場合もあります。

ドライバのバインドは互換性のある名前属性の使用によって実現されます。詳細については、『*IEEE 1275 USB binding* (英語版)』の 3.2.2.1 項と『*Writing Device Drivers* (英語版)』を参照してください。ドライバは、デバイス全体にバインドしてすべてのインタフェースを制御することも、1つのインタフェースだけにバインドすることも可能です。デバイス全体にバインドするドライバがベンダーにもクラスにも存在しない場合、汎用 USB マルチインタフェースドライバがデバイスレベルのノードにバインドされます。IEEE 1275 バインド仕様の 3.2.2.1 項で定義されているように、このドライバは互換名プロパティを使用して、各インタフェースに対してドライバのバインドを試みます。

Solaris USB アーキテクチャ (USBA) は、USB 1.1 の仕様に加え、Solaris ドライバ条件にも準拠しています。USBA モデルは Sun Common SCSI Architecture (SCSA) に似ています。USBA とは、汎用 USB トランスポート層という概念をクライアントドライバに提供する薄い層のことです。



図 29-2 Solaris USB アーキテクチャ (USBA)

---

## Solaris 環境における USB について

次の節では、Solaris 環境における USB について知っておく必要のある情報を説明します。

### USB キーボードとマウス

サポートされているのは、Sun USB キーボードとマウスデバイスだけです。また、複数の USB キーボードとマウスデバイスを含むシステム構成は場合によっては機能しますが、Solaris 環境ではサポートされていません。詳細については、次の説明を参照してください。

- USB キーボードおよびマウスは、バス上の任意の場所に接続して、コンソールキーボードおよびマウスとして構成することができます。キーボードとマウスがルートハブにない場合、システムのブートが遅くなります。
- リブート中または ok プロンプトの出ている間は、コンソールキーボードおよびマウスを移動しないでください。システムのリブート後であればいつでもコンソールキーボードおよびマウスを別のハブに移動することができます。キーボードおよびマウスは、差し込んだ後は再び完全に機能します。
- SPARC のみ - USB キーボードの電源キーと Sun タイプ 5 キーボードの電源キーの動作は異なります。USB キーボードでは、「SUSPEND/SHUTDOWN」キーを使用してシステムを中断またはシャットダウンすることができますが、そのキーを使用してシステムの電源を入れることはできません。
- Sun 社製以外の USB キーボードでは、キーパッドの左側にある機能は使用できません。
- 複数のキーボードはサポートされません。

- 複数のキーボードは認識され、使用できますが、コンソールキーボードとしては認識されません。
- ブート時に最初に認識されたキーボードがコンソールキーボードとなります。このため、複数のキーボードが接続されていると、ブート時に混乱の原因となります。
- コンソールキーボードを取り外しても、次に利用可能な USB キーボードはコンソールキーボードにはなりません。次にホットプラグされるキーボードがコンソールキーボードになります。
- 複数のマウスはサポートされません。
  - 複数のマウスは認識され、使用できますが、コンソールマウスとしては認識されません。
  - ブート時に最初に認識されたマウスがコンソールマウスとなります。このため、複数のマウスが差し込まれていると、ブート時に混乱の原因となります。
  - コンソールマウスを取り外しても、次に利用可能な USB マウスはコンソールマウスにはなりません。次にホットプラグされるマウスがコンソールマウスになります。
- Sun 社製以外のキーボードを PS/2 マウスと合成して使用する場合、このキーボードがブート時に最初に認識されると、PS/2 マウスが差し込まれていなくても、このキーボードとマウスがコンソールキーボードとマウスになります。このため、別の USB マウスがシステムに差し込まれていても、コンソールマウスとして構成されないので機能しません。
- 2 ボタンと 3 ボタンのマウスだけがサポートされます。ホイール付きマウスは動作しません。3 ボタンよりも多いマウスは 3 ボタンのマウスのように動作します。

## USB ホストコントローラとルートハブ

USB ハブは次のことを行います。

- ポートにおけるデバイスの取り付けと取り外しの監視
- ポートにおける個々のデバイスの電源管理
- ポートへの電源の制御

USB ホストコントローラは「ルートハブ」という埋め込みハブを持っています。システムの背面パネルに見えるポートはルートハブのポートです。USB ホストコントローラは次のことを行います。

- USB バスの管理。個々のデバイスはバスの調整はできません。
- デバイスによって決定されるポーリング間隔による、デバイスのポーリング。ポーリング間隔 (時間) を考慮してデバイスに十分なバッファがあることを前提とします。
- USB ホストコントローラとそれに接続されているデバイス間でのデータの送信。ピアツーピア通信はサポートされません。

## USB ハブデバイス

- SPARC システムと x86 システムのどちらにおいても、ハブを 4 段を超えて多段接続しないでください。SPARC システムでは、OpenBoot™ PROM (OBP) は 4 段を超えるデバイスを正確に認識できません。
- バス電源供給方式のハブ同士をカスケード接続しないでください。バス電源供給方式のハブは独自の電源を持っていません。
- 大量の電源を必要とするデバイスをバス電源供給方式のハブに接続しないでください。これらのデバイスがバス電源供給方式のハブ上で正しく機能しなかったり、他のデバイス用の電源がなくなったりする可能性があります。このようなデバイスとして、USB フロッピーディスクデバイスなどがあります。

## SPARC: USB 電源管理

SPARC システムでは、USB デバイスの保存停止・復元再開機能が完全にサポートされます。ただし、稼働中のデバイスを保存停止したり、システムの保存停止で電源がオフになっているときにデバイスを取り外すことは決してしないでください。

SPARC システムで電源管理を有効にしている場合、USB のフレームワークはすべてのデバイスの電源管理を最大限に試みます。USB デバイスの電源管理により、ハブドライバはデバイスが接続されているポートの中断も行います。「リモートウェイクアップ」をサポートするデバイスは、そのデバイスが利用可能な状態になるように、そのデバイスのバス上にあるすべてのデバイス呼び起こすようシステムに通知できます。アプリケーションがデバイスに入出力を送信した場合も、ホストシステムはデバイス呼び起こすことができます。

リモートウェイクアップ機能がサポートされている場合、すべての HID (キーボード、マウス、スピーカ、マイクなど)、ハブ、およびストレージデバイスは、デフォルトで電源管理されます。USB プリンタが電源管理されるのは、2 つの印刷ジョブ間だけです。

電源消費を減らすために電源管理を行っている場合は、まず USB 末端デバイスの電源が切断されます。また、ポートに接続されているすべてのデバイスの電源が切断されると、しばらくしてからハブの電源が切断されます。最も効率的に電源管理をするためには、あまり多くのハブをカスケード接続しないでください。

## USB ケーブルに関するガイドライン

USB ケーブルを接続する際には、以下のガイドラインに従ってください。

- USB デバイスを接続するときは、必ず USB 1.0 に準拠したフルレート (12M ビット/秒) の 20/28 AWG ケーブルを使用してください。また、バス電源供給方式のハブは低速のデバイスだけに使用してください。USB デバイスを接続するときは、必ずフルレート (12M ビット/秒) の 20/28 AWG ケーブルを使用してください。

- サポートされている最長ケーブル長は5 m です。
- 延長ケーブルを使用しないでください。ケーブルを延長するには、最良の結果が得られるよう、自己電源供給方式のハブを使ってください。

詳細については、<http://www.usb.org/channel/training/warning> を参照してください。

## USB 大容量ストレージデバイスの使用 (作業マップ)

| 作業                         | 説明                                                                                    | 参照先                                                                                                                             |
|----------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 1. USB 大容量ストレージデバイスを追加する   | <p>void を使用して USB 大容量ストレージデバイスを取り付ける</p> <p>void を使用しないで USB 大容量ストレージデバイスを取り付ける</p>   | <p>425 ページの「void を使用して USB 大容量ストレージデバイスを追加する方法」</p> <p>425 ページの「void を使用しないで USB 大容量ストレージデバイスを追加する方法」</p>                       |
| 2. USB 大容量ストレージデバイスを取り外す   | <p>void を使用して USB 大容量ストレージデバイスを取り外す</p> <p>void を使用しないで USB 大容量ストレージデバイスを取り外す</p>     | <p>426 ページの「void を使用して USB 大容量ストレージデバイスを取り外す方法」</p> <p>426 ページの「void を使用しないで USB 大容量ストレージデバイスを取り外す方法」</p>                       |
| 3. USB 大容量ストレージデバイスをマウントする | <p>void を使用して USB 大容量ストレージデバイスをマウントする</p> <p>void を使用しないで USB 大容量ストレージデバイスをマウントする</p> | <p>428 ページの「void を使用して USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」</p> <p>429 ページの「void を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」</p> |
| 4. USB カメラを追加する            | USB カメラを追加して、デジタル画像にアクセスする                                                            | 430 ページの「USB カメラを追加する方法」                                                                                                        |

---

## USB 大容量ストレージデバイスの使用

Solaris 9 から、USB の CD-RW、ハードディスク、DVD、デジタルカメラ、Zip、Peerless、SmartMedia、CompactFlash、ORB、および USB フロッピーディスクなどのリムーバブル大容量ストレージデバイスがサポートされるようになりました。

Solaris 環境でサポートされる USB デバイスについては、[http://www.sun.com/io\\_technologies/usb.html](http://www.sun.com/io_technologies/usb.html) を参照してください。

これらのデバイスは、ボリューム管理を実行している場合でも実行していない場合でも管理することができます。ボリューム管理を実行している場合のデバイス管理については、`vold(1M)` のマニュアルページを参照してください。

## 非準拠 USB 大容量ストレージデバイスの使用

デバイスの識別が間違っていたり、USB 大容量ストレージクラスと互換性がないとされているドライバでも、USB 大容量ストレージドライバをサポートする場合があります。 `scsa2usb.conf` ファイルには、大容量デバイスに対応するかどうかを示す、ベンダー ID、製品 ID、およびバージョンを一覧表示した属性オーバーライドリストが含まれています。またデフォルトのデバイス属性をオーバーライドするフィールドも含まれています。このリストのエントリにはデフォルトでコメントが表示されています。特定のデバイスのサポートを有効にするために、コメントをコピーしたり削除したりすることもできます。

Solaris で動作するシステムに USB 大容量ストレージデバイスを接続したが、システムでそのデバイスを使用できない場合、`/kernel/drv/scsa2usb.conf` ファイルでこのデバイスに一致するコメントエントリがあるかどうか確認できます。`scsa2usb.conf` ファイルの指示に従って、オーバーライド情報を使用して、特定のデバイスをサポートできるかどうか確認してください。推奨されている USB 大容量ストレージデバイスの一覧については、<http://www.sun.com/io> を参照してください。

詳細については、`scsa2usb(7D)` のマニュアルページを参照してください。

---

## USB デバイスのホットプラグ

デバイスのホットプラグとは、オペレーティングシステムをシャットダウンすることなくあるいはシステムの電源を切ることなく、デバイスを追加したり取り外したりすることを指します。USB デバイスはすべてホットプラグ可能です。



USB デバイスは、ホットプラグするとすぐにシステムのデバイス階層に表示されます (prtconf コマンドで確認可能)。また、デバイスが使用中でない限り、USB デバイスを取り外すとシステムのデバイス階層から消えます。

使用中の USB デバイスを取り外した場合、ホットプラグの動作は若干異なります。使用中の USB デバイスを取り外した場合、デバイスノードは残り、このデバイスを制御しているドライバはデバイス上のすべての動作を停止します。それ以降、このデバイスに発行される新しい入出力動作はエラーで戻されます。

このような場合、システムは元のデバイスを接続するようにユーザーにプロンプトを表示します。デバイスが使用できない場合は、アプリケーションを停止してください。数秒後に、ポートが再び使用できるようになります。

---

注 - 動作中の、つまり開いているデバイスを削除すると、データの整合性が損なわれる可能性があります。デバイスを取り外す前には、必ずデバイスを閉じるようにしてください。ただし、コンソールキーボードとマウスは例外で、動作中でも移動することができます。

---

## ▼ vold を使用して USB 大容量ストレージデバイスを追加する方法

次の手順は、vold を使用して USB デバイスを追加する方法を示します。

ボリューム管理デバイス名の詳細については、第 17 章を参照してください。

1. USB 大容量ストレージデバイスを追加します。
2. vold を使用して新しいデバイスがあるかどうか調べます。

```
touch /etc/vold.conf
```

3. vold を再起動します。

```
pkill -HUP vold
```

4. デバイスが取り付けられていることを確認します。

```
$ ls device-alias
```

## ▼ vold を使用しないで USB 大容量ストレージデバイスを追加する方法

次の手順は、vold を使用しないで USB デバイスを追加する方法を示します。

1. USB デバイスを USB ポートに追加します。

2. **USB** デバイスが追加されたことを確認します。

非 USB ストレージデバイスのデバイスリンクの間にあるかもしれない USB ディスクのデバイスリンクを、次のようにして確認します。

```
$ cd /dev/rdisk
$ ls -l c*0 | grep usb
lrwxrwxrwx 1 root root 67 Apr 30 15:12 c1t0d0s0 ->
 ../../devices/pci@1f,0/pci@5/pci@0/usb@8,2/storage@1/disk@0,0:a,raw
```

## ▼ vold を使用して USB 大容量ストレージデバイスを取り外す方法

次の手順では、vold を使用して USB デバイスを取り外す例として、Zip ドライブを使用しています。

1. デバイスをマウント解除します。

```
$ volrmount -e zip0
```

2. そのデバイスを使用しているアプリケーションを実行中の場合は、そのアプリケーションを停止します。

3. デバイスをマウント解除します。

```
$ volrmount -e zip0
```

4. デバイスを取り出します。

```
$ eject zip0
```

5. スーパーユーザーになり、vold を停止します。

```
/etc/init.d/volmgt stop
```

6. **USB** 大容量ストレージデバイスを取り外します。

7. vold を開始します。

```
/etc/init.d/volmgt start
```

## ▼ vold を使用しないで USB 大容量ストレージデバイスを取り外す方法

次の手順は、vold を使用しないで USB デバイスを取り外す方法を示します。

1. スーパーユーザーになります。

2. そのデバイスを使用しているアプリケーションを実行中の場合は、そのアプリケーションを停止します。

3. USB デバイスを取り外します。
  - a. デバイスをマウント解除します。

```
umount /mount-point
```

- b. デバイスを取り外します。

## vold を使用してまたは使用しないで USB 大容量ストレージデバイスをマウントする

Solaris 共通デスクトップ環境 (CDE) が実行されている場合は、USB リムーバブル大容量ストレージデバイスは、CDE ファイルマネージャのコンポーネントであるリムーバブルメディア・マネージャによって管理されます。CDE ファイルマネージャの詳細については、`dtfile(1)` のマニュアルページを参照してください。

---

注 - この節に記載されているマニュアルページを表示するには、`MANPATH` 変数に `/usr/dt/man` ディレクトリを含める必要があります。これらのコマンドを使用するには、パスに `/usr/dt/bin` ディレクトリを含め、さらに CDE を実行している必要があります。また、これらのコマンドをリモートで使用する場合は、`DISPLAY` 変数を設定する必要があります。

---

次の表に、リムーバブルメディア・マネージャが CDE 環境からストレージデバイスを管理する際に使用するコマンドを示します。

| コマンド                         | マニュアルページ                        | 作業                     |
|------------------------------|---------------------------------|------------------------|
| <code>sdtmedia_format</code> | <code>sdtmedia_format(1)</code> | デバイスのフォーマットおよびラベル付けを行う |
| <code>sdtmedia_prop</code>   | <code>sdtmedia_prop(1)</code>   | デバイスのプロパティを表示する        |
| <code>sdtmedia_prot</code>   | <code>sdtmedia_prot(1)</code>   | デバイスのアクセス権を変更する        |
| <code>sdtmedia_slice</code>  | <code>sdtmedia_slice(1)</code>  | デバイス上のスライスを作成または変更する   |

USB デバイスのフォーマットが終了すると、通常は `/rmdisk/label` ディレクトリの下にマウントされます。リムーバブルストレージデバイスの構成方法については、`rmmount.conf(4)` または `vold.conf(4)` のマニュアルページを参照してください。

デバイスノードは `/vol/dev` ディレクトリの下に作成されます。詳細については、`scsa2usb(7D)` のマニュアルページを参照してください。

USB 大容量ストレージデバイスは、ボリュームマネージャ (vold) を使用しないで管理することもできます。ボリュームマネージャを使用しないようにする方法は、次の 2 とおりあります。

- 次のコマンドを実行して、vold を停止します。

```
/etc/init.d/volmgt stop
```

- vold を実行したままで、USB 大容量ストレージデバイスを vold に登録しないようにします。以下に示すように /etc/vold.conf ファイル内の次の行で # マークを挿入してコメント扱いにし、USB 大容量ストレージデバイスのボリュームマネージャの登録を削除します。

```
use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

上記のようにコメントにした後、vold を再起動します。

```
/etc/init.d/volmgt start
```



---

注意 - この行をコメントにすると、そのシステム内に他の SCSI、ATAPI Zip、または Jaz リムーバブルデバイスがある場合、それらのデバイスの vold の登録も無効になります。

---

詳細については、vold.conf(4) のマニュアルページを参照してください。

次の手順は、vold を使用しないで USB 大容量ストレージデバイスを管理する方法を示しています。デバイスノードは、文字型デバイスについては /dev/rdisk ディレクトリ、ブロック型デバイスについては /dev/dsk ディレクトリの下に作成されます。詳細については、scsa2usb(7D) のマニュアルページを参照してください。

## vold を使用して USB 大容量ストレージデバイスをマウントまたはマウント解除する方法

1. すべてのリムーバブル大容量ストレージデバイス (**USB 大容量ストレージデバイスを含む**) について、デバイスの別名を表示します。

```
$ eject -n
.
.
.
rmdisk0 -> /vol/dev/rdsk/c4t0d0/clik40 (Generic USB storage)
cdrom0 -> /vol/dev/rdsk/c0t6d0/audio_cd (Generic CD device)
zip1 -> /vol/dev/rdsk/c2t0d0/fat32 (USB Zip device)
zip0 -> /vol/dev/rdsk/c1t0d0/zip100 (USB Zip device)
jaz0 -> /vol/dev/rdsk/c3t0d0/jaz1gb (USB Jaz device)
```

2. 上記のようにリストされたデバイス別名を使用して、**USB 大容量ストレージデバイス**をマウントします。

```
$ volrmount -i device-alias
```

次の例では、USB Zip ドライブ (/rmdisk/zip0) をマウントします。

```
$ volrmount -i zip0
```

3. **USB** 大容量ストレージデバイスをマウント解除します。

```
$ volrmount -e device-alias
```

次の例では、USB Zip ドライブ (/rmdisk/zip0) をマウント解除します。

```
$ volrmount -e zip0
```

4. **USB** デバイスを、汎用の **USB** ドライブから取り出します。

```
$ eject device-alias
```

たとえば、次のようになります。

```
$ eject rmdisk0
```

---

注 - デバイスのマウントが解除されていない場合、eject コマンドは、そのデバイスのマウント解除も行います。また、そのデバイスにアクセスするアプリケーションが実行されている場合は、そのアプリケーションを終了させます。

---

## vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法

1. スーパーユーザーになります。
2. **USB** 大容量ストレージデバイスをマウントします。

```
mount -F fs-type /dev/dsk/cntndnsn /mount-point
```

このコマンドは、デバイスが読み取り専用の場合には失敗することがあります。CD-ROM デバイスには、次のコマンドを使用します。

```
mount -F fs-type -o ro /dev/dsk/cntndnsn /mount-point
```

たとえば、次のようになります。

```
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
```

3. **USB** 大容量ストレージデバイスをマウント解除します。

```
umount /mount-point
```

4. デバイスを取り出します。

```
eject /dev/[r]dsk/cntndnsn
```

## ▼ USB カメラを追加する方法

USB カメラを追加するには、次の手順を実行します。

1. スーパーユーザーになります。

2. **USB** カメラを差し込み、電源をオンにします。

システムによって、カメラ用の論理デバイスが作成されます。カメラが差し込まれると、`/var/adm/messages` ファイルにメッセージが出力され、デバイスの接続が確認されます。カメラは、システムのストレージデバイスとみなされます。

3. `/var/adm/messages` ファイルの出力を確認します。

出力表示を確認すると、どの論理デバイスが作成されたかを確認でき、そのデバイスを使用してイメージにアクセスできます。出力表示は次のようになります。

```
more /var/adm/messages
Jul 15 09:53:35 buffy usba: [ID 349649 kern.info] OLYMPUS, C-3040ZOOM,
000153719068
Jul 15 09:53:35 buffy genunix: [ID 936769 kern.info] scsa2usb1 is
/pci@0,0/pci925,1234@7,2/storage@2
Jul 15 09:53:36 buffy scsi: [ID 193665 kern.info] sd3 at scsa2usb1:
target 0 lun 0
```

4. **USB** カメラファイルシステムをマウントします。

ほとんどの場合、カメラのファイルシステムは PCFS ファイルシステムです。作成されたデバイス上にファイルシステムをマウントするために、ディスクを表すスライスを指定する必要があります。スライスは通常、SPARC システムでは `s0`、x86 システムでは `p0` です。

たとえば、x86 システムにファイルシステムをマウントするには、次のコマンドを実行します。

```
mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

SPARC システムにファイルシステムをマウントするには、次のコマンドを実行します。

```
mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
```

ファイルシステムのマウントについては、第 40 章を参照してください。

異なる PCFS ファイルシステムのマウントの詳細については、`mount_pcfs(1M)` のマニュアルページを参照してください。

5. イメージが使用可能であることを確認します。

たとえば、次のようになります。

```
ls /mnt/DCIM/100OLYMP/
P7220001.JPG* P7220003.JPG* P7220005.JPG*
P7220002.JPG* P7220004.JPG* P7220006.JPG*
```

6. **USB** カメラが作成したイメージファイルを表示します。

```
/usr/dt/bin/sdtimage P7220001.JPG &
```

- カメラを切り離す前に、ファイルシステムをマウント解除します。  
たとえば、次のようになります。

```
umount /mnt
```

- カメラの電源をオフにし、切り離します。

---

## USB オーディオデバイスの使用 (作業マップ)

| 作業                               | 説明                                                                                                                   | 参照先                                         |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| 1. USB オーディオデバイスを追加する            | USB マイクおよびスピーカを追加する                                                                                                  | 433 ページの「USB オーディオデバイスを追加する方法」              |
| 2. システムの主オーディオデバイスを特定する          | システムの主オーディオデバイスを特定する                                                                                                 | 433 ページの「システムの主オーディオデバイスを識別する方法」            |
| 3. 主 USB オーディオデバイスを変更する          | USB オーディオデバイスを取り外したり変更したりする場合、特定の1つのオーディオデバイスを主オーディオデバイスにすることがある                                                     | 435 ページの「主 USB オーディオデバイスを変更する方法」            |
| 4. 使用していない USB オーディオデバイスリンクを取り外す | システムの電源がオフのときに USB オーディオデバイスを取り外すと、 <code>/dev/audio</code> デバイスが、存在しない <code>/dev/sound/*</code> デバイスを指したままになることがある | 436 ページの「使用されていない USB オーディオデバイスのリンクを削除する方法」 |
| 5. USB オーディオデバイスに関する問題の障害追跡      | USB スピーカの電源を再投入する必要がある                                                                                               | 437 ページの「USB スピーカ障害に対処する」                   |

---

## USB オーディオデバイスの使用

この Solaris リリースでは、2つの連携するドライバ、`usb_ac` および `usb_as` の実装によって、USB オーディオサポートを提供しています。オーディオコントロールドライバである `usb_ac` は USB A (Solaris USB Architecture) 準拠のクライアントドライバ

で、ユーザーアプリケーションのインタフェースを制御します。オーディオストリーミングドライバである `usb_as` は、再生中および録音中にオーディオデータメッセージを処理し、サンプル周波数と精度を設定し、`usb_ac` ドライバからのエンコーディング要求を処理します。

どちらのドライバも、USB オーディオクラス 1.0 仕様に準拠しています。

Solaris では、再生専用または録音専用の外部 USB オーディオデバイスをサポートします。オンボードの USB オーディオデバイスはサポートされません。サポートされているオーディオデータ形式については、`usb_ac(7D)` のマニュアルページを参照してください。

- オーディオデバイスの音量をソフトウェアが制御する場合、音量ボタンを管理するために `usb_ah` という STREAMS モジュールが HID ドライバの先頭に置かれます。
- USB オーディオデバイスは、USB コネクタを備えた SPARC Ultra および x86 プラットフォームでサポートされます。
- ホットプラグ可能な USB オーディオデバイスがサポートされます。
- USB オーディオデバイスは、8000 ~ 48000 Hz の連続サンプリングレートをサポートしている必要があります。また、Solaris 8 10/01、Solaris 8 2/02、または Solaris 9 リリース上で再生または録音するには、48000 Hz のサンプリングレートをサポートしている必要があります。

主オーディオデバイスは、`/dev/audio` です。次のコマンドを使用して、`/dev/audio` が USB オーディオを指しているかを確認できます。

```
% mixerctl
Device /dev/audioc1:
 Name = USB Audio
 Version = 1.0
 Config = external
```

```
Audio mixer for /dev/audioc1 is enabled
```

使用する USB オーディオデバイスを接続した後、`audioplay` コマンドおよび `audiorecord` コマンドを使用して、次のファイルにアクセスします。

```
/dev/sound/N
```

AUDIODEV 環境変数を設定するか、`audioplay` コマンドおよび `audiorecord` コマンドで `-d` オプションを指定して、特定のオーディオデバイスを選択できます。ただし、`/dev/audio` をオーディオファイルとしてハードコードしているアプリケーションでは AUDIODEV は動作しません。

USB オーディオデバイスを差し込むと、`/dev/audio` が使用中でない限り、自動的にそれが主オーディオデバイス `/dev/audio` になります。オンボードのオーディオから USB オーディオへ、および USB オーディオからオンボードのオーディオへ `/dev/audio` を変更する方法については、435 ページの「主 USB オーディオデバイスを変更する方法」 および `usb_ac(7D)` のマニュアルページを参照してください。



## 複数の USB オーディオデバイスのホットプラグ

USB オーディオデバイスがシステムに差し込まれると、それが主オーディオデバイス /dev/audio になります。そのデバイスは、システムをリブートした後でも、主オーディオデバイスのままです。USB オーディオデバイスが追加で差し込まれた場合、最後に差し込まれたデバイスが主オーディオデバイスになります。

USB オーディオデバイスに関する問題の障害追跡については、usb\_ac(7D) のマニュアルページを参照してください。

### ▼ USB オーディオデバイスを追加する方法

USB オーディオデバイスを追加するには、次の手順を実行します。

1. **USB** スピーカとマイクを差し込みます。

主オーディオデバイス /dev/audio は通常、オンボードのオーディオを指します。USB オーディオデバイスを接続した後、/dev/audio は /dev/sound ディレクトリで認識される USB オーディオデバイスを指します。

2. 作成されたオーディオデバイスファイルを確認します。

```
% ls /dev/sound
0 Oct1 1 1ctl 2 2ctl
```

3. 左右の **USB** スピーカをテストします。

```
% cd /usr/share/audio/samples/au
% audioplay -d /dev/sound/1 -b 100 spacemusic.au
% audioplay -d /dev/sound/1 -b -100 spacemusic.au
```

4. **USB** マイクをテストします。

```
% cd $HOME/au
% audiorecord -d /dev/sound/2 -p mic -t 30 test.au
```

### ▼ システムの主オーディオデバイスを識別する方法

この手順は、すでに USB オーディオデバイスが接続されていることを前提としています。

1. 現在のオーディオデバイスのリンク状態を識別します。

たとえば、次のようになります。

```
% ls -lt /dev/audio*
lrwxrwxrwx 1 root root 7 Jul 23 15:41 /dev/audio -> sound/0
lrwxrwxrwx 1 root root 10 Jul 23 15:41 /dev/audioctl ->
sound/0ctl
% ls -lt /dev/sound/*
```

```
lrwxrwxrwx 1 root other 66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx 1 root other 69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioc1
%
```

主オーディオデバイス /dev/audio は、現在オンボードのオーディオ、  
/dev/sound/0 を指しています。

2. (省略可能) 新しい **USB** オーディオデバイスを追加します。

3. システムの新しいオーディオリンクを調べます。

たとえば、次のようになります。

```
% ls -lt /dev/audio*
lrwxrwxrwx 1 root root 7 Jul 23 15:46 /dev/audio -> sound/1
lrwxrwxrwx 1 root root 10 Jul 23 15:46 /dev/audioc1 ->
sound/1ctl
% ls -lt /dev/sound/*
lrwxrwxrwx 1 root root 74 Jul 23 15:46 /dev/sound/1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,...
lrwxrwxrwx 1 root root 77 Jul 23 15:46 /dev/sound/1ctl ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,...
lrwxrwxrwx 1 root other 66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx 1 root other 69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioc1
%
```

主オーディオデバイス /dev/audio が、新しく差し込まれた USB オーディオデバイスの /dev/sound/1 を指していることがわかります。

ここで USB オーディオデバイスを取り外しても、主オーディオデバイス /dev/audio はオンボードのオーディオには戻りません。主オーディオデバイスをシステムのオンボードのオーディオに戻す方法については、次の手順を参照してください。

prtconf コマンドを使用して USB デバイス情報を参照して、システム上の USB オーディオデバイスを調べることもできます。

```
% prtconf
.
.
.
usb, instance #0
 hub, instance #0
 mouse, instance #0
 keyboard, instance #1
 device, instance #0
 sound-control, instance #0
 sound, instance #0
 input, instance #0
.
.
.
```

## ▼ 主 USB オーディオデバイスを変更する方法

USB オーディオデバイスを取り外すか変更して、特定のオーディオデバイス 1 つを主オーディオデバイスにしたい場合は、次の手順を実行してください。この手順では、例として主オーディオデバイスをオンボードのオーディオデバイスに変更します。

1. スーパーユーザーになります。
2. すべてのオーディオアプリケーションを閉じます。
3. オーディオドライバおよび USB ドライバが読み込まれているかを調べます。

```
modinfo | grep -i audio
124 780e6a69 bb6e - 1 audiosup (Audio Device Support 1.12)
modinfo | grep -i usb
48 13dba67 18636 199 1 ohci (USB OpenHCI Driver 1.31)
49 78020000 1dece - 1 usba (USBA: USB Architecture 1.37)
50 12e5f1f 35f 195 1 hubd (USB Hub Driver 1.4)
51 13ef53d 5e26 194 1 hid (USB HID Client Driver 1.16)
54 13f67f2 1b42 10 1 usbms (USB mouse streams 1.6)
56 127bbf0 2c74 11 1 usbkbm (USB keyboard streams 1.17)
#
```

4. オンボードのオーディオドライバを読み込んで再接続します。

```
devfsadm -i audiocs
```

オンボードのオーディオドライバは、Sunblade 1000 では audiocs、Sunblade 100 では audiotcs です。

5. 主オーディオデバイスのリンクがオンボードのオーディオを指していることを確認します。

```
ls -lt /dev/audio*
lrwxrwxrwx 1 root other 7 Jul 23 15:49 /dev/audio -> sound/0
lrwxrwxrwx 1 root other 10 Jul 23 15:49 /dev/audioctl ->
sound/0ctl
ls -lt /dev/sound/*
lrwxrwxrwx 1 root other 66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx 1 root other 69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioc1
#
```

6. オンボードのオーディオが主オーディオデバイスになっていることを確認します。

```
% audioplay /usr/demo/SOUND/sounds/bark.au
```

audioplay コマンドは、デフォルトでオンボードのオーディオデバイスを使用します。

7. (省略可能) 別の USB オーディオデバイスを差し込む前に、読み込み解除できるオーディオドライバをすべて読み込み解除します。

- a. すべてのオーディオアプリケーションを閉じます。

- b. オーディオドライバ情報を表示して、オーディオドライバが読み込まれていないことを確認します。

```
modinfo | grep -i audio
60 78048000 bb6e - 1 audiosup (Audio Device Support 1.12)
61 78152000 39a97 - 1 mixer (Audio Mixer 1.49)
62 78118000 bf9f - 1 amsrc1 (Audio Sample Rate Conv. #1 1.3)
128 7805e000 14968 54 1 audiocs (CS4231 mixer audio driver 1.21)
#
```

- c. オーディオドライバを読み込み解除します。

```
modunload -i 0
modinfo | grep -i audio
60 78048000 bb6e - 1 audiosup (Audio Device Support 1.12)
61 78152000 39a97 - 1 mixer (Audio Mixer 1.49)
#
```

この時点で、オンボードのオーディオドライバ audiocs は読み込み解除され、オープンすることはできません。ただし、主オーディオデバイスの /dev/audio は、アプリケーションによって開かれている場合は何も変化しません。

8. (省略可能) USB オーディオデバイスを差し込みます。

9. (省略可能) 新しいオーディオリンクを調べます。

```
% ls -lt /dev/audio*
lrwxrwxrwx 1 root root 7 Jul 23 16:12 /dev/audio -> sound/1
lrwxrwxrwx 1 root root 10 Jul 23 16:12 /dev/audioc1 ->
sound/1c1
% ls -lt /dev/sound/*
lrwxrwxrwx 1 root root 77 Jul 23 16:12 /dev/sound/1c1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,...
lrwxrwxrwx 1 root root 74 Jul 23 16:12 /dev/sound/1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,...
lrwxrwxrwx 1 root root 66 Jul 23 15:59 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx 1 root root 69 Jul 23 15:59 /dev/sound/0c1 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,aud...
%
```

## ▼ 使用されていない USB オーディオデバイスのリンクを削除する方法

システムの電源がオフのときに USB オーディオデバイスを取り外した場合には、次の手順を実行します。システムの電源がオフのときに USB オーディオデバイスを取り外した場合、/dev/audio デバイスが、存在しない /dev/sound/\* デバイスを指したままになることがあります。

1. スーパーユーザーになります。

2. すべてのオーディオアプリケーションを閉じます。
3. 使用されていないオーディオリンクを削除します。

```
devfsadm -C -c audio
```

---

## USB オーディオデバイスに関する問題の障害追跡

この節では、USB オーディオデバイス問題の障害追跡の方法を説明します。

### USB スピーカ障害に対処する

ドライバを適用し、音量も上げているのに、USB スピーカから音が出ないことがあります。デバイスをホットプラグしてもこの動作が変化しないことがあります。

この問題を解決するには、USB スピーカの電源を再投入します。

### オーディオデバイスの所有権に関する注意事項

オーディオデバイスを操作するときは、オーディオデバイスの所有権に関する、次に挙げる点に注意してください。

- USB オーディオデバイスを差し込む時にコンソールからログインしていると、コンソールが `/dev/*` エントリの所有者になります。つまり、コンソールでログインしている限り、オーディオデバイスを使用できることとなります。
- USB オーディオデバイスを差し込む時にコンソールにログインしていない場合、`root` がそのデバイスの所有者になります。ただし、その後にコンソールにログインして USB オーディオデバイスにアクセスしようとする、デバイスの所有権はコンソールに変更されます。詳細については、`logindevperm(4)` のマニュアルページを参照してください。
- リモートから `rlogin` コマンドでログインして USB オーディオデバイスにアクセスしようとした場合は、所有権は変更されません。たとえば、権限のないユーザーが、他の人の所有するマイクを通して行われる会話を聞くことはできません。

---

## cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)

| 作業                        | 説明                                            | 参照先                                    |
|---------------------------|-----------------------------------------------|----------------------------------------|
| 1. USB デバイス情報を表示する        | USB デバイスおよびバスについての情報を表示する                     | 439 ページの「USB デバイス情報を表示する方法」            |
| 2. USB デバイスの構成を解除する       | システムに物理的に接続されている USB デバイスを論理的に構成解除する          | 441 ページの「USB デバイスの構成を解除する方法」           |
| 3. USB デバイスを構成する          | 以前に構成を解除した USB デバイスを構成する                      | 441 ページの「USB デバイスの構成方法」                |
| 4. 論理的に USB デバイスの接続を解除する  | 物理的にシステムの近くにいないときには、USB デバイスを論理的に切り離すことができる   | 441 ページの「論理的に USB デバイスを接続解除する方法」       |
| 5. 論理的に USB デバイスに接続する     | 以前に論理的に接続解除または構成解除した USB デバイスを物理的に接続する        | 442 ページの「論理的に USB デバイスを接続する方法」         |
| 6. USB デバイスのサブツリーの接続を解除する | ハブの下位階層 (または下位のツリー) である USB デバイスサブツリーの接続を解除する | 442 ページの「論理的に USB デバイスのサブツリーを接続解除する方法」 |
| 7. USB デバイスをリセットする        | USB デバイスをリセットして、デバイスを論理的に削除し、再作成する            | 443 ページの「USB デバイスのリセット方法」              |

---

## cfgadm コマンドを使った USB デバイスのホットプラグ

cfgadm コマンドを使用せずに稼働中のシステムから USB デバイスを追加または削除することができます。ただし、USB デバイスは、デバイスを物理的に削除しなくても「論理的に」ホットプラグすることができます。この方法は、リモートで作業中に機能していない USB デバイスを無効にしたりリセットしたりする必要がある場合に便利です。cfgadm コマンドを使うと、メーカーや製品情報を含む USB デバイスツリーを表示することもできます。

cfgadm コマンドは接続点についての情報を表示します。接続点とは、動的再構成を行うことができるシステム内の特定の場所のことです。接続点は、次の要素から構成されています。

- 占有装置 (occupant): システムに構成できるハードウェアリソースのことです。
- 受容体 (receptacle): 占有装置を受け入れる場所のことです。

接続点は、論理と物理の両方の接続点 ID (Ap\_Id) で表現されます。物理 Ap\_Id は接続点の物理的なパス名です。論理 Ap\_Id は物理 Ap\_Id に代わるユーザーに理解しやすい ID です。Ap\_Id の詳細については、cfgadm\_usb (1M) のマニュアルページを参照してください。

cfgadm コマンドを使用すると、USB デバイスステータス情報を取得できます。

| 受容体の状態                    | 説明                                                         |
|---------------------------|------------------------------------------------------------|
| empty/unconfigured        | デバイスが接続されていない                                              |
| disconnected/unconfigured | デバイスが論理的に接続解除されており利用不可。デバイスが物理的に接続されていても、devinfo ノードが削除される |
| connected/unconfigured    | デバイスは論理的に接続されているが利用不可。devinfo ノードが存在する                     |
| connected/configured      | デバイスは接続されており利用可能                                           |

次の節では、cfgadmin コマンドを使用した USB デバイスのホットプラグ方法について説明します。次のすべてのサンプル USB デバイス情報は、関連した情報に焦点を合わせるために一部省略されています。

## USB デバイス情報を表示する方法

prtconf コマンドを使用して USB デバイス情報を表示します。

```
$ prtconf
usb, instance #0
 hub, instance #2
 device, instance #8
 interface (driver not attached)
 printer (driver not attached)
 mouse, instance #14
 device, instance #9
 keyboard, instance #15
 mouse, instance #16
 storage, instance #7
 disk (driver not attached)
 communications, instance #10
 modem (driver not attached)
 data (driver not attached)
```

```

storage, instance #0
 disk (driver not attached)
storage, instance #1
 disk (driver not attached)

```

cfgadm コマンドを使用して USB バス情報を表示します。たとえば、次のようになります。

```

% cfgadm
Ap_Id Type Receptacle Occupant Condition
usb0/4.5 usb-hub connected configured ok
usb0/4.5.1 usb-device connected configured ok
usb0/4.5.2 usb-printer connected configured ok
usb0/4.5.3 usb-mouse connected configured ok
usb0/4.5.4 usb-device connected configured ok
usb0/4.5.5 usb-storage connected configured ok
usb0/4.5.6 usb-communi connected configured ok
usb0/4.5.7 unknown empty unconfigured ok
usb0/4.6 usb-storage connected configured ok
usb0/4.7 usb-storage connected configured ok

```

前の例で usb0/4.5.1 は、第 2 レベルの外部ハブのポート 1 に接続されているデバイスを識別します。この第 2 レベルハブは第 1 レベルの外部ハブのポート 5 に接続されており、また第 1 レベルのハブは最初の USB コントローラのルートハブであるポート 4 に接続されています。

次の cfgadm コマンドを使用して、特定の USB デバイス情報を表示します。たとえば、次のようになります。

```

% cfgadm -l -s "cols=ap_id:info"
Ap_Id Information
usb0/4.5.1 Mfg: Inside Out Networks Product: Edgeport/421 NConfigs:
1 Config: 0 : ...
usb0/4.5.2 Mfg: <undef> Product: <undef> NConfigs: 1 Config: 0
<no cfg str descr>
usb0/4.5.3 Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1 Config: 0
<no cfg str descr>
usb0/4.5.4 Mfg: NMB Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0
usb0/4.5.5 Mfg: Hagiwara Sys-Com Product: SmartMedia R/W NConfigs:
1 Config: 0 : Default
usb0/4.5.6 Mfg: 3Com Inc. Product: U.S.Robotics 56000 Voice USB Modem
NConfigs: 2 ... usb0/4.5.7
usb0/4.6 Mfg: Iomega Product: USB Zip 250 NConfigs: 1 Config: 0
: Default
usb0/4.7 Mfg: Iomega Product: USB Zip 100 NConfigs: 1 Config: 0
: Default
#

```



## ▼ USB デバイスの構成を解除する方法

システムに物理的に接続されている USB デバイスの構成を解除することはできますが、ドライバを適用することはできません。USB デバイスの構成を解除すると、そのデバイスが `prtconf` 出力に表示されます。

1. スーパーユーザーになります。
2. USB デバイスの構成を解除します。

```
cfgadm -c unconfigure usb0/4.7
Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7
This operation will suspend activity on the USB device
Continue (yes/no)? y
```

3. デバイスの構成が解除されていることを確認します。

```
cfgadm
Ap_Id Type Receptacle Occupant Condition
usb0/4.5 usb-hub connected configured ok
usb0/4.5.1 usb-device connected configured ok
usb0/4.5.2 usb-printer connected configured ok
usb0/4.5.3 usb-mouse connected configured ok
usb0/4.5.4 usb-device connected configured ok
usb0/4.5.5 usb-storage connected configured ok
usb0/4.5.6 usb-communi connected configured ok
usb0/4.5.7 unknown empty unconfigured ok
usb0/4.6 usb-storage connected configured ok
usb0/4.7 usb-storage connected unconfigured ok
```

## ▼ USB デバイスの構成方法

1. スーパーユーザーになります。
  2. USB デバイスを構成します。
- ```
# cfgadm -c configure usb0/4.7
```
3. USB デバイスが構成されていることを確認します。

```
# cfgadm usb0/4.7
Ap_Id                               Type           Receptacle   Occupant     Condition
usb0/4.7                             usb-storage    connected    configured   ok
```

▼ 論理的に USB デバイスを接続解除する方法

システムから USB デバイスを取り外し、`prtconf` 出力を削除したいが、物理的にシステムの近くにいない場合、USB デバイスの接続を論理的に解除できます。デバイスが物理的に接続されているが、論理的に接続解除されている場合には、そのデバイスは使用できず、システムにも表示されません。

1. スーパーユーザーになります。
2. **USB** デバイスの接続を解除します。
3. デバイスが接続解除されていることを確認します。

```
# cfgadm -c disconnect -y usb0/4.7
```

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	unknown	disconnected	unconfigured	ok

▼ 論理的に USB デバイスを接続する方法

次の手順を使用して、以前に論理的に接続解除または構成解除された USB デバイスを論理的に接続します。

1. スーパーユーザーになります。
2. **USB** デバイスを接続します。
3. デバイスが接続されていることを確認します。

```
# cfgadm -c configure usb0/4.7
```

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

デバイスを利用できるようになり、システムにも表示されるようになります。

▼ 論理的に USB デバイスのサブツリーを接続解除する方法

次の手順を使用して、USB デバイスのサブツリーを接続解除します。サブツリーは、ハブの下位デバイスの階層 (ツリー) です。

1. スーパーユーザーになります。
2. **USB** デバイスサブツリーを削除します。
3. **USB** デバイスサブツリーの接続解除を確認します。

```
# cfgadm -c disconnect -y usb0/4
```

```
# cfgadm usb0/4
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4	unknown	disconnected	unconfigured	ok

▼ USB デバイスのリセット方法

USB デバイスでエラーが発生した場合は、`cfgadm` コマンドを使ってデバイスをリセットします。このコマンドを使うと、デバイスを論理的に削除し、再作成することができます。

1. スーパーユーザーになります。
2. デバイスが使用中でないことを確認します。
3. デバイスをリセットします。

```
# cfgadm -x usb_reset -y usb0/4.7
```

4. デバイスが接続されていることを確認します。

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

第 30 章

デバイスへのアクセス (概要)

ここでは、システム上のデバイスにアクセスする方法について説明します。

この章の内容は次のとおりです。

- 445 ページの「デバイスへのアクセス」
- 447 ページの「論理ディスクデバイス名」
- 450 ページの「論理テープデバイス名」
- 451 ページの「論理リムーバブルメディアデバイス名」

デバイスの構成についての概要は、第 27 章を参照してください。

デバイスへのアクセス

コマンドを使用してディスク、ファイルシステムなどのデバイスを管理する場合、デバイス名を指定する方法を知っている必要があります。通常、論理デバイス名を使用して、システムに接続されたデバイスを表すことができます。論理デバイス名と物理デバイス名は、システム上でそれぞれ論理デバイスファイルと物理デバイスファイルによって表現されます。

デバイス情報が作成される方法

システムがブートされると、デバイス階層が作成されて、システムに接続されたすべてのデバイスが表示されます。カーネルは、このデバイス階層情報を使用して、ドライバを該当するデバイスに対応づけて、特定の操作を実行するドライバへの一連のポインタを与えます。デバイス階層の詳細については、『*OpenBoot 3.x コマンド・リファレンスマニュアル*』を参照してください。

デバイスの管理方法

devfsadm コマンドによって、/dev と /devices ディレクトリ内にある特殊デバイスファイルを管理します。デフォルトでは、devfsadm コマンドはすべてのドライバをシステムに読み込み、可能な限りのデバイスに接続しようとします。次に、devfsadm コマンドはデバイスファイルを /devices ディレクトリに作成し、論理リンクを /dev ディレクトリに作成します。devfsadm コマンドは、/dev と /devices のディレクトリの管理に加えて、path_to_inst(4) インスタンスデータベースも保守します。

動的再構成イベントに応答する、再構成ブート処理とディレクトリ /dev および /devices の更新は、両方とも devfsadmd (devfsadm コマンドのデーモン版) によって処理されます。このデーモンは、システムブート時に /etc/rc* スクリプトから起動します。

devfsadmd デーモンは再構成イベントによるデバイス構成の変化を自動的に検出するため、このコマンドを対話的に実行する必要はありません。

詳細については、devfsadm(1M) のマニュアルページを参照してください。

デバイス名の命名規則

Solaris 環境では、デバイスは次の 3 つの方法で参照されます。

- 物理デバイス名 – デバイス情報階層の完全デバイスパス名を表します。物理デバイス名は、次のコマンドを使用して表示できます。

- dmesg
- format
- sysdef
- prtconf

物理デバイスファイルは、/devices ディレクトリにあります。

- インスタンス名 – システム上のデバイスすべてのカーネル短縮名を表します。たとえば、sd0 と sd1 は、2 つのディスクデバイスのインスタンス名を表します。インスタンス名は、/etc/path_to_inst ファイルにマップされており、次のコマンドによって表示できます。

- dmesg
- sysdef
- prtconf

- 論理デバイス名 – デバイスを参照する際に、ほとんどのファイルシステムコマンドで使用されます。論理デバイス名を使用するファイルコマンドについては、表 30-1 を参照してください。/dev ディレクトリの論理デバイスファイルは、/devices ディレクトリの物理デバイスファイルにシンボリックリンクされています。

論理ディスクデバイス名

論理デバイス名は、次の場合に、ディスクデバイスにアクセスするために使用されます。

- システムに新しいディスクを追加する
- システム間でディスクを移動する
- ローカルディスク上にあるファイルシステムにアクセスまたはそれをマウントする
- ローカルファイルシステムのバックアップをとる

管理コマンドの多くは、ディスクスライスまたはファイルシステムを参照する引数を使用します。

シンボリックリンクされるサブディレクトリ (`/dev/dsk` または `/dev/rdisk` のどちらか) に続けて、特定のコントローラ、ディスク、およびスライスを識別する文字列を指定することによって、ディスクデバイスを参照してください。

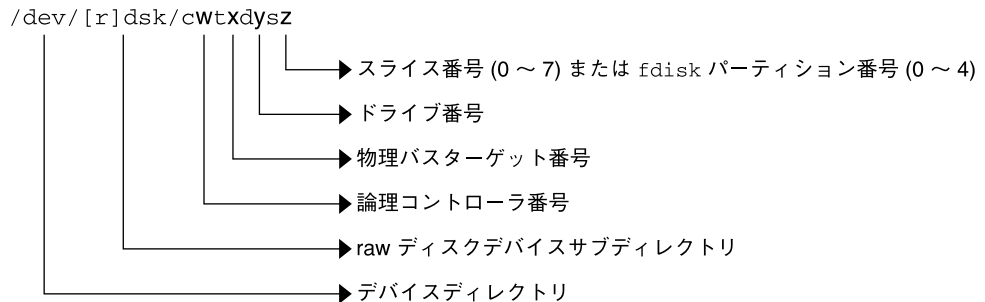


図 30-1 論理デバイス名

ディスクサブディレクトリの指定

ディスクとファイルの管理コマンドには、`raw` (または「キャラクタ型」) デバイスインタフェースか、「ブロック」デバイスインタフェースを使用する必要があります。この区別は、データがデバイスから読み取られる方法によって決まります。

`raw` デバイスインタフェースは、一度に少量のデータだけを転送します。ブロックデバイスインタフェースには、大量のデータブロックが一度に読み取られるバッファが含まれます。

コマンドによって、必要なインタフェースは異なります。

- コマンドが `raw` デバイスインタフェースを必要とする場合は、`/dev/rdisk` サブディレクトリを指定してください。(rdisk の `r` は、`raw` を表します。)

- コマンドがブロックデバイスインタフェースを必要とする場合は、`/dev/dsk` サブディレクトリを指定してください。
- コマンドが `/dev/dsk` または `/dev/rdisk` のどちらを必要とするかがわからない場合は、そのコマンドのマニュアルページの説明を参照してください。

次の表に、一般的に使用されるディスクコマンドとファイルシステムコマンドの一部に必要なインタフェースを示します。

表 30-1 使用頻度の高いコマンドに必要なデバイスインタフェースのタイプ

コマンド	インタフェースのタイプ	使用例
<code>df (1M)</code>	ブロック	<code>df /dev/dsk/c0t3d0s6</code>
<code>fsck (1M)</code>	Raw	<code>fsck -p /dev/rdisk/c0t0d0s0</code>
<code>mount (1M)</code>	ブロック	<code>mount /dev/dsk/c1t0d0s7 /export/home/ziggy</code>
<code>newfs (1M)</code>	Raw	<code>newfs /dev/rdisk/c0t0d1s1</code>
<code>prtvtoc (1M)</code>	Raw	<code>prtvtoc /dev/rdisk/c0t0d0s2</code>

スライスの指定

特定のスライスを特定のディスクで識別するために使用する文字列は、コントローラのタイプが、直接またはバス指向のどちらであるかによって決まります。次の表に、直接コントローラとバス指向コントローラのどちらであるかによって、プラットフォームが何になるかを示します。

表 30-2 コントローラのタイプ

直接コントローラ	バス指向コントローラ
IDE (x86)	SCSI (SPARC/x86)
	FCAL (SPARC)
	ATA (SPARC/x86)

両方のタイプのコントローラについて、次の節で説明します。

注 - コントローラ番号は、システム初期設定時に自動的に割り当てられます。この番号は、厳密に論理的なものであり、物理コントローラに直接対応するものではありません。

x86: 直接コントローラでアクセスされるディスク

x86 システムにおいて IDE コントローラでアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

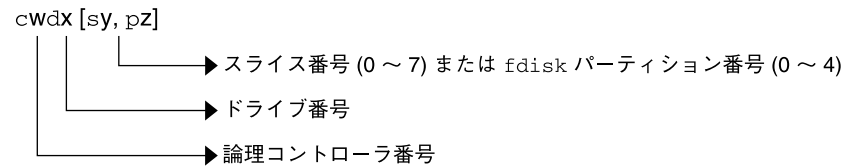


図 30-2 x86: 直接コントローラでアクセスされるディスク

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

SPARC: バス指向コントローラでアクセスされるディスク

SPARC システムにおいてバス指向コントローラ (SCSI など) でアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

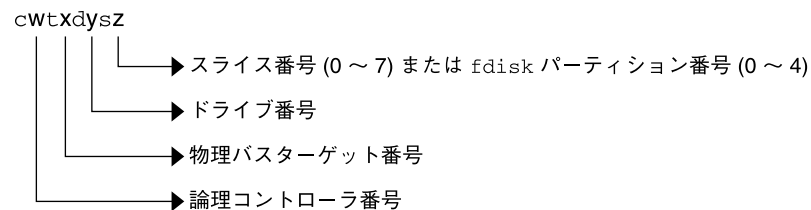


図 30-3 SPARC: バス指向コントローラでアクセスされるディスク

直接接続されるディスク (Ultra10 の IDE ディスクなど) を備えている SPARC システムでは、バス指向コントローラを備えているシステムと同じ命名規則になります。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

SCSI コントローラの場合、*x* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*y* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*y* は通常 0 になります。

ディスク全体を示すには、スライス 2 (s2) を指定してください。

x86: SCSI コントローラでアクセスされるディスク

x86 システムにおいて SCSI コントローラでアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

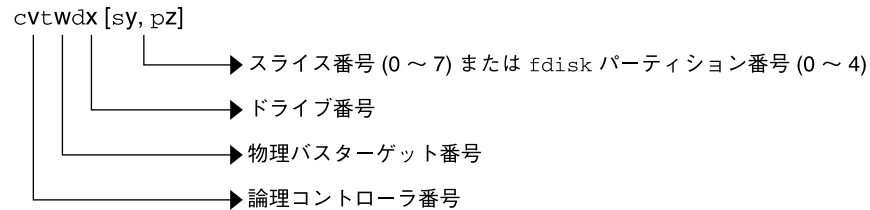


図 30-4 x86: SCSI コントローラでアクセスされるディスク

システムにコントローラが1つしかない場合、*v*は、通常、0になります。

SCSI コントローラの場合、*w* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*x* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*x* は通常 0 になります。

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

論理テープデバイス名

論理テープデバイスファイルは、/devices ディレクトリからのシンボリックリンクとして、/dev/rmt/* ディレクトリにあります。

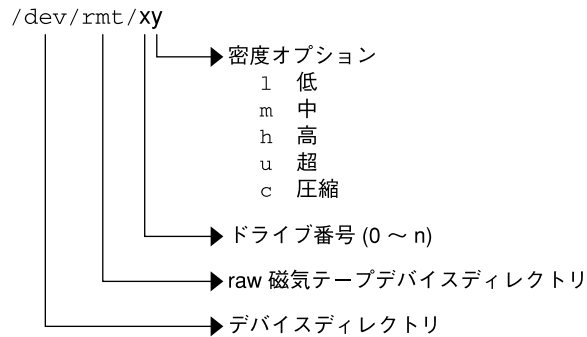


図 30-5 論理テープデバイス名

システムに接続された最初のテープデバイスは 0 (`/dev/rmt/0`) です。テープ密度の値 (l、m、h、c、および u) の詳細については、第 52 章を参照してください。

論理リムーバブルメディアデバイス名

リムーバブルメディアは、ボリューム管理 (vol1d) によって管理されるため、論理デバイス名は、手動でメディアをマウントしない限り、通常使用されません。

システムのリムーバブルメディアデバイスを表す、論理デバイス名については、第 18 章の説明を参照してください。

第 31 章

ディスクの管理

以下の各章で、ディスク管理について説明します。

第 32 章	Solaris のディスクスライスの概要と、 <code>format</code> ユーティリティを紹介します。
第 33 章	ディスクのフォーマット、ディスクラベルの検査、欠陥ディスクセクターの修復を行う手順について説明します。
第 34 章	SPARC システムにディスクを追加する手順について説明します。
第 35 章	x86 システムにディスクを追加する手順について説明します。
第 36 章	<code>format</code> ユーティリティのメニューとコマンドについて説明します。また、 <code>format.dat</code> ファイル、 <code>format</code> コマンドの入力規則、ヘルプ機能の使い方についても説明します。

第 32 章

ディスクの管理 (概要)

この章では、Solaris ディスクスライスの概要および `format` ユーティリティについて説明します。

この章の内容は次のとおりです。

- 455 ページの「Solaris 9 8/03 リリースの新しいディスク管理機能」
- 460 ページの「Solaris 9 リリースの新しいディスク管理機能」
- 460 ページの「ディスク管理作業についての参照先」
- 461 ページの「ディスク管理の概要」
- 461 ページの「ディスク関連の用語」
- 462 ページの「ディスクスライスについて」
- 467 ページの「`format` ユーティリティ」
- 471 ページの「ディスクラベルについて」
- 474 ページの「ディスクをスライスに分割する」

システムにディスクを追加する方法については、第 34 章または第 35 章を参照してください。

Solaris 9 8/03 リリースの新しいディスク管理機能

この節では、この Solaris リリースの新しいディスク管理機能について説明します。

SPARC: EFI ディスクラベルによるマルチテラバイトボリュームのサポート

この Solaris リリースでは、64 ビット Solaris カーネルを実行しているシステムで 1T バイトを超えるディスクをサポートします。32 ビット Solaris カーネルを実行するシステム (Solaris x86 プラットフォーム版を実行するシステムなど) に接続されたディスクでは、EFI (Extensible Firmware Interface) ディスクラベルは使用できません。

EFI の仕様は

http://www.intel.com/technology/efi/main_specification.htm からダウンロードできます。

EFI ラベルは、物理ディスクボリュームと仮想ディスクボリュームをサポートします。このリリースには、1T バイトを超えるディスクを管理するための更新版ディスクユーティリティが付属しています。UFS ファイルシステムには EFI ディスクラベルとの互換性があり、1T バイトを超えるサイズの UFS ファイルシステムを作成することができます。マルチテラバイトの UFS ファイルシステムの作成方法については、543 ページの「SPARC: マルチテラバイトの UFS ファイルシステムのサポート」を参照してください。

今回の Solaris リリースでは、Solaris ボリュームマネージャでも 1T バイトを超えるディスク管理が可能です。Solaris ボリュームマネージャの使用方法については、『Solaris ボリュームマネージャの管理』を参照してください。

サイズが 1T バイトに満たないディスクでは、引き続き VTOC ラベルを使用できます。現在のシステムでサイズが 1T バイトに満たないディスクしか使用しない場合は、以前の Solaris リリースと同じ方法でディスク管理を行います。サイズが 1T バイトに満たないディスクに EFI ラベルでラベル付けを行う場合は、`format-e` コマンドを使用します。詳細については、485 ページの「例 — サイズが 1T バイトに満たないディスクに EFI ラベルを付ける」を参照してください。

EFI ラベルと VTOC ラベル

EFI ディスクラベルと VTOC ディスクラベルとの相違点は次のとおりです。

- サイズが 1T バイトを超えるディスクをサポートする
- スライス 0 ~ 6 を使用できる (スライス 2 はその他のスライス)
- 一次 (バックアップ) ラベルまたはその他のパーティションを使ってパーティションやスライスをオーバーラップすることはできない。EFI ラベルのサイズは通常 34 セクターであり、パーティションの開始セクターは 34 になる。つまり、開始セクターが 0 のパーティションは存在しない
- シリンダ、ヘッド、およびセクターの情報は、ラベルには格納されない。サイズはブロック単位で報告される
- 代替シリンダ領域 (ディスクの終わりから 2 つめまでのシリンダ) に格納されていた情報は、スライス 8 に格納される

- `format` ユーティリティを使ってパーティションサイズを変更する場合、サイズ 0 のパーティションには `unassigned` パーティションタグが割り当てられる。`format` ユーティリティでは、0 より大きいサイズのパーティションには、デフォルトにより `usr` パーティションタグが割り当てられる。パーティションを変更したあと新しいパーティションタグを割り当てたい場合は、パーティション変更メニューを使用する。ただし、サイズ 0 以外のパーティションに `unassigned` パーティションタグを割り当て直すことはできない

EFI ディスクラベルの制限

サイズが 1T バイトを超えるディスクを使用することが現在の環境にとって適切かどうかを判断するときは、次の点を考慮してください。

- EFI ラベル付きディスクを使用するシステム向けの階層化されたソフトウェア製品で、EFI ラベル付きディスクにアクセスできないことがある
- 以前の Solaris リリースを実行しているシステムは、EFI ラベル付きディスクを認識しない
- IDE ディスクでは EFI ディスクラベルがサポートされていない
- EFI ラベル付きディスクからブートを実行できない
- Solaris 管理コンソールの「ディスク」ツールでは、EFI ラベル付きディスクを管理できない。EFI ラベル付きディスクを管理したい場合は、`format` ユーティリティを使ってディスクをパーティションに分割したあと、`format` ユーティリティか、Solaris 管理コンソールの拡張ストレージツールを使用する
- EFI 仕様では、スライスのオーバーラップは禁止されている。`cxydz` でディスク全体を表現する
- UFS ファイルシステムには、EFI ディスクラベルとの互換性がある。Solaris 9 8/03 リリース以降は、1T バイトを超える UFS ファイルシステムを作成できる。マルチテラバイトの UFS ファイルシステムの詳細については、543 ページの「SPARC: マルチテラバイトの UFS ファイルシステムのサポート」を参照
1T バイトを超えるファイルシステムを作成する必要がある場合は、別パッケージの Sun QFS ファイルシステムも使用できる。Sun QFS ファイルシステムについては、<http://docs.sun.com/> で、『Sun QFS, Sun SAM-FS, Sun SAM-QFS ファイルシステム管理マニュアル』を参照
- ディスクやパーティションのサイズ情報を提供する。使用可能な単位はセクターまたはブロック。シリンダおよびヘッドは使用できない
- 次の `format` オプションは、EFI ラベル付きディスクではサポートされていないか、不適切である
 - `save` オプション。EFI ラベル付きディスクは `format.dat` ファイルにエントリを必要としない。したがって、このオプションはサポートされない
 - `backup` オプション。ディスクドライバは、一次ラベルを検出してディスクに書き込む。したがって、このオプションは不適切である

EFI ラベル付きディスクを使用したシステムへのインストール

Solaris インストールユーティリティは、EFI ラベル付きディスクを自動的に認識します。しかし、これらのディスクのパーティション分割をやり直すことはできません。インストール前またはインストール後にこれらのディスクのパーティション分割をやり直したい場合は、format ユーティリティを使用してください。Solaris Upgrade ユーティリティおよび Live Upgrade ユーティリティも EFI ラベル付きディスクを認識します。しかし、EFI ラベル付きディスクからシステムのブートを実行することはできません。

EFI ラベル付きディスクを使用するシステム上に Solaris をインストールした場合、次のようなパーティションテーブル情報が得られます。

Current partition table (original):

Total disk sectors available: 2576924638 + 16384 (reserved sectors)

Part	Tag	Flag	First Sector	Size	Last Sector
0	root	wm	34	1.20TB	2576924636
1	unassigned	wm	0	0	0
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	unassigned	wm	0	0	0
8	reserved	wm	2576924638	8.00MB	2576941021

EFI ラベル付きディスクの管理

EFI ラベル付きディスクの管理方法は、次の表で確認できます。

作業	参照先
インストール済みのシステムにディスクを接続し、再構成 (ブート) を実行する	499 ページの「SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)」
format ユーティリティを使ってディスクのパーティション分割を行う (必要に応じて)	503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」
ディスクボリュームを作成する。その後、Solaris ボリュームマネージャでソフトパーティションを作成する (必要に応じて)	『Solaris ボリュームマネージャの管理』の「記憶装置管理の概念」
newfs コマンドを使用して新しいディスクの UFS ファイルシステムを作成する	508 ページの「SPARC: ファイルシステムを作成する方法」
または、QFS ファイルシステムを作成する	http://docs.sun.com/db/coll/20445.2

EFI ラベル付きディスクの複製

以前の Solaris リリースでは、ディスク全体がスライス 2 (s2) で表されていました。dd コマンドを使って、ディスクを複製またはコピーできます。このコマンドは、次のように入力します。

```
dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2 bs=128k
```

ここでは、複製後のディスクに一意的 UUID を割り当てる必要があります。したがって、1T バイトを超えるサイズのディスクの複製 (コピー) 方法が若干異なります。たとえば、次のようになります。

1. dd コマンドを使って EFI ラベル付きディスクを複製します。

```
# dd if=/dev/rdisk/c0t0d0 of=/dev/rdisk/c0t2d0 bs=128k
```

2. 複製対象のディスクの prtvtoc 出力を fmthard コマンドにパイプして、複製済みディスク用の新しいラベルを作成します。

```
# prtvtoc /dev/rdisk/c0t0d0 | fmthard -s - /dev/rdisk/c0t2d0
```



注意 - 複製済みディスク用の新しいラベルは、必ず作成してください。作成しないと、その他のソフトウェア製品によって UUID の重複が検出された時点で、EFI ラベル付きディスクのデータが破壊されることがあります。

EFI ディスクラベルに関する問題の障害追跡

EFI ラベル付きディスクに関する問題の障害追跡には、次のエラーメッセージと解決法を参考にしてください。

エラーメッセージ

```
Dec 3 09:26:48 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/sf@1,0/ssd@w50020f23000002a4,0 (ssd1):  
Dec 3 09:26:48 holoship disk has 2576941056 blocks, which is too large  
for a 32-bit kernel
```

原因

32 ビット SPARC カーネルを実行しているシステムを、サイズが 1T バイトを超えるディスクでブートしようとしてしました。

解決法

サイズが 1T バイト以上のディスクでは、64 ビット SPARC カーネルを実行しているシステムをブートしてください。

エラーメッセージ

```
Dec 3 09:12:17 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/sf@1,0/ssd@w50020f23000002a4,0 (ssd1):  
Dec 3 09:12:17 holoship corrupt label - wrong magic number
```

原因

Solaris 9 8/03 より古い Solaris リリースで動作するシステムにこのディスクを追加しようとした。

解決法

このディスクは、EFI ディスクラベルをサポートしている Solaris リリースで動作するシステムに追加してください。

Solaris 9 リリースの新しいディスク管理機能

この節では、Solaris 9 リリースの新しいディスク管理機能について説明します。

Solaris ボリュームマネージャとソフトパーティション

これまでバンドルされていなかった Solstice DiskSuite™ 製品が、Solaris ボリュームマネージャという名前で Solaris 9 リリースに含まれることになりました。Solaris ボリュームマネージャの新機能である「ソフトパーティション」を利用すると、各ディスクに 8 つ以上のパーティションを作成できます。

Solaris ボリュームマネージャの詳細については、『Solaris ボリュームマネージャの管理』の「記憶装置管理の概念」を参照してください。ソフトパーティションの詳細は、『Solaris ボリュームマネージャの管理』の「ソフトパーティション (概要)」を参照してください。

ディスク管理作業についての参照先

ディスク管理の手順については、次を参照してください。

ディスク管理作業	参照先
ディスクをフォーマットしてディスクラベルを確認する。	第 33 章

ディスク管理作業	参照先
SPARC システムに新しいディスクを追加する。	第 34 章
x86 システムに新しいディスクを追加する。	第 35 章
SCSI または PCI ディスクのホットプラグを実行する。	第 28 章

ディスク管理の概要

一般に、Solaris 環境におけるディスクの管理とは、システムを設定し、Solaris インストールプログラムを実行し、適切なディスクスライスおよびファイルシステムを作成してオペレーティングシステムをインストールすることを意味します。また、`format` ユーティリティを使用して、新しいディスクドライブを追加したり、欠陥ディスクドライブを交換したりしなければならない場合もあります。

注 - Solaris オペレーティング環境は、SPARC と x86 の 2 種類のハードウェア (プラットフォーム) で動作します。また、Solaris オペレーティング環境は、64 ビットのアドレス空間でも 32 ビットのアドレス空間でも動作します。このマニュアルで説明する情報は、章、節、注、箇条書き項目、図、表、例、コード例などで特に明記しないかぎり、両方のプラットフォーム、およびアドレス空間に適用されます。

ディスク関連の用語

この節の説明を有効に利用するには、基本的なディスクアーキテクチャを理解しておく必要があります。特に、次の用語を理解しておいてください。

用語	説明
トラック	ディスクが回転するときに 1 つの静止したディスクヘッドの下を通過する同心リング
シリンダ	ディスクが回転する軸から同じ距離にあるトラックの集まり
セクター	各ディスクプラッタのセクション。1 セクターは 512 バイト

用語	説明
ブロック	ディスク上のデータ記憶領域。1 ディスクブロックは 512 バイト
ディスクコントローラ	ディスクドライブを制御するチップおよび関連する回路
ディスクラベル	ディスクのジオメトリおよびパーティション情報を含む、ディスクの第 1 セクター
デバイスドライバ	ハードウェアまたは仮想デバイスを制御するカーネルモジュール

詳細については、ディスクメーカーの製品情報を参照してください。

ディスクスライスについて

ディスク上に格納されたファイルは、ファイルシステム中で管理されます。ディスク上の各ファイルシステムは「スライス」、つまりファイルシステム用に確保されたセクターセットのグループに割り当てられます。オペレーティングシステム (および、システム管理者) からは、各ディスクスライスは別個のディスクドライブであるかのように見えます。

ファイルシステムの詳細については、第 38 章を参照してください。

注 - スライスをパーティションと呼ぶこともあります。このマニュアルでは「スライス」と呼びますが、`format` ユーティリティなど、特定のインタフェースではスライスを「パーティション」と呼びます。

スライスを設定するときには、次の規則に注意してください。

- 各ディスクスライスは、ファイルシステムを 1 つしか持てない。
- ファイルシステムを複数のスライスにまたがって割り当てることはできない。

SPARC プラットフォームと x86 プラットフォームでは、スライスの設定が少し異なります。次の表に、主な相違点を示します。

表 32-1 プラットフォームによるスライスの違い

SPARC プラットフォーム	x86 プラットフォーム
ディスク全体が Solaris オペレーティング環境用になる。	ディスクはオペレーティング環境ごとに1つの fdisk パーティションに分割される。
VTOC - ディスクは8スライス(スライス0~7)に分割される。	VTOC - Solaris の fdisk パーティションは10スライス(スライス0~9)に分割される。
EFI - ディスクは7スライス(スライス0~6)に分割される。	

SPARC: ディスクスライス

次の表に、SPARC システムのスライスの説明を示します。

表 32-2 SPARC: ディスクスライス

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	説明
0	ルート (/)	両方	オペレーティングシステムを構成するファイルとディレクトリを含む。 EFI - EFI ラベル付きディスクからはブートを実行できない。
1	スワップ	両方	仮想メモリ、つまり「スワップ空間」を提供する。
2	—	両方	VTOC - 慣例的にディスク全体を表す。このスライスのサイズは変更しないこと。 EFI - サイトの必要に応じてオプションスライスを定義する。
3	/export	両方	サイトの必要に応じて定義可能なスライス(任意)。 クライアントシステムが必要とする代替オペレーティングシステムを格納するため、サーバー上で使用できる。
4		両方	サイトの必要に応じて定義可能なスライス(任意)。

表 32-2 SPARC: ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	説明
5		両方	サイトの必要に応じて定義可能なスライス (任意)。 システムに追加されるアプリケーションソフトウェアを格納する。インストール時に、/opt ファイルシステムにスライスが割り当てられていなければ、スライス 0 に /opt ディレクトリが作成される。
6	/usr	両方	オペレーティングシステムのコマンド (「実行可能」コマンドとも呼ぶ) を含む。このスライスには、マニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。
7	/home または /export/home	両方	VTOC - ユーザーによって作成されたファイルを格納する。 EFI - 適用できない
8	なし	該当しない	VTOC - 適用できない。 EFI - デフォルトで作成された予約済みスライス。VTOC の代替シリンダによく似た領域。このスライスは変更または削除しないこと。

x86: ディスクスライス

x86 システム上では、ディスクは fdisk パーティションに分割されます。fdisk パーティションは、Solaris など、特定のオペレーティングシステムで使用するよう確保されたディスクの一部です。

次の表に示すように、Solaris は Solaris fdisk パーティション上に、0 から 9 までの番号が付いた 10 個のスライスを配置します。

表 32-3 x86: ディスクスライス

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	目的
0	ルート (/)	両方	オペレーティングシステムを構成するファイルとディレクトリを含む。

表 32-3 x86: ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	目的
1	スワップ	両方	仮想メモリー、つまり「スワップ空間」を提供する。
2	—	両方	慣例的に、このスライスは Solaris fdisk パーティション全体を表す。このスライスのサイズは変更しないこと。
3	/export	両方	サイトの必要に応じて定義可能なスライス (任意)。 クライアントシステムが必要とする代替オペレーティングシステムを格納するため、サーバー上で使用できる。
4			サイトの必要に応じて定義可能なスライス (任意)。
5		両方	サイトの必要に応じて定義可能なスライス (任意)。 システムに追加されるアプリケーションソフトウェアを格納する。インストール時に、/opt ファイルシステムにスライスが割り当てられていなければ、スライス 0 に /opt ディレクトリが作成される。
6	/usr	両方	オペレーティングシステムのコマンド (「実行可能」コマンドとも呼ぶ) を含む。このスライスには、マニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。
7	/home または /export/home	両方	ユーザーによって作成されるファイルを含む。
8	—	両方	Solaris がハードディスクからブートするために必要な情報を含む。このスライスのスライス番号は 8 であるが、Solaris fdisk パーティションの先頭に存在するため、ブートスライスと呼ばれる。
9	—	両方	代替ディスクブロック用に予約された領域。スライス 9 は代替セクタスライスと呼ばれる。

raw データスライスの使用

SunOS オペレーティングシステムは、各ディスクのブロック 0 にディスクラベルを格納します。これは、raw データスライスを作成する Sun 以外の製品のデータベースアプリケーションを使用するときは、ブロック 0 から開始してはならないことを意味します。この領域に raw データスライスを作成すると、ディスクラベルが上書きされて、ディスク上のデータにアクセスできなくなります。

ディスク上の次の領域は、raw データスライス用に使用しないでください。raw データスライスは Sun 以外のデータベースアプリケーションによって作成されることがあります。

- ブロック 0 (ディスクラベルが格納される領域)
- スライス 2 (VTOC ラベル付きディスク全体を表す)

複数のディスク上のスライス配置

十分な大きさのディスクであれば、1 台ですべてのスライスとそれに対応するファイルシステムを確保できますが、通常はシステムのスライスとファイルシステムを確保するために複数のディスクが使用されます。

注-1 つのスライスを複数のディスクに分割することはできません。ただし、複数のスワップスライスを別々のディスクに配置することはできます。

たとえば、1 台のディスクにルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムを入れ、別のディスクにユーザーデータを含む /export/home ファイルシステムやその他のファイルシステムを入れます。

複数のディスクを使用する場合、オペレーティングシステムソフトウェアとスワップ領域が入っているディスク (つまり、ルート (/)、/usr ファイルシステム、およびスワップ領域用のスライスが入っているディスク) を、「システムディスク」と呼びます。システムディスク以外のディスクを、「二次ディスク」または「非システムディスク」と呼びます。

システムのファイルシステムを複数のディスクに入れると、システムをシャットダウンしたりオペレーティングシステムソフトウェアをロードし直したりしなくても、二次ディスクのファイルシステムとスライスを変更できます。

また、複数のディスクを使用すると、入出力 (I/O) のパフォーマンスが改善されます。ディスク負荷を複数のディスクに分散すると、I/O のボトルネックを回避できます。

使用するスライスの決定

ディスクのファイルシステムを設定するときには、各スライスのサイズだけでなく、どのスライスを使用するかも決定します。どのように決定するかは、ディスクを接続するシステムの構成と、ディスクにインストールするソフトウェアによって異なります。

次のシステム構成があります。

- サーバー
- スタンドアロンシステム

システムの構成が異なれば、スライスの使用方法も異なります。次の表に例を挙げます。

表 32-4 システム構成とスライス

スライス	サーバー	スタンドアロンシステム
0	ルート	ルート
1	スワップ	スワップ
2	—	—
3	/export	—
6	/usr	/usr
7	/export/home	/home

システム構成の詳細については、124 ページの「システムタイプの概要」を参照してください。

注 - Solaris インストールプログラムは、インストール用に選択したソフトウェアに基づいてデフォルトのスライスサイズを表示します。

format ユーティリティ

使用方法や参照情報に進む前に以下の情報に目を通して、format ユーティリティの概要を確認してください。

format ユーティリティは、Solaris システム用にハードディスクドライブを用意するためのシステム管理ツールです。

次の表に、format ユーティリティの機能とその利点を示します。

表 32-5 format ユーティリティの機能と利点

機能	利点
システム内で接続されている全ディスクドライブを検索する	次の状態を報告する <ul style="list-style-type: none"> ■ ターゲットの位置 ■ ディスクのジオメトリ ■ ディスクがフォーマット済みかどうか ■ ディスク上にマウントされているパーティションが存在するかどうか
ディスクラベルを検索する	修復処理に使用する
欠陥セクターを修復する	回復可能なエラーが発生したディスクドライブをメーカーに返送しなくても、熟練した管理者なら修復できる
ディスクをフォーマットして、分析する	ディスク上でセクターを作成し、検査する
ディスクをパーティションに分割する	個々のファイルシステムを別々のスライス上で作成できるようにディスクを分割する
ディスクにラベルを付ける	後から検索できるように (通常は修復用)、ディスクにディスク名と構成情報を書き込む

format ユーティリティのオプションの詳細については、第 36 章を参照してください。

format ユーティリティを使用する場合

Solaris のインストール時に、Solaris インストールプログラムによってディスクがパーティションに分割され、ラベルが付けられます。次のような場合に、format ユーティリティを使用できます。

- スライス情報を表示する
- ディスクをスライスに分割する
- 既存のシステムにディスクドライブを追加する
- ディスクドライブをフォーマットする
- ディスクにラベルを付ける
- ディスクドライブを修復する
- ディスクのエラーを分析する

システム管理者が format ユーティリティを使用するのは、主にディスクをディスクスライスに分割するためです。これらの手順については、第 34 章と第 35 章を参照してください。

format ユーティリティの使用上のガイドラインについては、表 32-6 を参照してください。

format ユーティリティ使用上のガイドライン

表 32-6 format ユーティリティのガイドライン

作業	注意事項	参照先
ディスクをフォーマットする	<ul style="list-style-type: none">■ ディスクをフォーマットし直すと、既存のデータが失われる■ ディスクドライブをフォーマットしてパーティションに分割した状態で出荷するメーカーが増えているので、ディスクドライブをフォーマットする必要性は減少している。既存のシステムにディスクドライブを追加する場合は、format ユーティリティを使用しなくてもすむことがある■ ディスクを配置し直したら多数のディスクエラーが表示される場合は、フォーマットし直してみるとよい。不良セクターが自動的にマッピングし直される	480 ページの「ディスクをフォーマットする方法」
システムディスクを交換する	<ul style="list-style-type: none">■ 損傷したシステムディスクのデータは、バックアップメディアから復元しなければならない。復元しなければ、インストールプログラムを使用してシステムをもう一度インストールしなければならないくなる	501 ページの「SPARC: システムディスクを接続してブートする方法」または 513 ページの「x86: システムディスクを接続してブートする方法」、システムをインストールし直さなければならない場合は、『Solaris 9 12/03 インストールガイド』を参照
ディスクをスライスに分割する	<ul style="list-style-type: none">■ すでにスライスに分割されているディスクで、パーティションを再分割してラベルを付け直すと、既存のデータが失われる■ ディスクのパーティションを分割し直して復元する前に、既存のデータをバックアップメディアにコピーする必要がある	503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」または 521 ページの「x86: ディスクスライスとディスクラベルを作成する方法」
既存のシステムにディスクを追加する	<ul style="list-style-type: none">■ 二次ディスクをフォーマットし直すか、パーティションに分割し直す場合は、既存のデータをバックアップメディアから復元しなければならない	502 ページの「SPARC: 二次ディスクを接続してブートする方法」または 513 ページの「x86: 二次ディスクを接続してブートする方法」

表 32-6 format ユーティリティのガイドライン (続き)

作業	注意事項	参照先
ディスクドライブを修復する	<ul style="list-style-type: none"> ■ 顧客のサイトによっては、欠陥ドライブの修復ではなくドライブ自体の交換を希望する場合があります。サイトがディスクドライブのメーカーと保守契約を結んでいる場合は、format ユーティリティを使用してディスクドライブを修復する必要はない ■ 通常、ディスクドライブの修復とは、不良セクターを欠陥リストに追加することを意味する。新しいコントローラは不良セクターを自動的にマップし直すので、システムを中断する必要はない ■ システムに旧型のコントローラがある場合や、失われたデータを復元する場合は、不良セクターをマップし直す必要がある 	493 ページの「欠陥セクターの修復」

ディスクのフォーマット

ほとんどの場合、ディスクはメーカーまたは再販業者によってフォーマットされています。このため、ドライブをインストールするときにフォーマットし直す必要はありません。ディスクがフォーマットされているかどうかを判別するには、format ユーティリティを使用します。詳細については、479 ページの「ディスクがフォーマット済みかを調べる方法」を参照してください。

ディスクがフォーマットされていない場合、format ユーティリティを使用してフォーマットしてください。

ディスクのフォーマットでは、次の 2 つのステップが行われます。

- ディスクメディアを使用できるようにする
- 表面解析に基づいてディスクの欠陥リストを作成する



注意 - フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常はメーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、format ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドのみを使用するように注意してください。詳細については、480 ページの「ディスクをフォーマットする方法」を参照してください。

データに利用できる合計ディスク容量のうち、ごくわずかな容量が欠陥情報とフォーマット情報の格納に使用されます。この容量はディスクのジオメトリによって異なり、使用年数がたち欠陥箇所が多くなるにつれて、少なくなります。

ディスクの種類とサイズに応じて、フォーマットは数分から数時間かかります。

ディスクラベルについて

どのディスクにも、そのディスクのコントローラ、ジオメトリ、スライスに関する情報を格納する特殊な領域が確保されています。そのような情報をディスクの「ラベル」と呼びます。VTOC ラベル付きのディスク上のディスクラベルを「VTOC (Volume Table of Contents)」と呼びます。「ディスクにラベルを付ける」とは、ディスクにスライス情報を書き込むことを意味します。通常は、ディスクのスライスを変更した後にラベルを付けます。

スライスを作成した後でディスクにラベルを付けないと、オペレーティングシステムはスライスを「認識」する方法がないので、そのスライスを利用できなくなります。

パーティションテーブル

ディスクラベルのうち重要な部分は「パーティションテーブル」です。この部分は、ディスクのスライス、スライスの境界 (シリンダ単位)、スライスの合計サイズを表します。ディスクのパーティションテーブルは、format ユーティリティを使用して表示できます。次の表に、パーティションテーブル関連の用語を示します。

表 32-7 パーティションテーブル関連の用語

用語	値	説明
番号	0 - 7	VTOC -0 ~ 7 の番号が付いたパーティションまたはスライス EFI -0 ~ 6 の番号が付いたパーティションまたはスライス
タグ	0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME 11=RESERVED	一般にこのパーティションにマウントされたファイルシステムを記述する数値
フラグ	wm wu rm	パーティションは書き込み可能でマウント可能 パーティションは書き込み可能でマウント不可。これは、スワップ領域専用のパーティションのデフォルト状態である。ただし、mount コマンドでは「マウント不可」のフラグはチェックされない

表 32-7 パーティションテーブル関連の用語 (続き)

用語	値	説明
	rm	パーティションは読み取り専用でマウント可能

パーティションのフラグとタグは必ず割り当てられるので、管理する必要はありません。

パーティションテーブルを表示する手順については、481 ページの「ディスクスライス情報を表示する方法」または 486 ページの「ディスクラベルを検査する方法」を参照してください。

パーティションテーブル情報の表示

次の例は、format ユーティリティを使って、4.0G バイトの VTOC ラベル付きディスクのパーティションテーブルを表示したものです。

Total disk cylinders available: 8892 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	1110 - 4687	1.61GB	(0/3578/0) 3381210
1	swap	wu	0 - 1109	512.00MB	(0/1110/0) 1048950
2	backup	wm	0 - 8891	4.01GB	(0/8892/0) 8402940
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wm	0	0	(0/0/0) 0
7	home	wm	4688 - 8891	1.89GB	(0/4204/0) 3972780

format ユーティリティを使用して表示されるパーティションテーブルには、次の情報が含まれます。

列名	説明
Part	パーティション (またはスライス番号)。この列についての説明は、表 32-7 を参照。
Tag	パーティションのタグ。この列についての説明は、表 32-7 を参照。
Flags	パーティションのフラグ。この列についての説明は、表 32-7 を参照。
Cylinders	スライスの開始シリンダ番号と終了シリンダ番号を示す。
Size	スライスのサイズを M バイト単位で示す。
Blocks	合計シリンダ数と 1 スライス当たりの合計セクター数 (カラムの右端) を示す。

列名	説明
First Sector	EFI - 開始ブロック番号。
Last Sector	EFI - 終了ブロック番号。

次の例は、`prtvtoc` コマンドを使用して EFI ディスクラベルを表示した結果です。

```
# prtvtoc /dev/rdisk/c4t1d0s0
* /dev/rdisk/c4t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
* 2576941056 sectors
* 2576940989 accessible sectors
*
* Flags:
*  1: unmountable
* 10: read-only
*
*
* Partition      Tag  Flags      First      Sector      Last
* Partition      Tag  Flags      Sector     Count       Sector     Mount Directory
*   0             2    00          34        629145600   629145633
*   1             4    00     629145634   629145600   1258291233
*   6             4    00     1258291234  1318633404  2576924637
*   8             11   00     2576924638    16384     2576941021
*
* Flags:
*  1: unmountable
* 10: read-only
*
```

`prtvtoc` コマンドで、次の情報が取得できます。

列名	説明
Dimensions	このセクションでは、ディスクドライブの物理的な構成を示す。
Flags	このセクションでは、パーティションテーブルのセクションに記載されたフラグを示す。パーティションフラグについての説明は、表 32-7 を参照。
Partition (または Slice) テーブル	このセクションには次の情報が含まれます。
Partition	パーティション (またはスライス番号)。この列についての説明は、表 32-7 を参照。
Tag	パーティションのタグ。この列についての説明は、表 32-7 を参照。
Flags	パーティションのフラグ。この列についての説明は、表 32-7 を参照。

列名	説明
First Sector	スライスの最初のセクターを示す。
Sector Count	スライス内の合計セクター数を示す。
Last Sector	スライスの最後のセクターを示す。
Mount Directory	ファイルシステムの最後のマウントポイントのディレクトリを示す。

ディスクをスライスに分割する

format ユーティリティは、主にシステム管理者がディスクをスライスに分割する場合に使われます。手順を次に示します。

- どのスライスが必要かを決定する
- 各スライスのサイズを決定する
- format ユーティリティを使用してディスクをスライスに分割する
- 新しいスライス情報を使用してディスクにラベルを付ける
- スライスごとにファイルシステムを作成する

ディスクをスライスに分割するには、format ユーティリティの `partition` メニューで `modify` コマンドを実行するのが最も簡単です。modify コマンドを使用すると、開始シリンダ境界を追跡しなくても、各スライスのサイズを指定してスライスを作成できます。modify コマンドを使用すると、「free hog」スライス内の残りのディスク領域も追跡できます。

free hog スライスの使用方法

format ユーティリティを使用してディスクスライスのサイズを変更するときには、サイズ変更操作に対応して拡大縮小する一時スライスを指定します。

このスライスは、スライスを拡大すると領域を「解放 (free)」し、スライスを圧縮すると放棄された領域を「回収 (hog)」します。このため、提供側のスライスを「free hog」と呼びます。

free hog スライスは、インストール時または format ユーティリティの実行時にのみ存在します。日常の操作中に free hog スライスが継続して存在することはありません。

free hog スライスの使用方法については、503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」または 521 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

第 33 章

ディスクの管理 (手順)

この章では、ディスク管理の手順について説明します。Solaris システム上でディスクを管理する方法に精通している場合は、この章で説明する多くの内容を読み飛ばすことができます。

ディスク管理に関連した手順の詳細については、475 ページの「ディスクの管理 (作業マップ)」を参照してください。

ディスク管理の概要については、第 32 章を参照してください。

ディスクの管理 (作業マップ)

作業	説明	参照先
システム上のディスクの確認	システム上のディスクの種類が分からない場合は、 <code>format</code> ユーティリティを使用して確認する。	477 ページの「システム上のディスクを確認する方法」
ディスクのフォーマット	<code>format</code> ユーティリティを使用して、ディスクがフォーマット済みかどうかを判断する。 ほとんどの場合、ディスクはフォーマット済みである。フォーマットする必要がある場合は、 <code>format</code> を使用する。	479 ページの「ディスクがフォーマット済みかを調べる方法」 480 ページの「ディスクをフォーマットする方法」

作業	説明	参照先
スライス情報の表示	format ユーティリティを使用してスライス情報を表示する。	481 ページの「ディスクスライス情報を表示する方法」
ディスクラベルの作成	format ユーティリティを使用してディスクラベルを作成する。	483 ページの「ディスクラベルを作成する方法」
ディスクラベルの検査	prtvtoc コマンドを使用してディスクラベルを検査する。	486 ページの「ディスクラベルを検査する方法」
破損したディスクラベルの復元	システム障害または電源障害のために破損したディスクラベルの復元を試みる。	487 ページの「破損したディスクラベルを復元する方法」
format.dat のエントリの作成	サードパーティのディスクをサポートするために format.dat のエントリを作成する。	490 ページの「format.dat のエントリを作成する方法」
SCSI ディスクの自動構成	特定のドライブタイプが /etc/format.dat ファイルに記載されていない場合でも、ディスクデバイスモードセンスページの SCSI-2 仕様を利用して、SCSI ディスクを自動構成できる。	492 ページの「SCSI ドライブを自動構成する方法」
欠陥ディスクセクターの検出	format ユーティリティを使用して欠陥ディスクセクターを調べる。	494 ページの「表面解析を使用して欠陥セクターを調べる方法」
欠陥ディスクセクターの修復 (必要な場合)	format ユーティリティを使用して欠陥ディスクセクターを修復する。	495 ページの「欠陥セクターを修復する方法」

システム上のディスクの確認

format ユーティリティを使用して、システムに接続されているディスクの種類を調べます。また、format ユーティリティを使用して、ディスクがシステムに認識されるかどうかを検査することもできます。format ユーティリティの使用方法については、第 36 章を参照してください。

▼ システム上のディスクを確認する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを使用して、システム上で認識されるディスクを確認します。

```
# format
```

format ユーティリティは、AVAILABLE DISK SELECTIONS という見出しの下に、認識されるディスクのリストを表示します。

例 — システム上のディスクを確認する

次の **format** 出力は、単一のディスクを持つシステムのもので、

```
# format
Searching for disks...done
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0t0d0 <ST34321A cyl 8892 alt 2 hd 15 sec 63>
   /pci@1f,0/pci@1,1/ide@3/dad@0,0
```

```
Specify disk (enter its number):
```

format の出力は、ディスクの物理デバイス名と論理デバイス名を括弧 <> 内の商品名に対応させています。次の例を参照してください。この方法では、どの論理デバイス名がシステムに接続されたディスクを表しているかをすぐに識別できます。論理デバイス名と物理デバイス名については、第 30 章を参照してください。

次の例では、ワイルドカードを使用して、追加コントローラに接続されたディスクを表示します。

```
# format /dev/rdisk/c2*
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. /dev/rdisk/c2t10d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
   /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
1. /dev/rdisk/c2t11d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
   /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
2. /dev/rdisk/c2t14d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@3,0/SUNW,fas@3,8800000/sd@e,0
3. /dev/rdisk/c2t15d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@3,0/SUNW,fas@3,8800000/sd@f,0
```

```
Specify disk (enter its number):
```

次の例では、SPARC システム上のディスクを表示します。

```
# format
```

```
0. c0t3d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
   /iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0
```

```
Specify disk (enter its number):
```

format の出力は、ディスク 0 (ターゲット 3) が第 1 の SCSI ホストアダプタ (espdma@...) に接続されており、そのホストアダプタは第 1 の SBus デバイス (sbus@0...) に接続されていることを示しています。また、この出力は物理デバイス名と論理デバイス名をディスクの商品名 SUN2.1G に対応しています。

次の例では、x86 システム上のディスクを確認する方法を示します。

```
# format
AVAILABLE DISK SELECTIONS:
 0. c0d0 <DEFAULT cyl 615 alt 2 hd 64 sec 63>
    /pci@0,0/pci-ide@7,1/ata@0/cmdk@0,0
 1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
    /pci@0,0/pci-ide@7,1/ata@0/cmdk@1,0
 2. c1d0 <DEFAULT cyl 817 alt 2 hd 256 sec 63>
    /pci@0,0/pci-ide@7,1/ata@1/cmdk@0,0
Specify disk (enter its number):
```

format の出力は、ディスク 0 が最初の PCI ホストアダプタ (pci-ide@7..) に接続され、このアダプタが ATA デバイス (ata...) に接続されていることを示しています。x86 システム上での format の出力には、ディスクは商品名では表示されません。

次に進む手順

format ユーティリティでディスクが認識されなかった場合は、次の表を参照してください。

ディスクの状態	参照先
ディスクを新しく追加したが、再構成ブートを実行しなかった	第 34 章または第 35 章
Sun 以外のディスク	490 ページの「format.dat のエントリの作成」を参照
電源障害など、システムの問題によってラベルが破損した	483 ページの「ディスクラベルを作成する方法」を参照
ディスクがシステムに正しく接続されていない	ディスクのハードウェアマニュアルを参照して、ディスクをシステムに接続する

ディスクのフォーマット

ディスクはメーカーまたは再販業者によってフォーマットされています。通常は再フォーマットしなくてもドライブを取り付けることができます。

次の作業の前にディスクをフォーマットしておかなければなりません。

- データの書き込み。ただし、ほとんどのディスクはフォーマット済みです。
- Solaris インストールプログラムを使用して行うシステムのインストール。



注意 - フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常はメーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、`format` ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。

▼ ディスクがフォーマット済みかを調べる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. `format` ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、チェックするディスクの番号を入力します。

```
Specify disk (enter its number): 0
```

4. ディスクがフォーマット済みかを調べます。選択したディスクがフォーマット済みであれば、次のメッセージが表示されます。

```
[disk formatted]
```

例 — ディスクがフォーマット済みかを調べる

次の例は、ディスク `c1t0d0` がフォーマット済みであることを示しています。

```
# format /dev/rdisk/c1*
AVAILABLE DISK SELECTIONS:
 0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@0,0
 1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@1,0
 2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@8,0
 3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting /dev/rdisk/c1t0d0s0
[disk formatted]
```

▼ ディスクをフォーマットする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、フォーマットするディスクの番号を入力します。

```
Specify disk (enter its number): 0
```



注意 - システムディスクを選択しないでください。システムディスクをフォーマットすると、オペレーティングシステムや、システムディスク上のデータがすべて削除されます。

4. ディスクのフォーマットを開始するには、**format>** プロンプトで **format** と入力します。y と入力してコマンドを確認します。

```
format> format
Ready to format. Formatting cannot be interrupted
and takes 26 minutes (estimated). Continue? y
```

5. フォーマットが正常に行われたことを、次のメッセージによって確認します。

```
Beginning format. The current time Tue ABC xx xx:xx:xx xxxx

Formatting...
done

Verifying media...
    pass 0 - pattern = 0xc6dec6de
    2035/12/18

    pass 1 - pattern = 0x6db6db6d
    2035/12/18

Total of 0 defective blocks repaired.
```

例 — ディスクをフォーマットする

次の例は、ディスク c0t3d0 のフォーマット方法を示します。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
    /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
 1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
    /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
```



```
Specify disk (enter its number):1
Selecting c0t3d0
[disk formatted]
format> format
Ready to format. Formatting cannot be interrupted
and takes 23 minutes (estimated). Continue? yes
Beginning format. The current time is Thu Dec 6 09:54:40 2001
Formatting ...
done
Verifying media...
    pass 0 - pattern = 0xc6dec6de
    2035/12/18

    pass 1 - pattern = 0xdb6db6d
    2035/12/18

Total of 0 defective blocks repaired.
format>
```

ディスクスライスの表示

format ユーティリティを使用すると、ディスクに適切なディスクスライスがあるかどうかをチェックできます。使用したいスライスがディスクに入っていないことが判明した場合は、format ユーティリティを使用してスライスを作成し直し、ディスクにラベルを付けます。ディスクスライスの作成方法については、503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」または 521 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

注 - format ユーティリティでは、「スライス」ではなく「パーティション」という用語を使用します。

▼ ディスクスライス情報を表示する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを起動します。

```
# format
```
3. 画面に表示されたリストから、スライス情報を表示するディスクの番号を入力します。

```
Specify disk (enter its number):1
```
4. パーティションメニューを選択します。

```
format> partition
```

- 現在のディスクドライブのスライス情報を表示します。

```
partition> print
```

- format** ユーティリティを終了します。

```
partition> q  
format> q  
#
```

- 特定のスライスのタグとサイズについてスライス情報が表示されることを確認します。

画面の出力に、スライスサイズが割り当てられていないことが示された場合は、ディスクにスライスがないものと思われます。

例 — ディスクスライス情報を表示する

次に示すのは、VTOC ラベル付きディスクのスライス情報を表示する例です。

```
# format  
Searching for disks...done  
Specify disk (enter its number):1  
Selecting c0t0d0  
format> partition  
partition> print  
Current partition table (original):  
Total disk cylinders available: 8892 + 2 (reserved cylinders)  
  
Part      Tag      Flag      Cylinders      Size      Blocks  
  0      root     wm      1110 - 4687     1.61GB    (0/3578/0) 3381210  
  1      swap     wu         0 - 1109     512.00MB  (0/1110/0) 1048950  
  2      backup   wm         0 - 8891     4.01GB    (0/8892/0) 8402940  
  3 unassigned wm         0              0          (0/0/0)      0  
  4 unassigned wm         0              0          (0/0/0)      0  
  5 unassigned wm         0              0          (0/0/0)      0  
  6 unassigned wm         0              0          (0/0/0)      0  
  7      home     wm      4688 - 8891     1.89GB    (0/4204/0) 3972780  
partition> q  
format> q  
#
```

これらの例に表示されるスライス情報についての説明は、第 32 章を参照してください。

次に示すのは、EFI ラベル付きディスクのスライス情報を表示する例です。

```
# format  
Searching for disks...done  
Specify disk (enter its number): 9  
selecting c4t1d0  
[disk formatted]
```

```

format> partition
partition> print
Current partition table (original):
partition> q
format> q

```

Part	Tag	Flag	First Sector	Size	Last Sector
0	root	wm	34	300.00GB	629145633
1	usr	wm	629145634	300.00GB	1258291233
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	usr	wm	1258291234	628.77GB	2576924637
8	reserved	wm	2576924638	8.00MB	2576941021

ディスクラベルの作成と検査

一般に、ディスクにラベルを付ける操作は、システムのインストール時、または新しいディスクスライスを作成するときに行います。電源障害などが原因でディスクラベルが破損した場合は、ディスクラベルを作成し直さなければならないことがあります。

format ユーティリティは、ラベルが付いていない SCSI ディスクを自動構成しようとします。format ユーティリティがラベルの付いていないディスクを自動構成できる場合、次のようなメッセージが表示されます。

```
c0t0d1: configured with capacity of 4.00GB
```

ヒント - 複数のディスクに同じディスクラベルを付ける方法については、496 ページの「prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける」を参照してください。

▼ ディスクラベルを作成する方法

次の手順に従って、ディスクに VTOC ラベルを付けることができます。または、サイズが 1T バイトを超えるディスクに EFI ラベルを付けることができます。サイズが 1T バイトに満たないディスクに EFI ラベルを付ける場合は、485 ページの「例 — サイズが 1T バイトに満たないディスクに EFI ラベルを付ける」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、ラベルを作成したいディスクの番号を入力します。

```
Specify disk (enter its number):1
```

4. 次のいずれかを選択します。

- a. ディスクにラベルが付いておらず、正常に構成された場合、手順 5 に進みラベルを付けます。

format ユーティリティにより、ディスクラベルを付けるかを尋ねるプロンプトが表示されます。

- b. ディスクにラベルが付いており、そのタイプを変更する場合、または **format** ユーティリティでディスクを自動構成できなかった場合、手順 6 と 7 を実行してディスクタイプを設定し、ラベルを付けます。

5. **Label it now?** プロンプトで **y** と入力して、ディスクにラベルを付けます。

```
Disk not labeled. Label it now? y
```

これでディスクラベルが作成されました。手順 10 に進んで **format** ユーティリティを終了します。

6. **format>** プロンプトで **type** と入力します。

```
format> type
```

Available Drive Types メニューが表示されます。

7. ディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number)[12]: 12
```

または、0 を選択して SCSI-2 ディスクを自動構成します。詳細については、492 ページの「SCSI ドライブを自動構成する方法」を参照してください。

8. ディスクにラベルを付けます。ディスクにラベルが付いていない場合は、次のメッセージが表示されます。

```
Disk not labeled. Label it now? y
```

ディスクラベルが付いている場合は、次のメッセージが表示されます。

```
Ready to label disk, continue? y
```

9. ディスクラベルを検査します。

```
format> verify
```

10. **format** ユーティリティを終了します。

```
partition> q
```

```
format> q
```

```
#
```

例 — ディスクラベルを作成する

次の例では、1.05G バイトのディスクを自動構成してラベルを付ける方法を示します。

```
# format
   c1t0d0: configured with capacity of 1002.09MB

AVAILABLE DISK SELECTIONS:
   0. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
   1. c1t0d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
Specify disk (enter its number): 1
Disk not labeled. Label it now? yes
format> verify
#
```

例 — サイズが 1T バイトに満たないディスクに EFI ラベルを付ける

次に、`format -e` コマンドを使って、サイズが 1T バイトに満たないディスクに EFI ラベルを付ける例を示します。階層化されたソフトウェア製品が EFI ラベル付きディスクのシステムでも動作することを確認しておいてください。EFI ラベルの一般的な制限事項については、457 ページの「EFI ディスクラベルの制限」を参照してください。

```
# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
   1. c1t0d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
      /sbus@2,0/QLGC,isp@2,10000/sd@0,0
   2. c1t1d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
      /sbus@2,0/QLGC,isp@2,10000/sd@1,0
   3. c1t8d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
      /sbus@2,0/QLGC,isp@2,10000/sd@8,0
   4. c1t9d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
      /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 4
selecting c1t9d0
[disk formatted]
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 1
Ready to label disk, continue? yes
format> quit
```

▼ ディスクラベルを検査する方法

ディスクラベル情報の検査には、`prtvtoc` コマンドを使用します。ディスクラベルの説明と `prtvtoc` コマンドで表示される情報については、第 32 章を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ディスクラベル情報を表示します。

```
# prtvtoc /dev/rdisk/device-name
```

`device-name` には、検査する raw ディスクデバイスを指定してください。

例 — ディスクラベルを検査する

次に示すのは、VTOC ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdisk/c0t0d0s0
* /dev/rdisk/c0t0d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   63 sectors/track
*   15 tracks/cylinder
*   945 sectors/cylinder
*   8894 cylinders
*   8892 accessible cylinders
*
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition  Tag  Flags      First      Sector      Last
*           Count  Sector
* 0          2    00    1048950    3381210    4430159    /
* 1          3    01           0    1048950    1048949
* 2          5    00           0    8402940    8402939
* 7          8    00    4430160    3972780    8402939    /export/home
```

次に示すのは、EFI ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdisk/c3t1d0s0
* /dev/rdisk/c3t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
* 2479267840 sectors
* 2479267773 accessible sectors
*
* Flags:
*   1: unmountable
*   10: read-only
```

```

*
*
* Partition Tag Flags First Sector Last Sector Mount Directory
0 2 00 34 262144 262177
1 3 01 262178 262144 524321
6 4 00 524322 2478727100 2479251421
8 11 00 2479251422 16384 2479267805

```

破損したディスクラベルの復元

電源障害やシステム障害が原因で、ディスクが認識されなくなることがあります。このような場合に、必ずしもスライス情報やディスクのデータを作成し直したり、復元しなければならないとは限りません。

破損したディスクラベルを復元する作業の最初の手順は、正しいジオメトリとディスクタイプ情報を使用してディスクにラベルを付けることです。この作業は、通常のディスクラベル作成方法 (自動構成またはディスクタイプの手動指定) で実行できません。

`format` ユーティリティでディスクタイプが認識されたら、次の手順はバックアップラベルを検索してディスクにラベルを付けることです。バックアップラベルを使用してディスクにラベルを付けると、ディスクタイプとジオメトリだけでなく、正しいパーティション情報を使用してディスクラベルが作成されます。

▼ 破損したディスクラベルを復元する方法

1. システムをシングルユーザーモードにします。
必要であれば、シングルユーザーモードでローカル CD-ROM またはネットワークからシステムをブートして、ディスクにアクセスします。
システムをブートする方法については、第 13 章または第 14 章を参照してください。

2. ディスクのラベルを作成し直します。

```
# format
```

この時点で、`format` はラベルが付いていない SCSI ディスクを自動構成しようとします。ラベルが付いておらず破損したディスクを自動構成できない場合は、次のメッセージが表示されます。

```
cwtxdy: configured with capacity of abcMB
```

次に、システム上のディスクのリストが表示されます。

3. 画面に表示されたリストから、復元したいディスクの番号を入力します。

```
Specify disk (enter its number): 1
```

4. 次のいずれかを選択して、ディスクラベルの作成方法を決定します。
 - a. ディスクが正常に構成された場合、手順 5 と 6 を実行します。その後、手順 12 に進みます。
 - b. ディスクが正常に構成されなかった場合、手順 7 - 11 を実行します。その後、手順 12 に進みます。
5. バックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk or
use the 'backup' command.
Backup label contents:
Volume name = <>
ascii name = <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
pcyl      = 2038
ncyl      = 2036
acyl      = 2
nhead     = 14
nsect     = 72
Part      Tag      Flag      Cylinders      Size      Blocks
0         root      wm        0 - 300         148.15MB   (301/0/0)  303408
1         swap      wu        301 - 524       110.25MB   (224/0/0)  225792
2         backup   wm        0 - 2035       1002.09MB  (2036/0/0) 2052288
3 unassigned wm         0              0          (0/0/0)    0
4 unassigned wm         0              0          (0/0/0)    0
5 unassigned wm         0              0          (0/0/0)    0
6         usr      wm        525 - 2035     743.70MB   (1511/0/0) 1523088
7 unassigned wm         0              0          (0/0/0)    0
```

6. **format** ユーティリティがバックアップラベルを検出することができ、その内容が適切な場合は、**backup** コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
これで、ディスクラベルが復元されました。手順 12 へ進みます。
```

7. **format** でディスクを自動構成できなかった場合は、**type** コマンドを使用してディスクタイプを指定します。

```
format> type
Available Drives Type メニューが表示されます。
```

8. 0 を選択してディスクを自動構成するか、またはディスクタイプの候補のリストからディスクタイプを選択します。


```
Specify disk type (enter its number) [12]: 12
```

9. ディスクが正常に自動構成された場合は、ディスクラベルを作成するかどうか尋ねるプロンプトが表示されたときに **no** と応答します。

```
Disk not labeled. Label it now? no
```

10. **verify** コマンドを使用してバックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
.
.
.
```

11. **format** ユーティリティがバックアップラベルを検出することができ、その内容が適切な場合は、**backup** コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
これで、ディスクラベルが復元されました。
```

12. **format** ユーティリティを終了します。

```
format> q
```

13. **fsck** コマンドを使用して、復元されたディスク上のファイルシステムを確認します。

fsck コマンドの使用方法については、第 43 章を参照してください。

Sun 製品以外のディスクの追加

Solaris 環境では、Sun 製品以外の多数のディスクがサポートされます。ただし、ディスクを認識させるためには、デバイスドライバか、`format.dat` エントリのいずれか 1 つ、またはその両方を用意しなければならない場合があります。ディスク追加時のその他のオプションを、次に示します。

- SCSI ディスクを追加する場合、**format** ユーティリティの自動構成機能の使用を試みることができます。詳細については、491 ページの「SCSI ディスクドライブの自動構成」を参照してください。
- PCI、SCSI、または USB ディスクのホットプラグを試みることもできます。詳細については、第 27 章を参照してください。

Sun 以外のディスクが標準の SunOS 互換デバイスドライバで機能するように設計されている場合は、適切な `format.dat` エントリを作成するだけで、ディスクは `format` ユーティリティに認識されるはずですが、それ以外の場合は、そのディスクをサポートするために Sun 以外のデバイスドライバをロードする必要があります。

注 - Sun の `format` ユーティリティが Sun 以外の製品のどのディスクドライバでも正常に機能するとは限りません。ディスクドライバに Solaris の `format` ユーティリティとの互換性がない場合は、ディスクドライブのベンダーが独自のフォーマットプログラムを提供しているはずですが。

ここでは、ソフトウェアサポートのいずれかが不足している場合に必要な作業について説明します。一般に、`format` ユーティリティを起動し、ディスクタイプが認識されないなどという場合に、不足しているソフトウェアサポートがあることがわかります。

この節の説明にしたがって不足しているソフトウェアを用意してから、第 34 章または第 35 章で説明されている、システムディスクまたは二次ディスクを構成する手順を参照してください。

format.dat のエントリの作成

認識されないディスクは、そのディスクのジオメトリと運用パラメータに関する正確な情報がなければフォーマットできません。この情報は、`/etc/format.dat` ファイル内で指定します。

注 - SCSI-2 ドライブには `format.dat` のエントリは不要です。再構成ブート時にドライブに電源が投入されていれば、`format` ユーティリティは SCSI-2 ドライブを自動的に構成します。SCSI ディスクドライブを自動構成する手順については、492 ページの「SCSI ドライブを自動構成する方法」を参照してください。

ディスクが認識されない場合は、テキストエディタを使用して `format.dat` にディスクのエントリを作成します。作業を始める前に、ディスクとそのコントローラに関連するすべての技術仕様を収集する必要があります。この情報はディスクと一緒に提供されているはずですが、提供されない場合は、ディスクメーカーまたは購入先に問い合わせてください。

▼ format.dat のエントリを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. `/etc/format.dat` ファイルのコピーを作成します。

```
# cp /etc/format.dat /etc/format.dat.gen
```

- 第 36 章で説明する `format.dat` 情報を使用して、`/etc/format.dat` ファイルに、Sun 以外の製品のディスクのエントリを入力します。
ディスクのハードウェア製品マニュアルを参照して、必要な情報を収集してください。

SCSI ディスクドライブの自動構成

`/etc/format.dat` ファイルに特定のドライブタイプが含まれていない場合でも、`format` ユーティリティは SCSI ディスクドライブを自動的に構成します。この機能によって、ディスクデバイスモード検知に関する SCSI-2 仕様に準拠しているディスクドライブで、フォーマット、スライス作成、およびラベル作成を行うことができます。

ディスクの追加に関するその他のオプションを、次に示します。

- SCSI ディスクを追加する場合、`format` ユーティリティの自動構成機能の使用を試みることができます。詳細については、491 ページの「SCSI ディスクドライブの自動構成」を参照してください。
- PCI、SCSI、または USB ディスクのホットプラグを試みることもできます。詳細については、第 27 章を参照してください。

自動構成を使用して SCSI ドライブを構成する場合、次の手順を実行します。

- システムをシャットダウンする。
- SCSI ディスクドライブをシステムに接続する。
- ディスクドライブの電源をオンにする。
- 再構成用ブートを実行する。
- `format` ユーティリティを使用して SCSI ディスクドライブを自動構成する。

再構成ブートを実行した後に、`format` ユーティリティを呼び出すと、`format` はディスクを構成しようとします。成功すると、ディスクが構成されたことを示すメッセージを表示します。SCSI ディスクドライブを自動構成する手順については、492 ページの「SCSI ドライブを自動構成する方法」を参照してください。

1.3G バイトの SCSI ディスクドライブに関して、`format` ユーティリティが表示するパーティションテーブルの例を、次に示します。

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 96	64.41MB	(97/0/0)
1	swap	wu	97 - 289	128.16MB	(193/0/0)
2	backup	wu	0 - 1964	1.27GB	(1965/0/0)
6	usr	wm	290 - 1964	1.09GB	(1675/0/0)

SCSI 自動構成機能の使用方法については、第 36 章を参照してください。

▼ SCSI ドライブを自動構成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. システムのブート時に読み込まれる **/reconfigure** ファイルを作成します。

```
# touch /reconfigure
```

3. システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0 システムを init レベル 0 (電源切断) にする。

-gn ログインしているユーザーに、*n* 秒後にシステムのシャットダウンを開始することを通知する。

-y ユーザーの介入なしでコマンドを実行するように指定する。

システムのシャットダウン後に、ok というプロンプトが表示されます。

4. システムとすべての外部周辺デバイスの電源を切ります。
5. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの裏側にそのための小型のスイッチが付いています。
6. ディスクがシステムに正しく接続されているかどうかを確認します。
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
7. すべての外部周辺デバイスの電源を入れます。
8. システムの電源を入れます。
システムがブートし、ログインプロンプトが表示されます。
9. スーパーユーザーになるか、同等の役割を引き受けて、再度ログインします。
10. **format** ユーティリティを起動して、自動構成するディスクを選択します。

```
# format
Searching for disks...done
c1t0d0: configured with capacity of 1002.09MB
AVAILABLE DISK SELECTIONS:
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 1
```

11. プロンプトで **y** と入力してディスクにラベルを付けます。

y と入力すると、SCSI 自動構成機能により、ディスクラベルの生成およびディスクへの書き込みが実行されます。

```
Disk not labeled. Label it now? y
```

12. ディスクラベルを検査します。

```
format> verify
```

13. format ユーティリティを終了します。

```
format> q
```

欠陥セクターの修復

システム上のディスクに欠陥セクターが存在する場合は、次の手順にしたがって修復できます。欠陥セクターを発見するのは次のような場合です。

- ディスク上で表面解析を実行した場合

format ユーティリティの解析機能については、530 ページの「analyze メニュー」を参照してください。

システムの実行中にレポートされる欠陥領域は正確ではない場合があります。システムは一度に多数のセクターでディスク処理を実行するので、通常は、どのセクターが所定のエラーの原因かを正確に突き止めるのは困難です。正確なセクターを検出するには、494 ページの「表面解析を使用して欠陥セクターを調べる方法」を参照してください。

- システムの実行中に、ディスクドライバからディスクの特定部分に関して多数のエラーメッセージが表示される場合

ディスクエラーに関連するメッセージは次のよう出力されます。

```
WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):  
  Error for command 'read' Error Level: Retryable  
  Requested Block 126, Error Block: 179  
  Sense Key: Media Error  
  Vendor 'name':  
  ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
```

上記のコンソールメッセージは、ブロック 179 が不良であることを示しています。format ユーティリティの repair コマンドを使用して欠陥ブロックを配置し直すか、修復オプションを有効にして analyze コマンドを実行してください。

▼ 表面解析を使用して欠陥セクターを調べる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 欠陥セクターの存在するスライス内のファイルシステムをマウント解除します。

```
# umount /dev/dsk/device-name
```

詳細については、mount (1M) のマニュアルページを参照してください。

3. **format** ユーティリティを起動します。

```
# format
```

4. 調べるディスクを選択します。

```
Specify disk (enter its number):1
selecting c0t2d0:
[disk formatted]
Warning: Current Disk has mounted partitions.
```

5. **analyze** メニューを選択します。

```
format> analyze
```

6. **analyze>** プロンプトで **setup** と入力して、検索手順に使用する解析パラメータを設定します。

次のパラメータを使用してください。

```
analyze> setup
Analyze entire disk [yes]? n
Enter starting block number [0, 0/0/0]: 12330
Enter ending block number [2052287, 2035/13/71]: 12360
Loop continuously [no]? y
Repair defective blocks [yes]? n
Stop after first error [no]? n
Use random bit patterns [no]? n
Enter number of blocks per transfer [126, 0/1/54]: 1
Verify media after formatting [yes]? y
Enable extended messages [no]? n
Restore defect list [yes]? y
Create defect label [yes]? y
```

7. **read** コマンドを使用して欠陥を見つけます。

```
analyze> read
Ready to analyze (won't harm SunOS). This takes a long time,
but is interruptible with Control-C. Continue? y
    pass 0
      2035/12/1825/7/24
    pass 1
Block 12354 (18/4/18), Corrected media error (hard data ecc)
      25/7/24
^C
Total of 1 defective blocks repaired.
```

▼ 欠陥セクターを修復する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを起動します。

```
# format
```

3. 欠陥セクターの存在するディスクを選択します。

```
Specify disk (enter its number): 1
selecting c0t3d0
[disk formatted]
format>
```

4. **repair** コマンドを選択します。

```
format> repair
```

5. 欠陥ブロック番号を入力します。

```
Enter absolute block number of defect: 12354
Ready to repair defect, continue? y
Repairing block 12354 (18/4/18)...ok.
format>
```

format を使って欠陥セクターを調べる方法については、494 ページの「表面解析を使用して欠陥セクターを調べる方法」を参照してください。

ディスク管理のヒント

次のヒントに従って、ディスクの管理効率を高めることができます。

format セッションのデバッグ

format -M と入力して、ATA および SCSI デバイス用の拡張および診断メッセージを有効にします。

この例の **Inquiry:** の下の数値の列は、数値の右側に表示される **inquiry** データの 16 進数値です。

```
# format -M
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
    /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
 1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
```

```
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
```

```
Specify disk (enter its number): 0
selecting c0t3d0
[disk formatted]
format> inquiry
Inquiry:
00 00 02 02 8f 00 00 12 53 45 41 47 41 54 45 20      .....NAME....
53 54 31 31 32 30 30 4e 20 53 55 4e 31 2e 30 35      ST11200N SUN1.05
38 33 35 38 30 30 30 33 30 32 30 39 00 00 00 00      835800030209....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 43 6f 70 79 72 69 67 68 74 20 28 63 29 20 31      .Copyright (c) 1
39 39 32 20 53 65 61 67 61 74 65 20 41 6c 6c 20      992 NAME All
72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 20      rights reserved
30 30 30                                               000
Vendor:   name
Product:  ST11200N SUN1.05
Revision: 8358
format>
```

prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける

prtvtoc コマンドと fmthard コマンドを使用して、同じディスクジオメトリを持つ複数のディスクにラベルを付けます。

この for ループをスクリプト内で使用して、1 台のディスクからディスクラベルをコピーし、複数のディスク上で複製します。

```
# for i in xyz
> do
> prtvtoc /dev/rdisk/cwtxdysz | fmthard -s - /dev/rdisk/cwt${i}d0s2
> done
```

例 — 複数のディスクにラベルを付ける

この例では、ディスクラベルがディスク c2t0d0s0 から他の 4 台のディスクにコピーされます。

```
# for i in 1 2 3 5
> do
> prtvtoc /dev/rdisk/c2t0d0s0 | fmthard -s - /dev/rdisk/c2t${i}d0s2
> done
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
```


#

第 34 章

SPARC: ディスクの追加 (手順)

この章では、SPARC システムにディスクを追加する手順について説明します。

SPARC システムへのディスクの追加に関連した手順については、499 ページの「SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

ディスク管理の概要については、第 32 章を参照してください。x86 システムにディスクを追加する手順については、第 35 章を参照してください。

SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)

次の作業マップは、SPARC システムにディスクを追加する手順を示します。

作業	説明	参照先
1. ディスクの接続とブート	システムディスク 新しいディスクを接続して、ローカルまたはリモートの Solaris CD または DVD からブートする。	501 ページの「SPARC: システムディスクを接続してブートする方法」

作業	説明	参照先
	二次ディスク 新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行する。	502 ページの「SPARC: 二次ディスクを接続してブートする方法」
2. スライスとディスクラベルの作成	ディスクスライスを作成してディスクにラベルを付ける (ディスクメーカーにより実行されていない場合)。	503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」
3. ファイルシステムの作成	<code>newfs</code> コマンドを使用してディスクスライス上に UFS ファイルシステムを作成する。システムディスクの場合はルート (/) または <code>/usr</code> ファイルシステム、あるいはその両方を作成しなければならない。	508 ページの「SPARC: ファイルシステムを作成する方法」
4. ファイルシステムの復元	システムディスク上にルート (/) または <code>/usr</code> ファイルシステム、あるいはその両方を復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。	第 49 章
5. ブートブロックのインストール	システムディスクのみ。システムをブートできるように、ルート (/) ファイルシステムにブートブロックをインストールする。	508 ページの「SPARC: システムディスクにブートブロックをインストールする方法」

SPARC: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと、`/usr` ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元する。

二次ディスクには、ルート (/) ファイルシステムも /usr ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。

▼ SPARC: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

1. 損傷したシステムディスクをシステムから外します。
2. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
3. 交換用のディスクをシステムに正しく接続します。
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
4. ローカルの **Solaris CD/DVD** またはリモートの **Solaris CD/DVD** (ネットワーク経由) のどちらでブートするかに応じて、次の表の手順で操作します。

ブート方法	作業
ローカルドライブの Solaris CD または DVD	1. CD または DVD がドライブに入っていることを確認する。 2. メディアからシングルユーザーモードでブートする。 <code>ok boot cdrom -s</code>
ネットワーク経由	ネットワークからシングルユーザーモードでブートする。 <code>ok boot net -s</code>

数分後に、スーパーユーザーのプロンプト (#) が表示されます。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ SPARC: 二次ディスクを接続してブートする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ディスクタイプが **Solaris** ソフトウェアでサポートされていない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。

ディスク用の `format.dat` エントリを作成する方法については、490 ページの「`format.dat` のエントリを作成する方法」を参照してください。

3. システムのブート時に読み込まれる `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

`/reconfigure` ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0	システムを実行レベル 0 (電源切断状態) にする。
-gn	ログインしているユーザーに、 <i>n</i> 秒後にシステムのシャットダウンを開始することを通知する。
-y	ユーザーの介入なしでコマンドを実行するように指定する。

オペレーティング環境のシャットダウン後に、`ok` プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
7. ディスクがシステムに正しく接続されているかどうかを確認します。
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。
システムがブートし、ログインプロンプトが表示されます。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。503 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ SPARC: ディスクスライスとディスクラベルを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

利用可能なディスクのリストが表示されます。詳細については、`format (1M)` のマニュアルページを参照してください。

3. 画面に表示されるリストから、パーティション分割をし直したいディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number は、パーティションの再分割を実行するディスクの番号です。

4. **partition** メニューを選択します。

```
format> partition
```

5. 現在のパーティション (スライス) テーブルを表示します。

```
partition> print
```

6. 変更作業を開始します。

```
partition> modify
```

7. ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]? 1
```

free hog スライスの詳細については、474 ページの「**free hog** スライス の使用方法」を参照してください。

8. 続行するかどうかを尋ねるプロンプトが表示されたら、**y** と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition table based on  
above table[yes]? y
```

9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0) およびスワップ (スライス 1) (必須)
- /usr (スライス 6) (省略可能)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。ディスクスライスの作成例については、504 ページの「SPARC: 例 — システムディスクのディスクスライスとラベルを作成する」を参照してください。

10. プロンプトが表示されたら **y** と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? y
```

現在のパーティションテーブルが希望どおりでないために変更したい場合は、no と応答して手順 6 に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

partition-name には、新しいパーティションテーブルの名前を指定します。

12. 新しいディスク上でスライスを割り当て終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. **partition** メニューを終了します。

```
partition> q
```

14. ディスクラベルを検査します。

```
format> verify
```

15. **format** メニューを終了します。

```
format> q
```

SPARC: 例 — システムディスクのディスクスライスとラベルを作成する

次の例では、**format** ユーティリティを使用して 18G バイトのディスクを 3 つのスライスに分割します。各スライスをルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムに割り当てます。

```
# format
AVAILABLE DISK SELECTIONS:
  0. /dev/rdisk/clt0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@0,0
  1. /dev/rdisk/clt1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@1,0
  2. /dev/rdisk/clt8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@8,0
  3. /dev/rdisk/clt9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```



```

/sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting clt0d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  0. Current partition table (original)
  1. All Free Hog
Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0              0      (0/0/0)      0
  1      swap      wu         0              0      (0/0/0)      0
  2      backup    wu        0 - 7505      16.86GB   (7506/0/0) 35368272
  3 unassigned  wm         0              0      (0/0/0)      0
  4 unassigned  wm         0              0      (0/0/0)      0
  5 unassigned  wm         0              0      (0/0/0)      0
  6      usr       wm         0              0      (0/0/0)      0
  7 unassigned  wm         0              0      (0/0/0)      0

Choose base (enter number) [0]? 1
table based on above table[yes]? yes
Free Hog partition[6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:
Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0 - 1780      4.00GB   (1781/0/0) 8392072
  1      swap      wu       1781 - 3561   4.00GB   (1781/0/0) 8392072
  2      backup    wu        0 - 7505      16.86GB   (7506/0/0) 35368272
  3 unassigned  wm         0              0      (0/0/0)      0
  4 unassigned  wm         0              0      (0/0/0)      0
  5 unassigned  wm         0              0      (0/0/0)      0
  6      usr       wm       3562 - 7505   8.86GB   (3944/0/0) 18584128
  7 unassigned  wm         0              0      (0/0/0)      0

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "disk0"
Ready to label disk, continue? yes
partition> quit
format> verify
format> quit

```

SPARC: 例 — 二次ディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティを使用して 18G バイトのディスクを /export/home ファイルシステム用の 1 つのスライスに割り当てます。

```

# format /dev/rdisk/c1*
AVAILABLE DISK SELECTIONS:
  0. /dev/rdisk/clt0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@0,0
  1. /dev/rdisk/clt1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@1,0
  2. /dev/rdisk/clt8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@8,0
  3. /dev/rdisk/clt9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@9,0

Specify disk (enter its number): 1
selecting clt1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  0. Current partition table (original)
  1. All Free Hog

Choose base (enter number) [0]? 1

```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0	0	(0/0/0) 0
1	swap	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 7505	16.86GB	(7506/0/0) 35368272
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	usr	wm	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

```

Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 7
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:

```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0	0	(0/0/0) 0
1	swap	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 7505	16.86GB	(7506/0/0) 35368272
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	usr	wm	0	0	(0/0/0) 0
7	unassigned	wm	0 - 7505	16.86GB	(7506/0/0) 35368272

```

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "home"
Ready to label disk, continue? y
partition> q
format> verify
format> q
#

```

次の例では、formatユーティリティを使ってEFIラベル付きの1.15Tバイトのディスクを3つのスライスに分割します。

```
# format
.
.
.
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1
Part      Tag      Flag      First Sector      Size      Last Sector
0         root      wm         0                  0          0
1         usr       wm         0                  0          0
2 unassigned wm         0                  0          0
3 unassigned wm         0                  0          0
4 unassigned wm         0                  0          0
5 unassigned wm         0                  0          0
6         usr       wm         0                  0          0
8 reserved wm         2576924638        8.00MB      2576941021
Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition 0 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 1 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 2 [0b, 34e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 3 [0b, 838860834e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 5 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Enter size of partition 6 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Part      Tag      Flag      First Sector      Size      Last Sector
0 unassigned wm         0                  0          0
1 unassigned wm         0                  0          0
2         usr       wm         34                400.00GB   838860833
3         usr       wm         838860834         400.00GB   1677721633
4         usr       wm         1677721634        428.77GB   2576924637
5 unassigned wm         0                  0          0
6 unassigned wm         0                  0          0
8 reserved wm         2576924638        8.00MB      2576941021
Ready to label disk, continue? yes

partition> q
```

次に進む手順

ディスクスライスとディスクラベルを作成し終わったら、ディスク上にファイルシステムを作成できます。508ページの「SPARC: ファイルシステムを作成する方法」を参照してください。

▼ SPARC: ファイルシステムを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```

/dev/rdisk/cwtxdysz は、作成するファイルシステムの raw デバイスです。

newfs コマンドの詳細については、第 39 章または newfs (1M) のマニュアルページを参照してください。

3. マウントして、新規ファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls lost+found
```

SPARC: 次に進む手順

追加するディスク	作業
システムディスク	ディスク上にルート (/) と /usr のファイルシステムを復元する必要がある。第 49 章を参照。 ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールする。508 ページの「SPARC: システムディスクにブートブロックをインストールする方法」を参照。
二次ディスク	新しいディスク上にファイルシステムを復元しなければならないことがある。第 49 章を参照。 新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が終わったことになる。ファイルシステムをユーザーが利用できるようにする方法については、第 40 章を参照。

▼ SPARC: システムディスクにブートブロックをインストールする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 起動ブロックをシステムディスクにインストールします。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/cwtxdys0
```

<code>/usr/platform/`uname -i`/lib/fs/ufs/bootblk</code>	ブートブロックコード
<code>/dev/rdisk/cwtxdys0</code>	ルート (/) ファイルシステムの raw デバイス

詳細については、`installboot (1M)` のマニュアルページを参照してください。

3. システムをリブートし、レベル 3 で実行することによって、ブートブロックがインストールされていることを確認します。

```
# init 6
```

SPARC: 例 — システムディスクにブートブロックをインストールする

次の例では、ブートブロックを Ultra10 システムにインストールする方法を示します。

```
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0
```


第 35 章

x86: ディスクの追加 (手順)

この章では、x86 システムにディスクを追加する手順について説明します。

x86 システムへのディスク追加に関連した手順については、511 ページの「x86: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

ディスク管理の概要については、第 32 章を参照してください。SPARC システムにディスクを追加する手順については、第 34 章を参照してください。

x86: システムディスクまたは二次ディスクの追加 (作業マップ)

作業	説明	参照先
1. ディスクの接続とブート	<p>システムディスク</p> <p>新しいディスクを接続して、ローカルまたはリモートの Solaris CD または DVD からブートする。</p> <p>二次ディスク</p> <p>新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行する。</p>	<p>513 ページの「x86: システムディスクを接続してブートする方法」</p> <p>513 ページの「x86: 二次ディスクを接続してブートする方法」</p>

作業	説明	参照先
2. スライスとディスクラベルの作成	ディスクスライスを作成してディスクにラベルを付ける(ディスクメーカーにより実行されていない場合)。	515 ページの「x86: Solaris fdisk パーティションを作成する方法」および 521 ページの「x86: ディスクスライスとディスクラベルを作成する方法」
3. ファイルシステムの作成	newfs コマンドを使用してディスクスライス上に UFS ファイルシステムを作成する。システムディスクの場合はルート (/) または /usr ファイルシステム (あるいは両方) を作成しなければならない。	523 ページの「x86: ファイルシステムを作成する方法」
4. ファイルシステムの復元	システムディスク上にルート (/) または /usr ファイルシステム (あるいは両方) を復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。	第 49 章
5. ブートブロックのインストール	システムディスクのみ。システムをブートできるように、ルート (/) ファイルシステムにブートブロックをインストールする。	523 ページの「x86: システムディスクにブートブロックをインストールする方法」

x86: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと /usr ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元する。

二次ディスクには、ルート (/) ファイルシステムも /usr ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。

▼ x86: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

1. 損傷したシステムディスクをシステムから外します。
2. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
3. 交換用のディスクをシステムに正しく接続します。
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
4. ローカルとリモートのどちらの **Solaris CD** (または **DVD**) からブートするかに応じて、手順 **a** から **e** を実行します。
ネットワークからブートする場合は、手順 **a** をスキップします。
 - a. ローカルの **Solaris CD** または **DVD** からブートする場合は、ドライブに **Solaris 9 Installation CD** または **DVD** を挿入します。
 - b. **Solaris** ブートフロッピーディスクを主フロッピーディスクドライブ (**DOS** ドライブ **A**) に挿入します。
 - c. **Type any key to continue** プロンプトが表示されたら、任意のキーを押してシステムをリブートします。システムがシャットダウンされている場合は、リセットボタンを押してシステムを再起動します。
数分後に **Boot Solaris** 画面が表示されます。
 - d. **Boot Solaris** 画面から、ブートデバイスを **CD-ROM** ドライブにするか、ネットワークにするかを選択します。
Current Boot Parameters 画面が表示されます。
 - e. システムをシングルユーザーモードでブートします。

```
Select the type of installation: b -s
```


数分後に、スーパーユーザーのプロンプト (#) が表示されます。

x86: 次に進む手順

システムのブート後に、**fdisk** パーティションを作成できます。515 ページの「x86: Solaris **fdisk** パーティションを作成する方法」を参照してください。

▼ x86: 二次ディスクを接続してブートする方法

1. スーパーユーザーになるか、同等の役割を引き受けれます。

2. ディスクが **Solaris** ソフトウェアでサポートされない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。
3. システムのブート時に読み込まれる **/reconfigure** ファイルを作成します。

```
# touch /reconfigure
```

/reconfigure ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0	システムを実行レベル 0 (電源切断状態) にする。
-gn	ログインしているユーザーに、 <i>n</i> 秒後にシステムのシャットダウンを開始することを通知する。
-y	ユーザーの介入なしでコマンドを実行するように指定する。

Type any key to continue プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
7. ディスクがシステムに正しく接続されているかどうかを確認します。
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。
システムがブートし、ログインプロンプトが表示されます。

x86: 次に進む手順

システムのブート後に、fdisk パーティションを作成できます。515 ページの「x86: Solaris fdisk パーティションを作成する方法」を参照してください。

x86: fdisk パーティションの作成上のガイドライン

次のガイドラインに従って fdisk パーティションを設定してください。

- ディスクは最大 4 つの fdisk パーティションに分割できます。いずれか 1 つのパーティションを Solaris パーティションにしなければなりません。
- Solaris パーティションをディスク上でアクティブなパーティションにしなければなりません。アクティブなパーティションとは、システム起動時にデフォルトでオペレーティングシステムがブートされるパーティションです。
- Solaris の fdisk パーティションは、シリンダ境界から開始しなければなりません。
- 最初のディスクの先頭のセクターには、ブート情報 (マスターブートレコードを含む) が書き込まれるので、最初のディスクの 1 番目の fdisk パーティションとして、Solaris の fdisk パーティションを作成する場合は、ディスクのシリンダ 0 ではなくシリンダ 1 から開始しなければなりません。
- Solaris の fdisk パーティションにディスク全体を使用するか、それより小さくして DOS パーティションに使用する余地を残すことができます。また、既存のパーティションに影響を与えずに、ディスク上に新しい fdisk パーティションを作成できます (それを作成する余地がある場合)。

x86 のみ – Solaris スライスとはパーティションと呼ばれることがあります。このマニュアルでは「スライス」という用語を使用しますが、Solaris のマニュアルやプログラムによっては、スライスを「パーティション」と呼ぶ場合があります。

混乱を避けるために、Solaris のマニュアルでは、fdisk パーティション (x86 プラットフォーム版 Solaris でのみサポート) と、スライスやパーティションと呼ばれる Solaris の fdisk パーティションを区別しています。

▼ x86: Solaris fdisk パーティションを作成する方法

1. 515 ページの「x86: fdisk パーティションの作成上のガイドライン」を参照します。
2. スーパーユーザーになるか、同等の役割を引き受けます。
3. **format** ユーティリティを起動します。

```
# format
```

詳細については、format (1M) のマニュアルページを参照してください。

4. 画面に表示されたリストから、**Solaris fdisk** パーティションを作成するディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number は、Solaris fdisk パーティションを作成するディスクの番号です。

5. **fdisk** メニューを選択します。

```
format> fdisk
```

表示される fdisk メニューは、fdisk パーティションがすでにディスク上に存在しているかどうかによって異なります。次の表を使用して、次に行う手順を決定してください。

作業	次の手順	参照先
ディスク全体を占有する Solaris fdisk パーティションを作成する。	手順 6	518 ページの「x86: 例 — ディスク全体を占有する Solaris fdisk パーティションを作成する」
Solaris fdisk パーティションを作成し、既存の Solaris 以外の fdisk パーティションは変更しない。	手順 7	519 ページの「x86: 例 — 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する」
Solaris fdisk パーティションと、Solaris 以外の fdisk パーティションを追加作成する。	手順 7	520 ページの「x86: 例 — Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する」

6. ディスク全体にまたがる **Solaris fdisk** パーティションを作成してそれをアクティブにするには、プロンプトで **y** を入力します。次に、**手順 14** に進みます。

The recommended default partitioning for your disk is:

```
a 100% "SOLARIS System" partition.
```

To select this, please type "y". To partition your disk differently, type "n" and the "fdisk" program will let you select other partitions. **y**

7. ディスク全体にまたがる **Solaris fdisk** パーティションを作成しない場合は、プロンプトで **n** を入力します。

To select this, please type "n". To partition your disk differently, type "n" and the "fdisk" program will let you select other partitions. **n**

Total disk size is 2694 cylinders

Cylinder size is 765 (512 byte) blocks

```

                                Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE
FOLLOWING:
```

1. Create a partition
2. Change Active (Boot from) partition

3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exit without updating disk configuration)

Enter Selection:

8. 「1. Create a partition」を選択し、**fdisk** パーティションを作成します。

```
Total disk size is 2694 cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
```

Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	===	=====	===

THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exit without updating disk configuration)

Enter Selection: 1

9. 「1(=Solaris)」を選択して、**Solaris fdisk** パーティションを作成します。

```
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ? 1
```

10. **Solaris fdisk** パーティション用に割り当てるディスクのパーセントを指定します。このパーセントを計算するときには、既存の **fdisk** パーティションのサイズを考慮してください。

```
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). mm
```

11. プロンプトで **y** を入力して、**Solaris fdisk** パーティションをアクティブにします。

```
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
```

fdisk パーティションがアクティブになった後、Enter Selection: プロンプトが表示されます。

12. 別の **fdisk** パーティションを作成する場合は、「1. Create a partition」を選択します。

fdisk パーティションを作成する手順については、手順9～11を参照してください。

13. ディスク構成を更新し、**Selection** メニューから **fdisk** メニューに戻ります。

Selection: 4

14. **label** コマンドを使用して、ディスクに新しいラベルを付けます。

```
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format>
```

15. **format** メニューを終了します。

```
format> quit
```

x86: 次に進む手順

ディスク上に Solaris fdisk パーティションを作成し終わったら、ディスク上にスライスを作成できます。521 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

x86: 例 — ディスク全体を占有する Solaris fdisk パーティションを作成する

次の例は、format の fdisk オプションを使用して、ディスク全体にまたがる Solaris fdisk パーティションを作成する方法を説明しています。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c0d0 <DEFAULT cyl 2466 alt 2 hd 16 sec 63>
       /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0
    1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
       /pci@0,0/pci-ide@7,1/ide@0/cmdk@1,0
    2. c1d0 <DEFAULT cyl 13102 alt 2 hd 16 sec 63>
       /pci@0,0/pci-ide@7,1/ide@1/cmdk@0,0
Specify disk (enter its number): 0
selecting c0d0
Controller working list found
[disk formatted]
format> fdisk
The recommended default partitioning for your disk is:

    a 100% "SOLARIS System" partition.

To select this, please type "y".  To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. y

WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> quit
```

x86: 例 — 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する

次の例は、DOS-BIG fdisk パーティションがすでに存在しているディスクに、Solaris fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
Total disk size is 2694 cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type  Start  End  Length  %
=====  =====  =====  =====  =====  =====  =====
1          DOS-BIG  1      538    538    20
SELECT ONE OF THE FOLLOWING:
1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type  Start  End  Length  %
=====  =====  =====  =====  =====  =====  =====
1          DOS-BIG  1      538    538    20
2          Active  SOLARIS  539    2693  2155    80
SELECT ONE OF THE FOLLOWING:
1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
Enter Selection: Selection: 4
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> q
```

x86: 例 — Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する

次の例は、Solaris fdisk パーティションと DOSBIG fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
The recommended default partitioning for your disk is:
  a 100% "SOLARIS System" partition.
To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
                                Cylinders
Partition  Status    Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE FOLLOWING:
  1.  Create a partition
  2.  Change Active (Boot from) partition
  3.  Delete a partition
  4.  Exit (Update disk configuration and exit)
  5.  Cancel (Exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?8
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 20
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". n
      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
                                Cylinders
Partition  Status    Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
          1          DOS-BIG      1    538    538    20
SELECT ONE OF THE FOLLOWING:
  1.  Create a partition
  2.  Change Active (Boot from) partition
  3.  Delete a partition
  4.  Exit (Update disk configuration and exit)
  5.  Cancel (Exit without updating disk configuration)Enter
Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
```



```

cylinders
      Cylinder size is 765 (512 byte) blocks
      Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
      1          DOS-BIG      1   538    538    20
      2      Active  SOLARIS    539 2693   2155    80

SELECT ONE OF THE FOLLOWING:
  1.  Create a partition
  2.  Change Active (Boot from) partition
  3.  Delete a partition
  4.  Exit (Update disk configuration and exit)
  5.  Cancel (Exit without updating disk configuration)
Enter Selection: 4
format> q

```

▼ x86: ディスクスライスとディスクラベルを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、パーティション分割をし直したいディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number は、パーティションの再分割を実行するディスクの番号です。

4. **partition** メニューを選択します。

```
format> partition
```

5. 現在のパーティション (スライス) テーブルを表示します。

```
partition> print
```

6. 変更作業を開始します。

```
partition> modify
```

7. ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]? 1
```

free hog スライスの詳細については、474 ページの「free hog スライス の使用方法」を参照してください。

8. 続行するかどうかを尋ねるプロンプトが表示されたら、**yes** と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition
table based on above table[yes]? yes
```

9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0)、スワップ (スライス 1) (必須) および
- /usr (スライス 6)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。

10. プロンプトが表示されたら **yes** と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? yes
```

表示されたパーティションテーブルが希望どおりでないために変更したい場合は、**no** と応答して手順 6 に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

partition-name には、新しいパーティションテーブルの名前を指定します。

12. 新しいディスク上でスライスを割り当て終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. **partition** メニューを終了します。

```
partition> quit
```

14. 新しいディスクラベルを検査します。

```
format> verify
```

15. **format** メニューを終了します。

```
format> quit
```

x86: 次に進む手順

ディスクスライスとディスクラベルを作成し終わったら、ディスク上にファイルシステムを作成できます。523 ページの「x86: ファイルシステムを作成する方法」を参照してください。

▼ x86: ファイルシステムを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```

/dev/rdisk/cwtxdysz は、作成するファイルシステム用の raw デバイスです。
newfs コマンドの詳細については、第 39 章または newfs (1M) のマニュアルページを参照してください。

3. マウントして、新規ファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls /mnt
lost+found
```

x86: 次に進む手順

追加するディスク	作業
システムディスク	ディスク上にルート (/) と /usr のファイルシステムを復元する必要がある。第 49 章を参照。 ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールする。523 ページの「x86: システムディスクにブートブロックをインストールする方法」を参照。
二次ディスク	新しいディスク上にファイルシステムを復元しなければならないことがある。第 49 章を参照。 新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が終わったことになる。ファイルシステムをユーザーが利用できるようにする方法については、第 40 章を参照。

▼ x86: システムディスクにブートブロックをインストールする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 起動ブロックをシステムディスクにインストールします。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i` /lib/fs/ufs/bootblk /dev/rdisk/cwtxdysz2
```

<code>/usr/platform/`uname -i`/lib/fs/ufs/pboot</code>	パーティションのブートファイル
<code>/usr/platform/`uname -i`/lib/fs/ufs/bootblk</code>	ブートブロックコード
<code>/dev/rdisk/ <i>cwtxdys2</i></code>	ディスク全体を表す raw デバイス名

3. システムをリブートし、レベル 3 で実行することによって、ブートブロックがインストールされていることを確認します。

```
# init 6
```

x86: 例 — システムディスクにブートブロックをインストールする

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot  
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c0t6d0s2
```

第 36 章

format ユーティリティ (参照情報)

この章では、format ユーティリティのメニューとコマンドについて説明します。

この章の内容は次のとおりです。

- 525 ページの「format ユーティリティを使用する上での推奨事項および要件」
- 526 ページの「format のメニューとコマンドの説明」
- 532 ページの「format.dat ファイル」
- 537 ページの「format コマンドへの入力規則」
- 539 ページの「format ユーティリティのヘルプを利用する」

format ユーティリティの概要については、第 32 章を参照してください。

format ユーティリティを使用する上での推奨事項および要件

format ユーティリティを使用するには、スーパーユーザーになるか、同等の役割を引き受ける必要があります。そうではない場合、format ユーティリティを使用しようとすると次のエラーメッセージが表示されます。

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

既存のデータを維持しつつ format ユーティリティを使用する場合、次の指針に従って操作を行ってください。

- ディスクドライブ上のすべてのファイルのバックアップを作成します。
- format ユーティリティの dump コマンドを使用して、欠陥領域リストをファイルに保存します。ファイル名には、ドライブタイプ、モデル番号、シリアル番号を含めておくべきです。

- メーカーから出荷時にドライブといっしょに提供された、欠陥領域リストを保管します。

format のメニューとコマンドの説明

format のメインメニューは次のようになっています。

```

FORMAT MENU:
  disk          - select a disk
  type          - select (define) a disk type
  partition     - select (define) a partition table
  current       - describe the current disk
  format        - format and analyze the disk
  repair        - repair a defective sector
  label         - write label to the disk
  analyze       - surface analysis
  defect        - defect list management
  backup        - search for backup labels
  verify        - read and display labels
  save          - save new disk/partition definitions
  inquiry       - show vendor, product and revision
  volname       - set 8-character volume name
  quit

```

format>

次の表に、format のメインメニュー項目を示します。

表 36-1 format のメインメニュー項目の説明

項目	コマンド/メニュー	説明
disk	コマンド	システムのドライブをすべて表示する。後の操作で使用するディスクを選択することもできる。このディスクは、「現在のディスク」と呼ばれる。
type	コマンド	現在のディスクのメーカーとモデルを表示する。認識されているドライブタイプのリストも表示する。SCSI-2 対応ディスクドライブの場合は Auto configure オプションを選択する。
partition	メニュー	スライスの作成および変更を行う。詳細は、528 ページの「partition メニュー」を参照。

表 36-1 format のメインメニュー項目の説明 (続き)

項目	コマンド/メニュー	説明
current	コマンド	現在のディスクに関する次の情報を表示する。 <ul style="list-style-type: none"> ■ デバイス名とデバイスタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ 物理デバイス名
format	コマンド	次のいずれかの情報源をこの順番に使用して、現在のディスクをフォーマットする。 <ol style="list-style-type: none"> 1. format.dat ファイル内の情報 2. 自動構成プロセスからの情報 3. format.dat エントリが見つからない場合、プロンプトに入力した情報 <p>このコマンドは、IDE ディスクには適用できない。IDE ディスクは、あらかじめ製造元でフォーマットされる。</p>
fdisk	メニュー	x86 プラットフォームのみ: fdisk プログラムを実行し、Solaris fdisk パーティションを作成する。
repair	コマンド	現在のディスク上で特定のブロックを修復する。
label	コマンド	現在のディスクに新しいラベルを書き込む。
analyze	メニュー	読み取り、書き込み、比較テストを実行する。詳細は、530 ページの「analyze メニュー」を参照。
defect	メニュー	欠陥リストを検索して出力する。詳細は、531 ページの「defect メニュー」を参照。この機能は、IDE ディスクには適用できない。IDE ディスクは、自動欠陥管理を行う。
backup	コマンド	VTOC - バックアップラベルを検索 EFI - サポートされない
verify	コマンド	現在のディスクに関する次の情報を出力する。 <ul style="list-style-type: none"> ■ デバイス名とデバイスタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ パーティションテーブル
save	コマンド	VTOC - 新しいディスク情報およびパーティション情報を保存 EFI - 適用できない
inquiry	コマンド	現在のドライブのベンダ、製品名、リビジョンレベルが出力される (SCSI ディスクのみ)。
volname	コマンド	新しい 8 文字のボリューム名を使用してディスクラベルを作成する。

表 36-1 format のメインメニュー項目の説明 (続き)

項目	コマンド/メニュー	説明
quit	コマンド	format メニューを終了する。

partition メニュー

partition メニューは次のようになっています。

```
format> partition
PARTITION MENU:
  0 - change `0' partition
  1 - change `1' partition
  2 - change `2' partition
  3 - change `3' partition
  4 - change `4' partition
  5 - change `5' partition
  6 - change `6' partition
  7 - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit
partition>
```

次の表に、partition メニューの項目を示します。

表 36-2 partition メニュー項目の説明

サブコマンド	説明
change `n' partition	新しいスライスに次の情報を設定する。 <ul style="list-style-type: none"> ■ 識別タグ ■ アクセス権フラグ ■ 開始シリンダ ■ サイズ
select	事前定義されたスライステーブルを選択する。
modify	スライステーブル内のすべてのスライスを変更可能にする。個々のスライスに対して change `x' partition コマンドを実行するよりも、このコマンドが使用されることが多い。
name	現在のスライステーブルの名前を指定する。
print	現在のスライステーブルを表示する。
label	スライスマップとラベルを現在のディスクに書き込む。

表 36-2 partition メニュー項目の説明 (続き)

サブコマンド	説明
quit	partition メニューを終了する。

x86: fdisk メニュー

x86 システム上でのみ、次のような fdisk メニューが表示されます。

```
format> fdisk
Total disk size is 1855 cylinders
Cylinder size is 553 (512 byte) blocks
Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ===
          1          Active  DOS-BIG   0     370    371    20
          2          Active  SOLARIS   370   1851   1482    80

SELECT ONE OF THE FOLLOWING:

1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)

Enter Selection:
```

次の表に、fdisk メニューの項目を示します。

表 36-3 x86: fdisk メニュー項目の説明

メニュー項目	説明
Create a partition	fdisk パーティションを作成する。Solaris や DOS など、オペレーティングシステムごとに別々のパーティションを作成しなければならない。1 台のディスクの最大パーティション数は 4 である。fdisk のパーティションのサイズをパーセンテージで入力するように促すプロンプトが表示される。
Change Active partition	ブートに使用するパーティションを指定する。このメニュー項目により、第 1 段階のブートプログラムが第 2 段階のブートプログラムを検索する場所を指定する。
Delete a partition	以前に作成したパーティションを削除する。このコマンドを実行すると、パーティション内のすべてのデータが失われる。
Exit	新しいパーティションテーブルを書き込んで fdisk メニューを終了する。

表 36-3 x86: fdisk メニュー項目の説明 (続き)

メニュー項目	説明
Cancel	パーティションテーブルを変更せずに fdisk メニューを終了する。

analyze メニュー

analyze メニューは次のようになっています。

```
format> analyze
```

```
ANALYZE MENU:
```

```
read      - read only test      (doesn't harm SunOS)
refresh   - read then write     (doesn't harm data)
test      - pattern testing     (doesn't harm data)
write     - write then read     (corrupts data)
compare   - write, read, compare (corrupts data)
purge     - write, read, write  (corrupts data)
verify    - write entire disk, then verify (corrupts data)
print     - display data buffer
setup     - set analysis parameters
config    - show analysis parameters
quit
```

```
analyze>
```

次の表に、analyze メニューの項目を示します。

表 36-4 analyze メニュー項目の説明

サブコマンド	説明
read	現在のディスクの各セクターを読み込む。デフォルトで欠陥ブロックを修復する。
refresh	データを損なわずに、現在のディスク上で読み取りおよび書き込みを実行する。デフォルトで欠陥ブロックを修復する。
test	データを損なわずに一連のパターンをディスクに書き込む。デフォルトで欠陥ブロックを修復する。
write	一連のパターンをディスクに書き込んでから、そのデータをディスクから読み込む。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。
compare	ディスクに一連のパターンを書き込み、そのデータを読み込み、書き込みバッファ内のデータと比較する。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。

表 36-4 analyze メニュー項目の説明 (続き)

サブコマンド	説明
purge	ディスク上のデータをすべて削除し、いかなる手段でも取り出せないようにする。ディスク全体(またはディスクのセクション)に3種類のパターンを書き込むことにより、データを削除する。検査に合格すると16進のビットパターンがディスク全体(またはディスクのセクション)に上書きされる。 デフォルトで欠陥ブロックを修復する。
verify	1度目にディスク全体の各ブロックに固有のデータを書き込む。2度目にそのデータを読み込んで検査する。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。
print	読み取り / 書き込みバッファ内のデータを表示する。
setup	次の解析パラメータを指定する。 Analyze entire disk? yes Starting block number: ドライブによって異なる Ending block number: ドライブによって異なる Loop continuously? no Number of passes: 2 Repair defective blocks? yes Stop after first error? no Use random bit patterns? no Number of blocks per transfer: 126 (0/n/nm) Verify media after formatting? yes Enable extended messages? no Restore defect list? yes Restore disk label? yes 太字はデフォルトを示す。
config	現在の解析パラメータを表示する。
quit	analyze メニューを終了する。

defect メニュー

defect メニューは次のようになっています。

```
format> defect

DEFECT MENU:
  primary - extract manufacturer's defect list
  grown   - extract manufacturer's and repaired defects lists
  both    - extract both primary and grown defects lists
  print   - display working list
  dump    - dump working list to file
  quit

defect>
```

次の表に、defect メニューの項目を示します。

表 36-5 defect メニュー項目の説明

サブコマンド	説明
primary	メーカーの欠陥リストをディスクドライブから読み込み、メモリー内の欠陥リストを更新する。
grown	増分の欠陥リスト(分析により検出された欠陥)を読み込んで、メモリー内の欠陥リストを更新する。
both	メーカーの欠陥リストと増分の欠陥リストを読み込んで、メモリー内の欠陥リストを更新する。
print	メモリー内の欠陥リストを表示する。
dump	メモリー内の欠陥リストをファイルに保存する。
quit	defect メニューを終了する。

format.dat ファイル

Solaris オペレーティングシステムと一緒に出荷される format.dat ファイルでは、多数の標準的なディスクがサポートされます。使用中のディスクドライブが format.dat ファイルに含まれていない場合は、ファイルにエンTRIESを追加するか、format ユーティリティで処理を実行中に type コマンドを選択してから other オプションを選択してエンTRIESを追加できます。

ディスクドライブをサイト全体で使用する場合は、format.dat ファイルにエンTRIESを追加すると時間を節約できます。format.dat ファイルを他のシステム上で使用する場合は、format.dat ファイルに追加する特定のディスクドライブを使用するシステムごとに、このファイルをコピーしてください。

次のいずれかの場合には、システムの /etc/format.dat ファイルを変更する必要があります。

- ディスクが Solaris オペレーティング環境でサポートされない場合
- スライステーブルが入っているディスクが、Solaris オペレーティング環境のデフォルト構成とは異なる場合

注 - /etc/format.dat ファイルのデフォルトエンTRIESは変更しないでください。デフォルトエンTRIESを変更する場合は、混乱を避けるために、そのエンTRIESをコピーし、別の名前を付けてから変更します。

/etc/format.dat は、EFI ラベル付きディスクには適用できません。

format.dat ファイルの内容

format.dat データファイルには、format ユーティリティに使用されるディスクドライブ情報が入っています。format.dat ファイル内では、次の3つの項目が定義されています。

- 検索パス
- ディスクタイプ
- スライステーブル

format.dat ファイルの構文

/etc/format.dat ファイルには、次の構文規則が適用されます。

- ポンド記号 (#) はコメント文字です。ポンド記号に続く1行のテキストは、format ユーティリティでは解釈されません。
- format.dat ファイル内の各定義は、1つの論理行で評価されます。定義が長すぎて1行に収まらない場合は、定義の最終行を除くすべての行末にバックスラッシュ (\) を付けなければなりません。
- 定義は、左辺に識別子、右辺に1つまたは複数の値を持つ一連の代入式からなっています。代入演算子は等号 (=) です。定義内の代入式はコロン (:) で区切らなければなりません。
- format ユーティリティは、空白を無視します。代入値に空白を含める場合は、値全体を二重引用符 (") で囲みます。この構文により、引用符の内側の空白は代入値の一部として保持されます。
- 代入式によっては、右辺に複数の値を指定できるものがあります。値はコンマ (,) で区切ります。

format.dat ファイル中のキーワード

format.dat ファイルには、format ユーティリティが起動時に読み込むディスク定義が含まれます。各定義の先頭には、キーワード `disk_type` または `partition` が付きます。これらのキーワードについて、次の表で説明します。

表 36-6 format.dat ファイルのキーワードの説明

キーワード	用途
disk_type	<p>コントローラとディスクのモデルを定義する。各 disk_type 定義には、ディスクの物理ジオメトリに関する情報が入っている。デフォルトのデータファイルには、Solaris オペレーティング環境でサポートされるコントローラとディスクの定義が入っている。</p> <p>サポートされないディスクを使用する場合にのみ、新しい disk_type を追加する必要がある。必要に応じて、disk_type 定義をデータファイルにいくつ追加してもかまわない。</p>
partition	<p>特定のディスクタイプのスライステーブルを定義する。スライステーブルには、スライス情報だけでなく、format ユーティリティ内で参照可能な名前が入っている。デフォルトの format.dat ファイルには、数種類のディスクドライブに対応するデフォルトのスライス定義が含まれる。システムのディスク上にスライスを作成し直した場合は、スライス定義を追加する。必要に応じて、スライス情報をデータファイルにいくつ追加してもかまわない。</p>

ディスクタイプ (format.dat)

format.dat ファイル内の disk_type キーワードは、コントローラとディスクのモデルを定義します。各 disk_type 定義には、ディスクの物理ジオメトリに関する情報が含まれます。デフォルトの format.dat ファイルには、Solaris オペレーティング環境でサポートされるコントローラとディスクの定義が入っています。サポートされないディスクを使用する場合にのみ、新しい disk_type を追加する必要があります。必要に応じて、disk_type 定義をデータファイルにいくつ追加してもかまいません。

キーワード自体が、ディスクタイプ名になります。この名前は、ディスクのラベルの一部になり、format ユーティリティの実行時にディスクタイプを識別するために使用されます。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべての disk_type 定義でキーワードのほかに割り当てなければならない識別子を示します。

表 36-7 必須の disk_type 識別子

識別子	説明
ctrlr	ディスクタイプで有効なコントローラのタイプ。現在、有効な値は SCSI と ATA である。
ncyl	ディスクタイプ内のデータシリンダ数。この数によって、システムがアクセスできるディスクの論理シリンダ数が決まる。
acyl	ディスクタイプ内の代替シリンダ数。format ユーティリティは、これらのシリンダを使用して、ドライブの欠陥リストなどの情報を格納する。代替シリンダとして、常に 2 つ以上のシリンダを残しておく必要がある。

表 36-7 必須の disk_type 識別子 (続き)

識別子	説明
pcyl	ディスクタイプ内の物理シリンダ数。この数値は、ディスクメディアの境界を計算するために使用される。通常、この数値は ncy1 と acyl の合計に等しくなる。
nhead	ディスクタイプ内のヘッド数。この数値は、ディスクメディアの境界を計算するために使用される。
nsect	ディスクタイプ内の 1トラック当たりのデータセクター数。この数値は、ディスクメディアの境界を計算するために使用される。これはデータセクターだけである。スベアは、各トラックのデータセクション数には含まれない。
rpm	ディスクタイプの 1分当たりの回転数。この情報はラベルに書き込まれ、後からファイルシステムでファイルデータの最適位置の計算に使用される。

コントローラによっては、他の識別子が必要な場合があります。次の表に、SCSI コントローラに必要な識別子を示します。

表 36-8 SCSI コントローラの disk_type 識別子

識別子	説明
fmt_time	所定のドライブのフォーマットに要する時間を示す数値。詳細は、コントローラのマニュアルを参照。
cache	format ユーティリティの処理中にオンボードキャッシュの動作を制御する数値。詳細は、コントローラのマニュアルを参照。
trks_zone	代替セクターのマッピング内で使用される 1つの欠陥領域当たりのトラック数を指定した数値。詳細は、コントローラのマニュアルを参照。
asect	所定の欠陥領域内で代替マッピングに利用可能なセクター数を指定する。詳細は、コントローラのマニュアルを参照。

次に、disk_type 定義の例を示します。

```
disk_type = "SUN1.3G" \
: ctrl = SCSI : fmt_time = 4 \
: trks_zone = 17 : asect = 6 : atrks = 17 \
: ncy1 = 1965 : acyl = 2 : pcyl = 3500 : nhead = 17 : nsect = 80 \
: rpm = 5400 : bpt = 44823
```

```
disk_type = "SUN2.1G" \
: ctrl = SCSI : fmt_time = 4 \
: ncy1 = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \
: rpm = 5400 : bpt = 44823
```

```
disk_type = "SUN2.9G" \
: ctrl = SCSI : fmt_time = 4 \
```

```
: n cyl = 2734 : a cyl = 2 : p cyl = 3500 : n head = 21 : n sect = 99 \  
: rpm = 5400
```

パーティションまたはスライステーブル (format.dat)

format.dat ファイル内のパーティションテーブルに、特定のディスクタイプのスライステーブルが定義されています。

format.dat ファイル内の partition キーワードが、スライステーブル名になります。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべてのスライステーブル内で値を代入しなければならない識別子を示します。

表 36-9 スライステーブルの必須識別子

識別子	説明
disk	このスライステーブルが定義されている disk_type の名前。この名前は disk_type 内で使用されるとおりに指定しなければならない。
ctlr	このスライステーブルを接続できるコントローラタイプディスク。現在、有効な値は ATA コントローラを表す ATA と SCSI コントローラを表す SCSI である。ここで指定したコントローラタイプは、disk_type 定義で選択した disk_type にも定義する必要がある。

スライス定義内の他の識別子では、実際のスライス情報を記述します。識別子は 0 から 7 までの番号です。これらの識別子は省略可能です。明示的に割り当てられていないスライスは、長さ 0 に設定されます。これらの識別子の値は、それぞれコンマで区切られた数値のペアになります。最初の数値はスライスの開始シリンダで、第 2 はスライス内のセクター数です。次に、スライス定義の例を示します。

```
partition = "SUN1.3G" \  
: disk = "SUN1.3G" : ctlr = SCSI \  
: 0 = 0, 34000 : 1 = 25, 133280 : 2 = 0, 2672400 : 6 = 123, 2505120  
  
partition = "SUN2.1G" \  
: disk = "SUN2.1G" : ctlr = SCSI \  
: 0 = 0, 62320 : 1 = 41, 197600 : 2 = 0, 4154160 : 6 = 171, 3894240  
  
partition = "SUN2.9G" \  
: disk = "SUN2.9G" : ctlr = SCSI \  
: 0 = 0, 195426 : 1 = 94, 390852 : 2 = 0, 5683986 : 6 = 282, 5097708
```

format ユーティリティの代替データファイルを指定する

format ユーティリティは、次の方法で代替ファイルの位置を認識します。

1. `format -x` オプションでファイル名を指定した場合、ファイルは常にデータファイルとして使用されます。
2. `-x` オプションを指定しない場合、`format` ユーティリティは現在のディレクトリ内でファイル `format.dat` を検索します。このファイルが見つかったら、データファイルとして使用されます。
3. どちらの方法でもデータファイルが見つからない場合、`format` ユーティリティはデータファイルとして `/etc/format.dat` を使用します。このファイルは Solaris オペレーティング環境と共に出荷されるので、必ず存在します。

format コマンドへの入力規則

`format` ユーティリティを使用する場合は、さまざまな情報を入力する必要があります。この節では、入力する情報に関する規則について説明します。データ入力時に `format` のヘルプ機能を使用する方法については、539 ページの「`format` ユーティリティのヘルプを利用する」を参照してください。

format コマンドへ番号を指定する

`format` ユーティリティを使用する際、いくつかの数値を入力する必要があります。入力方法には、データを指定する方法と、選択肢のリストから番号を選択する方法があります。どちらの場合も、`help` 機能を使用すると、`format` は期待する数値の上限と下限を表示します。したがって、目的の数値を入力するだけで済みます。数値は、その一部として底を明示的に指定しない限り (16 進数を表す `0x` など)、10 進数と見なされます。

次の例は、整数の入力を示しています。

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

format コマンドへブロック番号を指定する

ディスクのブロック番号を入力しなければならない場合は、情報を次の 2 つの方法で入力できます。

- ブロック番号を整数として入力する。
- ブロック番号をシリンダ / ヘッド / セクター書式で入力する。

この情報は、論理ブロック番号を表す整数として指定できます。任意の底の数値を指定できますが、デフォルトは 10 進です。また、ここで最大演算子 (ドル記号 `$`) を使用して、`format` ユーティリティに適切な値を選択させることもできます。論理ブロックの形式は、SunOS のディスクドライバによってエラーメッセージに使用されません。

ブロック番号を指定するには、シリンダ / ヘッド / セクター書式を使用する方法もあります。この形式では、ブロック番号の3つの論理構成要素である、シリンダ、ヘッド、およびセクターの値を明示的に指定しなければなりません。これらの値は論理値ですが、メディアのレイアウトに関連するディスク領域の定義に使用できます。

シリンダ / ヘッド / セクター番号を指定しない場合、値は0であると見なされます。また、番号の代わりに最大演算子を使用して、format ユーティリティに適切な値を選択させることもできます。次に、シリンダ、ヘッド、セクターエントリの例を示します。

```
Enter defective block number: 34/2/3
Enter defective block number: 23/1/
Enter defective block number: 457//
Enter defective block number: 12345
Enter defective block number: 0xabcd
Enter defective block number: 334/$/2
Enter defective block number: 892//
```

format は、ブロック番号を常に上記の両方の書式で出力します。また、help 機能によって、期待されるブロック番号の上限と下限が両方の書式で表示されます。

format のコマンド名を指定する

format ユーティリティでメニュープロンプトが表示される場合は、コマンド名を入力する必要があります。コマンド名は、目的のコマンドとして区別できる長さまで「省略」できます。

たとえば、p(partition) を使用して format メニューから partition メニューにアクセスできます。次に、p(rint) を使用して現在のスライステーブルを表示できます。

```
format> p
PARTITION MENU:
  0 - change `0' partition
  1 - change `1' partition
  2 - change `2' partition
  3 - change `3' partition
  4 - change `4' partition
  5 - change `5' partition
  6 - change `6' partition
  7 - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit
partition> p
```

format コマンドへディスク名を指定する

format ユーティリティでは、名前を指定しなければならない場合があります。このような場合は、名前に使用したい文字列を自由に指定できます。空白を含む名前は、二重引用符 (") で囲まなければならない。二重引用符で囲まなければ、名前の最初の語だけが使用されます。

たとえば、ディスクの特定のパーティションテーブルを指定する場合、partition メニューの name サブコマンドを使用できます。

```
partition> name
Enter table name (remember quotes): "new disk3"
```

format ユーティリティのヘルプを利用する

format ユーティリティにはヘルプ機能が組み込まれており、format ユーティリティが入力待ちの状態であればいつでも使用できます。疑問符 (?) を入力するだけで必要な入力に関するヘルプが表示されます。format ユーティリティでは、どんなタイプの入力が必要かについて簡潔な説明が表示されます。

メニュープロンプトから ? と入力すると、利用できるコマンドのリストが表示されます。

format ユーティリティに関連するマニュアルページには、次が含まれます。

- format (1M) - format ユーティリティの基本機能およびコマンド行で使用可能なすべての変数について説明する。
- format.dat (4) - format ユーティリティで使用するディスクドライブ構成に関する情報を提供する。

第 37 章

ファイルシステムの管理

以下の各章では、ファイルシステムを管理する方法について説明します。

第 38 章	ファイルシステムのタイプ、通常よく使用される管理コマンド、ファイルシステムのマウントやマウント解除など、ファイルシステム全般に関する概要について説明します。ファイルシステムのタイプを決める手順についても説明します。
第 39 章	UFS ファイルシステム、一時ファイルシステム (TMPFS)、およびループバックファイルシステム (LOFS) を作成する手順について説明します。
第 40 章	どのファイルシステムがマウントされているかを判断する手順、 <code>/etc/vfstab</code> に列挙されているファイルをマウントする方法、UFS、NFS、PCFS (DOS) ファイルシステムをマウントする方法について説明します。
第 41 章	CacheFS ファイルシステムの概要と使用手順について説明します。
第 42 章	スワップリソースを監視する手順、スワップファイルを作成して使用可能にする手順、不要になったスワップ空間を削除する手順について説明します。
第 43 章	ファイルシステムの状態を記録する方法、 <code>fsck</code> プログラムでチェックする内容、自動ブートチェック機能を変更する方法、 <code>fsck</code> プログラムの使用方法について説明します。
第 44 章	ルート (/) および <code>/usr</code> ファイルシステムのデフォルトディレクトリ、 <code>/kernel</code> ディレクトリに入っているデフォルトディレクトリ、 <code>mkfs</code> コマンドと <code>newfs</code> コマンドの詳細など、ファイルシステムの参照情報について説明します。

ファイルシステムの管理 (概要)

ファイルシステムの管理は、最も重要なシステム管理作業の 1 つです。

この章の内容は次のとおりです。

- 543 ページの「Solaris 9 8/03 リリースの新しいファイルシステム機能」
- 550 ページの「Solaris 9 リリースの新しいファイルシステム機能」
- 552 ページの「ファイルシステム管理作業についての参照先」
- 552 ページの「ファイルシステムの概要」
- 553 ページの「ファイルシステムのタイプ」
- 558 ページの「ファイルシステム管理用のコマンド」
- 560 ページの「デフォルトの Solaris ファイルシステム」
- 561 ページの「スワップ空間」
- 562 ページの「UFS ファイルシステム」
- 564 ページの「ファイルシステムのマウントとマウント解除」
- 569 ページの「ファイルシステムのタイプを調べる」

Solaris 9 8/03 リリースの新しいファイルシステム機能

この節では、この Solaris リリースの新しいファイルシステム機能について説明します。

SPARC: マルチテラバイトの UFS ファイルシステムのサポート

この Solaris リリースでは、64 ビットの Solaris カーネルを実行しているシステムで、マルチテラバイトの UFS ファイルシステムをサポートします。

以前は、64ビットと32ビットのどちらのシステムでも、UFS ファイルシステムはおよそ1Tバイトに制限されていました。UFS ファイルシステムのコマンドとユーティリティはすべて、マルチテラバイトのUFS ファイルシステムに対応するように更新されました。

たとえば、`ufsdump` コマンドは大規模なUFS ファイルシステムをダンプできるようにブロックサイズが大きくなりました。

```
# ufsdump 0f /dev/md/rdisk/d97 /dev/md/rdisk/d98
DUMP: Date of this level 0 dump: Tue Jan 07 14:23:36 2003
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/md/rdisk/d98 to /dev/md/rdisk/d97
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Forcing larger tape block size (2048).
DUMP: Writing 32 Kilobyte records
DUMP: Estimated 4390629500 blocks (2143862.06MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
```

1Tバイト未満のUFS ファイルシステムは、以前と同じように管理されます。1Tバイト未満のUFS ファイルシステムと1Tバイトを超えるファイルシステムとの間に管理面での違いはありません。

最初に1Tバイト未満のUFS ファイルシステムを作成し、最終的にマルチテラバイトのファイルシステムに拡張されるように指定できます。その場合は、`newfs -T` オプションを使用します。このオプションを使用すると、マルチテラバイトのファイルシステムに適した値に拡張が可能であるように*i* ノードとフラグメントの密度が設定されます。

32ビットのカーネルを実行しているシステムで1Tバイト未満のUFS ファイルシステムを作成するときに、`newfs -T` オプションを使用すると、最終的に64ビットのカーネルでこのシステムを起動するときに、`growfs` コマンドを使ってこのファイルシステムを拡張できます。詳細については、`newfs(1M)` のマニュアルページを参照してください。

`growfs` コマンドを使用すると、サービスやデータを失わずに、UFS ファイルシステムをスライスまたはボリュームのサイズまで拡張できます。詳細については、`growfs(1M)` のマニュアルページを参照してください。

これに関連して、EFI ディスクラベルによるマルチテラバイトボリュームのサポートと、Solaris ボリュームマネージャによるマルチテラバイトボリュームのサポートという新しい2つの機能が追加されました。詳細については、456ページの「SPARC: EFI ディスクラベルによるマルチテラバイトボリュームのサポート」および『Solaris ボリュームマネージャの管理』を参照してください。

マルチテラバイトのUFS ファイルシステムの機能

マルチテラバイトのUFS ファイルシステムには、次の機能があります。

- 最大16TバイトのUFS ファイルシステムを作成できる

- 16T バイト未満のファイルシステムを作成し、後で 16T バイトまでサイズを増やすことができる
- マルチテラバイトのファイルシステムを物理ディスク、Solaris ボリュームマネージャの論理ボリューム、および Veritas の VxVM 論理ボリューム上に作成できる
- 1T バイトを超えるファイルシステムで、UFS ロギングがデフォルトで有効になっている。マルチテラバイトのファイルシステムでは、UFS ロギングが有効になっているとパフォーマンスが向上するというメリットがある。また、ロギングが有効なときは `fsck` コマンドを実行しなくてもよい場合があるというロギングのメリットもある

マルチテラバイトの UFS ファイルシステムの制限事項

マルチテラバイトの UFS ファイルシステムには、次の制限事項があります。

- Solaris x86 システムでは、この機能はサポートされていない
- 32 ビットの Solaris カーネルを実行しているシステムでは、1T バイトを超えるファイルシステムをマウントすることはできない
- 64 ビットの Solaris カーネルを実行しているシステムでは、1T バイトを超えるファイルシステムからブートすることはできない。つまり、マルチテラバイトのファイルシステムにルート (/) ファイルシステムを配置することはできない
- 1T バイトを超える個々のファイルはサポートされていない
- UFS ファイルシステムの 1T バイトあたりの最大ファイル数は 100 万である。100 万に制限されているのは、`fsck` コマンドでファイルシステムを検査するのに要する時間を減らすためである。
- マルチテラバイトの UFS ファイルシステムに設定できる最大割り当て数は、2T バイトの 1024 バイトブロックである
- 現時点では、`fsnap` コマンドを使ってマルチテラバイトの UFS ファイルシステムのスナップショットを作成することはできない

▼ マルチテラバイトの UFS ファイルシステムを作成する方法

マルチテラバイトの UFS ファイルシステムは、Solaris ボリュームマネージャまたは VxVM ボリューム、あるいは 1T バイトを超える物理ディスクとして提供されたマルチテラバイトの LUN が利用できることを前提にサポートされています。

マルチテラバイトの UFS ファイルシステムを作成する前に、次のいずれかを行う必要があります。

- `format` ユーティリティまたは Solaris インストールユーティリティを使ってマルチテラバイトのディスクパーティションを作成しておく
- Solaris ボリュームマネージャを使ってマルチテラバイトのボリュームを設定しておく

1. スーパーユーザーになります。
2. 論理ボリューム上にマルチテラバイトの **UFS** ファイルシステムを作成します。
たとえば、次のコマンドを実行すると、1.8T バイトボリュームの UFS ファイルシステムが作成されます。

```
# newfs /dev/md/rdisk/d99
newfs: construct a new file system /dev/md/rdisk/d99: (y/n)? y
/dev/md/rdisk/d99:
 3859402752 sectors in 628158 cylinders of 48 tracks,
128 sectors
      1884474.0MB in 4393 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
 3850872736, 3851751456, 3852630176, 3853508896, 3854387616, 3855266336,
 3856145056, 3857023776, 3857902496, 3858781216,
#
```

3. 新しく作成したファイルシステムの整合性を検査します。
次に例を示します。

```
# fsck /dev/md/rdisk/d99
** /dev/md/rdisk/d99
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 2 used, 241173122 free (0 frags, 241173122 blocks, 0.0%
fragmentation)
#
```

4. 新しく作成したファイルシステムをマウントして検査します。
次に例を示します。

```
# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem              size  used  avail capacity  Mounted on
/dev/md/dsk/d99         1.8T   64M   1.8T     1%    /bigdir
```

▼ マルチテラバイトの UFS ファイルシステムを拡張する方法

マルチテラバイトの UFS ファイルシステムを作成した後は、`growfs` コマンドを使ってファイルシステムを拡張できます。たとえば、前の手順でボリュームに作成したファイルシステムを使用すると、別のディスクをこのボリュームに追加できます。その後で、ファイルシステムを拡張します。

1. スーパーユーザーになります。
2. 別のディスクをボリュームに追加します。
次に例を示します。

```
# metattach d99 c4t5d0s4
d99: component is attached
# metastat
d99: Concat/Stripe
Size: 5145882624 blocks (2.4 TB)
Stripe 0:
  Device      Start Block  Dbase  Reloc
  c0t1d0s4    36864      Yes    Yes
Stripe 1:
  Device      Start Block  Dbase  Reloc
  c3t7d0s4          0      No    Yes
Stripe 2:
  Device      Start Block  Dbase  Reloc
  c1t1d0s4          0      No    Yes
Stripe 3:
  Device      Start Block  Dbase  Reloc
  c4t5d0s4          0      No    Yes
```

3. ファイルシステムを拡張します。
次に例を示します。

```
# growfs -v /dev/md/rdisk/d99
/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d99 5145882624
/dev/md/rdisk/d99:
 5145882624 sectors in 837546 cylinders of 48 tracks,
128 sectors
 2512638.0MB in 5857 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
 5137130400, 5138009120, 5138887840, 5139766560, 5140645280, 5141524000,
 5142402720, 5143281440, 5144160160, 5145038880,
#
```

4. 拡張したファイルシステムをマウントして検査します。
次に例を示します。

```
# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem      size  used  avail capacity  Mounted on
/dev/md/dsk/d99 2.4T   64M   2.4T     1%    /bigdir
```

▼ UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法

UFS ファイルシステムを 1T バイトを超えるサイズに拡張するには、次の手順を実行します。この手順は、`newfs -T` オプションを使って UFS ファイルシステムを作成したことを前提としています。

1. スーパーユーザーになります。
2. 現在のディスクまたはボリュームのサイズを特定します。
たとえば、次のボリュームは 800G バイトです。

```
# metastat d98
d98: Concat/Stripe
Size: 1677754368 blocks (800 GB)
Stripe 0:
  Device      Start Block  Dbase  Reloc
  c0t1d0s4      0         No     Yes
Stripe 1:
  Device      Start Block  Dbase  Reloc
  c3t7d0s4      0         No     Yes
```

3. このボリュームを **1T** バイトよりも大きくします。
次に例を示します。

```
# metattach d98 c1t1d0s4
d98: component is attached
# metastat d98
d98: Concat/Stripe
Size: 2516631552 blocks (1.2 TB)
Stripe 0:
  Device      Start Block  Dbase  Reloc
  c0t1d0s4      0         No     Yes
Stripe 1:
  Device      Start Block  Dbase  Reloc
  c3t7d0s4      0         No     Yes
Stripe 2:
  Device      Start Block  Dbase  Reloc
  c1t1d0s4      0         No     Yes
```

4. ディスクまたはボリュームの **UFS** ファイルシステムを **1T** バイトよりも大きくします。
次に例を示します。

```
growfs -v /dev/md/rdisk/d98
/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d98 2516631552
/dev/md/rdisk/d98:
2516631552 sectors in 68268 cylinders of 144 tracks,
256 sectors
1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 921888, 1843744, 2765600, 3687456, 4609312, 5531168, 6453024, 737...
8296736,
```

```
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
2507714848, 2508636704, 2509558560, 2510480416, 2511402272, 2512324128,
2513245984, 2514167840, 2515089696, 2516011552,
```

5. 拡張したファイルシステムをマウントして検査します。
次に例を示します。

```
# mount /dev/md/dsk/d98 /datadir
# df -h /datadir
Filesystem              size  used  avail capacity  Mounted on
/dev/md/dsk/d98         1.2T   64M   1.2T     1%     /datadir
```

マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡

マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡には、次のエラーメッセージと解決法を参考にしてください。

エラーメッセージ (次に例を示す)

```
mount: /dev/rdisk/c0t0d0s0 はこのファイルシステム形式ではありません。
```

原因

Solaris 9 8/03 よりも以前のリリースで動作するシステムに 1T バイトを超える UFS ファイルシステムをマウントしようとした。

解決法

1T バイトを超える UFS ファイルシステムは、Solaris 9 8/03 以降のリリースで動作するシステムにマウントしてください。

エラーメッセージ

```
"ファイルシステムがマルチテラバイト形式に設定されて
いません。" "そのサイズは 1 テラバイト以上にできません。"
```

原因

newfs -T コマンドを使わずに作成したファイルシステムを拡張しようとした。

解決法

1. 1T バイトより大きくするファイルシステムのデータをバックアップします。
2. newfs コマンドを使用して、ファイルシステムをマルチテラバイトファイルシステムに再作成します。ファイルシステムが時間とともにマルチテラバイトファイルシステムに拡大すると想定できる場合には、newfs -T コマンドを使用します。
3. 新しく作成したファイルシステムにバックアップされていたデータを復元します。

Solaris 9 リリースの新しいファイルシステム機能

この節では、Solaris 9 リリースの新しいファイルシステム機能について説明します。

拡張ファイル属性

UFS、NFS、および TMPFS ファイルシステムは、拡張ファイル属性を包含するように機能拡張されました。拡張ファイル属性によりアプリケーション開発者は、ファイルを特定の属性に関連付けることができます。たとえば、ウィンドウシステムファイル管理アプリケーションの開発者は、表示アイコンをファイルに関連付けることができます。拡張ファイル属性は、論理的には、ターゲットファイルに関連付けられている隠しディレクトリ内のファイルとして表されます。

runat コマンドを使用すると、属性を追加したり、拡張属性の名前空間に入っているシェルコマンドを実行したりできます。拡張属性の名前空間とは、指定のファイルに関連付けられている隠し属性ディレクトリのことです。

runat コマンドを使用して属性をファイルに追加するには、最初に属性ファイルを作成する必要があります。

```
$ runat filea cp /tmp/attrdata attr.1
```

次に、runat コマンドを使用して、ファイルの属性をリストに表示します。

```
$ runat filea ls -l
```

詳細については、runat (1) のマニュアルページを参照してください。

ファイル属性の照会、コピー、または検索に使用できる属性対応のオプションを提供することにより、Solaris ファイルシステムコマンドの多くは、ファイルシステム属性をサポートするよう変更されました。詳細については、各ファイルシステムコマンドのマニュアルページを参照してください。

UFS スナップショット

fssnap コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。スナップショットは、バックアップ操作のためのファイルシステムの一時的イメージです。

詳細については、第 48 章を参照してください。

UFS 直接入出力の並行処理の向上

バッファ処理されていないファイルシステムのデータにアクセスするためにデータベースアプリケーションが使用する直接入出力のパフォーマンスが改良され、通常の UFS ファイルへの読み取りおよび書き込みのアクセスの並行処理が可能になりました。これまでは、ファイルデータを更新する操作は、その更新操作が完了するまで、その他すべての読み取りアクセスまたは書き込みアクセスをロックアウトするようになっていました。

書き込みの並行処理は、ファイルのリライトという特別なケースのみに限られます。ファイルを拡張する場合は、書き込みは従来のようにシングルスレッドで行われます。一般に、データベースはファイルを事前に割り当て、その後はあまり拡張することはありません。そのため、この拡張による効果は通常のデータベース操作時に見られます。

直接入出力の改良により、UFS ファイルシステムにおける入出力を長時間使用するデータベースのパフォーマンスは、raw パーティションのアクセススピードの約 90 % に短縮されます。データベースが CPU やバス帯域幅を長時間使用する場合は、パフォーマンスの向上が見られないことがあります。

データベーステーブルの保存にすでに UFS を使用している場合、入出力データベースのアプリケーションを直接入出力が可能で実行することを想定してみてください。可能であれば、直接入出力の有効化にデータベースの管理手順を使用してください。データベース製品を通じて直接入出力を有効にする方法がない場合は、`mount -forcedirectio` オプションを使用して、ファイルシステムごとに直接入出力を有効にしてください。あるいは、`directio(3C)` ライブラリコールを使用して、直接入出力を有効にしてください。

詳細については、`mount_ufs(1M)` または `directio(3C)` のマニュアルページを参照してください。

mkfs コマンドのパフォーマンスの向上

`mkfs` コマンドのパフォーマンスの向上は、ファイルシステムの作成時に見られます。これにより、`mkfs` コマンドのパフォーマンスは、以前の Solaris リリースの 10 倍の速さになることもあります。`mkfs` コマンドのパフォーマンスの向上は、大規模ファイルシステムと小規模ファイルシステムのどちらの作成時にも見られます。特に大容量のディスクや高速のディスクを備えたシステムで最も顕著に現れます。

UDF ファイルシステムの新しい `labelit` コマンドオプション

`labelit` コマンドは、UDF (Universal Disk Format) ファイルシステムで使用する新しいオプションをいくつか提供します。新しい `labelit` コマンドオプションを使用すると、UDF ボリュームの作成者名、組織、および製品サポート情報を指定することができます。

Solaris の以前のリリースでは、この情報は UDF ファイルシステムの汎用部分であり、この情報を更新するためのメカニズムはありませんでした。

labelit コマンドの新しい UDF 専用オプション (-o オプションを指定) は、次のとおりです。

- lvinfo1 - UDF ファイルシステムの作成者を指定します。
- lvinfo2 - UDF ファイルシステムの作成に携わっている組織を指定します。
- lvinfo3 - UDF ファイルシステムを含むメディアに関する製品サポート情報を指定します。

各オプションの長さは最大 35 バイトです。

詳細については、labelit_udfs(1M) のマニュアルページを参照してください。

ファイルシステム管理作業についての参照先

ファイルシステムの管理に関する手順を調べるには、次の参照先を使用します。

ファイルシステム管理作業	参照先
新しいファイルシステムを作成する	第 39 章と第 41 章
ローカルファイルとリモートファイルをユーザーが利用できるようにする	第 40 章
新しい記憶デバイスを接続して構成する	第 32 章
バックアップスケジュールを計画して導入し、必要に応じてファイルとファイルシステムを復元する	第 46 章
ファイルシステムの矛盾をチェックして訂正する	第 43 章

ファイルシステムの概要

ファイルシステムは、ファイルを編成して格納するためのディレクトリ構造です。「ファイルシステム」という用語は、次のような場合に使用されます。

- ディスクベース、ネットワークベース、または仮想ファイルシステムなど、特定のタイプのファイルシステムを指す場合
- ルートディレクトリから始まるファイルツリー全体を指す場合
- ディスクスライスや他の記憶メディアデバイスのデータ構造を指す場合
- ファイルツリー構造のうち、ファイルがアクセスできるように主なファイルツリー上のマウントポイントに接続されている部分を指す場合

通常、その意味は状況に応じて判断できます。

Solaris オペレーティング環境は、各種ファイルシステムへの標準インタフェースを提供する「仮想ファイルシステム」(VFS) アーキテクチャを使用します。VFS アーキテクチャを使用すると、カーネルはファイルの読み取り、書き込み、一覧表示などの基本操作を処理でき、新しいファイルシステムの追加が簡単になります。

ファイルシステムのタイプ

Solaris システムソフトウェアでは、次の3種類のファイルシステムがサポートされます。

- ディスクベースのファイルシステム
- ネットワークベースのファイルシステム
- 仮想ファイルシステム

ファイルシステムのタイプを確認するには、569 ページの「ファイルシステムのタイプを調べる」を参照してください。

ディスクベースのファイルシステム

ディスクベースのファイルシステムは、ハードディスク、CD-ROM、フロッピーディスクなどの物理メディアに格納されます。ディスクベースのファイルシステムは、さまざまな形式で作成できます。利用できる形式は次のとおりです。

ディスクベースのファイルシステム	形式の説明
UFS	<p>UNIX ファイルシステム (4.3 Tahoe リリースに組み込まれていた BSD Fast File システム)。UFS は、Solaris オペレーティング環境のデフォルトのディスクベースファイルシステムです。</p> <p>UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクのフォーマットとスライスへの分割については、第 32 章を参照してください。</p>
HSFS	<p>High Sierra と ISO 9660 のファイルシステム。High Sierra は、初めての CD-ROM ファイルシステムです。ISO 9660 は、High Sierra ファイルシステムの公式の標準バージョンです。HSFS ファイルシステムは CD-ROM 上で使用される読み取り専用ファイルシステムです。Solaris HSFS では、ISO 9660 への Rock Ridge 拡張がサポートされるので、CD-ROM 上でも、すべての UFS ファイルシステムのセマンティクスとファイルタイプを提供します。ただし、書き込み可能ではなく、ハードリンクも提供しません。</p>
PCFS	<p>PC ファイルシステム。DOS ベースのパーソナルコンピュータ用に作成された DOS フォーマットのディスク上のデータとプログラムに読み取りと書き込みのアクセスができます。</p>
UDF	<p>UDF (Universal Disk Format) ファイルシステム。DVD (Digital Versatile Disc または Digital Video Disc) と呼ばれる光学式メディアテクノロジーに情報を格納するための業界標準形式。</p>

ディスクベースの各種ファイルシステムは、次のように特定のメディアのタイプに対応しています。

- UFS とハードディスク
- HSFS と CD-ROM
- PCFS とフロッピーディスク
- UDF と DVD

ただし、上記以外の組み合わせも可能です。たとえば、CD-ROM やフロッピーディスクにも、UFS ファイルシステムを格納できます。

ネットワークベースのファイルシステム

ネットワークベースのファイルシステムは、ネットワークからアクセスされるファイルシステムです。一般に、ネットワークベースのファイルシステムは 1 つのシステム上 (通常はサーバー上) にあり、他のシステムからネットワーク経由でアクセスされません。

NFS で分散された「リソース」(ファイルやディレクトリ)を管理するには、サーバーからそれらのリソースをエクスポートして個々のクライアントシステムでマウントします。詳細については、568 ページの「NFS 環境」を参照してください。

仮想ファイルシステム

仮想ファイルシステムは、特殊なカーネル情報と機能へのアクセスを提供するメモリーベースのファイルシステムです。ほとんどの仮想ファイルシステムは、ディスク領域を使用しません。ただし、CacheFS ファイルシステムは、ディスク上のファイルシステムを使用してキャッシュを保持します。また、一時ファイルシステム (TMPFS) などの一部の仮想ファイルシステムは、ディスク上のスワップ空間を使用します。

CacheFS ファイルシステム

CacheFS™ ファイルシステムを使用すると、リモートファイルシステムや、CD-ROM ドライブのような低速デバイスのパフォーマンスを改善できます。ファイルシステムをキャッシュすると、リモートファイルシステムや CD-ROM から読み込まれたデータは、ローカルシステム上のキャッシュに格納されます。

NFS や CD-ROM ファイルシステムのパフォーマンスとスケーラビリティを向上させるには、CacheFS ファイルシステムを使用してください。CacheFS ソフトウェアは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティを改善する汎用ファイルシステムキャッシュメカニズムです。

CacheFS ソフトウェアは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ソフトウェアはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。また、CacheFS ファイルシステムと AutoFS サービスを組み合わせると、パフォーマンスとスケーラビリティをさらに向上させることができます。

CacheFS ファイルシステムの詳細については、第 41 章を参照してください。

UDF (Universal Disk Format) ファイルシステム

UDF ファイルシステムは、「DVD」(Digital Versatile Disc または Digital Video Disc) 光学式メディアに情報を格納するための業界標準形式です。

UDF ファイルシステムは、SPARC と x86 の両方のプラットフォームにおいて、動的に読み込み可能な 32 ビットと 64 ビットのモジュールとして提供されます。また、ファイルシステムを作成、マウント、および検査するシステム管理ユーティリティも同時に提供されます。Solaris の UDF ファイルシステムは、サポートされている ATAPI と SCSI の DVD ドライブ、CD-ROM デバイス、ハードディスク、およびフロッピーディスクドライブで機能します。さらに、Solaris の UDF ファイルシステムは UDF 1.50 仕様に完全に準拠しています。

UDF ファイルシステムには次のような機能があります。

- UDF ファイルシステムが入っている業界標準の CD-ROM や DVD-ROM のメディアにアクセスできます。

- さまざまなプラットフォームやオペレーティングシステムで情報を交換できます。
- UDF 形式に基づく DVD ビデオ仕様を使用することで、相互対話性が豊富な放送品質並みの映像や高品質のサウンドを持つ新しいアプリケーションを実装できます。

次の機能は、UDF ファイルシステムにはありません。

- CD-RW (何度でも書き換えが可能な CD メディア) と DVD-RAM への順次ディスク同時記録方式と増分記録方式での書き込み
- UDF 1.50 仕様の一部ではないディスク割り当て、ACL、トランザクションのロギング、ファイルシステムのロック、およびファイルシステムのスレッドなど UFS 構成要素

次に、UDF ファイルシステムの要件を示します。

- Solaris 7 11/99、Solaris 8、または Solaris 9 リリースで動作していること
- SPARC または x86 プラットフォームがサポートされていること
- CD-ROM または DVD-ROM ドライブがサポートされていること

Solaris で実装された UDF ファイルシステムには、次のような互換性があります。

- 業界標準の読み取り / 書き込み UDF バージョン 1.50 のサポート
- 完全に国際化されたファイルシステムのユーティリティ

一時ファイルシステム

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べてアクセス速度が高速です。TMPFS を使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドを軽減でき、システムパフォーマンスを改善できます。たとえば、プログラムをコンパイルすると一時ファイルが作成されます。オペレーティングシステムは、これらのファイルを処理する間に大量のディスク処理やネットワーク処理を行います。TMPFS を使用してこれらの一時ファイルを格納すると、その作成、処理、または削除が大幅に高速になります。

TMPFS ファイルシステムのファイルは、永続的に保存されるわけではありません。ファイルシステムのマウントが解除されるときと、システムがシャットダウンまたはリブートされるときに、一時ファイルシステムのファイルは削除されます。

TMPFS は、Solaris オペレーティング環境内の /tmp ディレクトリのデフォルトのファイルシステムです。UFS ファイルシステムの場合と同様に、/tmp ディレクトリとの間でファイルをコピーまたは移動できます。

TMPFS ファイルシステムは、一時的な退避場所としてスワップ空間を使用します。TMPFS ファイルシステムがマウントされたシステムのスワップ空間が足りないと、次の 2 つの問題が発生する可能性があります。

- TMPFS ファイルシステムは、通常のファイルシステムと同様に容量不足になる可能性がある。

- TMPFS はスワップ空間を割り当ててファイルのデータを保存するので (必要な場合)、一部のプログラムがスワップ空間不足のために実行できなくなる可能性がある。

TMPFS ファイルシステムの作成方法については、第 39 章を参照してください。スワップ空間を拡張する方法については、第 42 章を参照してください。

ループバックファイルシステム

ループバックファイルシステム (LOFS) を使用すると、代替パス名を使用してファイルにアクセスできるように、新しい仮想ファイルシステムを作成できます。たとえば、ルート (/) のループバックマウントを /tmp/newroot 上で作成できます。ファイルシステム階層全体が、NFS サーバーからマウントされるファイルシステムを含め、/tmp/newroot 上に複製されたように見えます。どのファイルにも、ルート (/) で始まるパス名または /tmp/newroot で始まるパス名を使用してアクセスできます。

LOFS ファイルシステムの作成方法については、第 39 章を参照してください。

プロセスファイルシステム

プロセスファイルシステム (PROCFS) はメモリー内にあり、/proc ディレクトリには、アクティブなプロセスのプロセス番号別リストが入っています。/proc ディレクトリの内容は、ps などのコマンドで使用されます。デバッガや他の開発ツールも、ファイルシステムコールを使用して、プロセスのアドレス空間にアクセスできます。



注意 - /proc ディレクトリ内のファイルは削除しないでください。/proc ディレクトリからプロセスを削除しても、そのプロセスは強制終了されません。/proc ファイルはディスク容量を消費しないため、このディレクトリからファイルを削除してもあまり意味がありません。

/proc ディレクトリは、管理が不要です。

その他の仮想ファイルシステム

次のタイプの仮想ファイルシステムは、参考のために掲載してあります。管理は不要です。

仮想ファイルシステム	説明
FIFOFS (先入れ先出し)	プロセスにデータへの共通アクセス権を与える名前付きパイプのファイル
FDFS (ファイル記述子)	開いているファイルに、記述子を使用して名前を明示的に与える
MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供する
NAMEFS	ほとんどの場合、ファイル記述子をファイルの先頭に動的にマウントするために STREAMS に使用される
SPECFS (特殊)	キャラクタ型特殊デバイスとブロック型特殊デバイスへのアクセスを提供する
SWAPFS	カーネルがスワッピングに使用するファイルシステム

ファイルシステム管理用のコマンド

ほとんどのファイルシステム管理コマンドには、汎用コマンドとファイルシステムに固有のコマンドの2種類があります。可能な場合には、常に汎用コマンドを使用してください。汎用コマンドは、ファイルシステム固有のコマンドを呼び出します。次の表に、ファイルシステム管理用の汎用コマンドを示します。これらのコマンドは、`/usr/sbin` ディレクトリに入っています。

表 38-1 ファイルシステム管理用の汎用コマンド

コマンド	マニュアルページ	説明
<code>clri</code>	<code>clri (1M)</code>	i ノードをクリアする。
<code>df</code>	<code>df (1M)</code>	空きディスクブロック数とファイル数を出力する。
<code>ff</code>	<code>ff (1M)</code>	ファイルシステムのファイル名と統計情報を表示する。
<code>fsck</code>	<code>fsck (1M)</code>	ファイルシステムの整合性を検査し、検出された損傷を修復する。

表 38-1 ファイルシステム管理用の汎用コマンド (続き)

コマンド	マニュアルページ	説明
fsdb	fsdb (1M)	ファイルシステムをデバッグする。
fstyp	fstyp (1M)	ファイルシステムのタイプを調べる。
labelit	labelit (1M)	テープにコピーするときに、ファイルシステムのラベルを表示または作成する (volcopy コマンド専用)。
mkfs	mkfs (1M)	新しいファイルシステムを作成する。
mount	mount (1M)	ローカルおよびリモートのファイルシステムをマウントする。
mountall	mountall (1M)	仮想ファイルシステムテーブル (/etc/vfstab) に指定されているすべてのファイルシステムをマウントする。
ncheck	ncheck (1M)	パス名とその i ノード番号のリストを生成する。
umount	mount (1M)	ローカルおよびリモートのファイルシステムをマウント解除する。
umountall	mountall (1M)	仮想ファイルシステムテーブル (/etc/vfstab) に指定されているすべてのファイルシステムをマウント解除する。
volcopy	volcopy (1M)	ファイルシステムのイメージコピーを作成する。

ファイルシステムコマンドによるファイルシステムタイプの判断

汎用ファイルシステムコマンドは、次の順序でファイルシステムのタイプを判断します。

1. 指定されている場合は、-F オプションで指定されているファイルシステムのタイプから判断します。
2. 特殊デバイスを /etc/vfstab ファイルのエントリと突き合わせて判断します (special が指定されている場合)。たとえば fsck は、まず fsck device フィールドと突き合わせて一致するエントリを検索します。一致するエントリが見つからなければ、特殊デバイスフィールドと突き合わせてチェックします。
3. ローカルファイルシステムの場合は /etc/default/fs ファイル内に指定されたデフォルトを使用し、リモートファイルシステムの場合は /etc/dfs/fstypes ファイル内に指定されたデフォルトを使用して判断します。

汎用コマンドと専用コマンドのマニュアルページ

汎用コマンドと専用コマンドについては、『*man pages section 1M: System Administration Commands*』を参照してください。ファイルシステムの汎用コマンドのマニュアルページには、汎用コマンドオプションに関する情報だけが記載されています。専用コマンドのマニュアルページには、該当するファイルシステムに固有のオプション情報が記載されています。特定のマニュアルページを見つけるには、汎用コマンド名の末尾にアンダースコアとファイルシステムタイプの略称を追加してください。たとえば、UFS ファイルシステムのマウントについてのマニュアルページを参照するには、次のように入力します。

```
$ man mount_ufs
```

デフォルトの Solaris ファイルシステム

Solaris UFS ファイルシステムは階層構造になっており、ルートディレクトリ (/) から始まり、下位に多数のディレクトリが形成されています。Solaris のインストールプロセスは、デフォルトのディレクトリセットをインストールし、一連の規則を適用して類似するタイプのファイルをグループ化します。次の表に、デフォルトの Solaris ファイルシステムの概要を示します。

表 38-2 デフォルトの Solaris ファイルシステム

ファイルシステムまたはディレクトリ	ファイルシステムのタイプ	説明
ルート (/)	UFS	階層ファイルツリーの最上位。ルートディレクトリには、カーネル、デバイスドライバ、システムのブートに使用されるプログラムなど、システム処理に欠かせないディレクトリとファイルが入っている。また、ローカルとリモートのファイルシステムをファイルツリーに接続できるマウントポイントディレクトリも入っている。
/usr	UFS	他のユーザーと共有できるシステムファイルとディレクトリ。特定のタイプのシステム上でのみ実行できるファイルは、/usr ファイルシステムに入っている (SPARC 実行可能ファイルなど)。どのタイプのシステム上でも使用できるファイル (マニュアルページなど) は、/usr/share ディレクトリに入っている。

表 38-2 デフォルトの Solaris ファイルシステム (続き)

ファイルシステムまたはディレクトリ	ファイルシステムのタイプ	説明
/export/home または /home	NFS、UFS	ユーザーのホームディレクトリのマウントポイント。ホームディレクトリには、そのユーザーの作業ファイルが格納される。デフォルトでは、/home ディレクトリは自動マウントされるファイルシステムである。スタンドアロンシステムでは、/home ディレクトリはローカルディスクスライス上の UFS ファイルシステムの場合がある。
/var	UFS	ローカルシステムの使用中に変化または拡大する可能性のあるシステムファイルとディレクトリ。これには、システムログ、vi と ex のバックアップファイル、および uucp ファイルが含まれる。
/opt	NFS、UFS	オプションの Sun 以外のソフトウェア製品のマウントポイント。一部のシステムでは、/opt ディレクトリはローカルディスクスライス上の UFS ファイルシステムの場合がある。
/tmp	TMPFS	システムがブートされるか、/tmp ファイルシステムがマウント解除されるたびに消去される一時ファイル。
/proc	PROCFS	アクティブなプロセスの番号別リスト。
/etc/mnttab	MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供するファイルシステム。
/var/run	TMPFS	システムのブート後に不要になる一時ファイルを格納するためのファイルシステム。

システムを動作させるには、ルート (/) と /usr のファイルシステムが必要です。/usr ファイルシステムに置かれている最も基本的なコマンドの一部 (mount など) は、システムのブート時や、システムがシングルユーザーモードで実行しており、/usr ファイルシステムがマウントされていない場合でも使用できるように、ルート (/) ファイルシステムにも置かれています。ルート (/) と /usr ファイルシステムのデフォルトディレクトリの詳細については、第 44 章を参照してください。

スワップ空間

Solaris オペレーティング環境は、一部のディスクスライスをファイルシステムではなく一時記憶域として使用します。これらのスライスを「スワップスライス」または「スワップ空間」と呼びます。スワップスライスは、現在のプロセスを処理するだけの十分な物理メモリーがシステムにない場合に、仮想メモリー記憶域として使用されます。

多くのアプリケーションは十分なスワップ空間が使用できることを前提に作成されているため、スワップ空間を割り当て、その使われ方を監視して、必要に応じてスワップ空間を追加する方法を知っておく必要があります。スワップ空間の概要とスワップ空間を追加する手順については、第 42 章を参照してください。

UFS ファイルシステム

UFS は、Solaris オペレーティング環境内のデフォルトのディスクベースファイルシステムです。ほとんどの場合、ディスクベースのファイルシステムを管理するときには、UFS を管理していることとなります。UFS ファイルシステムの機能は次の通りです。

UFS の機能	説明
状態フラグ	ファイルシステムの状態を、クリーン、安定、使用中、ロギング処理、または不明として示します。これらのフラグによって、必要のないチェックをファイルシステム上で行わなくて済みます。ファイルシステムが「クリーン」状態、「安定」状態、または「ロギング処理」状態になっていると、ファイルシステムのチェックは実行されません。
拡張基礎タイプ (EFT)	32 ビットのユーザー ID (UID)、グループ ID (GID)、およびデバイス番号。
大規模ファイルシステム	最大 16T バイトまで増やせるファイルシステムに、およそ 1T バイトの大きさのファイルを指定できます。EFI ディスクラベルを使ってディスク上にマルチテラバイトの UFS ファイルシステムを作成できます。

UFS ファイルシステム構造の詳細については、第 44 章を参照してください。

UFS ロギング

UFS ロギングは、個々の UFS 操作による複数のメタデータ変更を、1 つのトランザクションにまとめます。一連のトランザクションは、ディスク上のログに記録された後で、実際の UFS ファイルシステムのメタデータに適用されます。

システムはリブート時に、不完全なトランザクションを廃棄しますが、完結している操作のトランザクションは適用します。完結しているトランザクションだけが適用されるために、ファイルシステムの整合性が保たれます。通常であればシステムコールの実行が中断され、UFS ファイルシステムの整合性が確保できないシステムクラッシュ時にも、ファイルシステムの整合性が保たれます。

UFS ロギングには 2 つの長所があります。

- トランザクションログによりファイルシステムの整合性がすでに保持されている場合、システムがクラッシュしたとき、またはクリーンでないシステム停止が発生したときに、`fsck` コマンドを実行する必要がないことがあります。クリーンでないシステム停止については、642 ページの「`fsck` コマンドでチェックして修復される内容」を参照してください。
- 通常の場合、パフォーマンスが大幅に改善されます。というのも、ロギングを有効にしたファイルシステムでは、同じデータに対する複数回の更新が 1 回の更新に変換されるため、オーバーヘッドの大きいディスク操作の実行回数が減るからです。

ログはファイルシステムの空きブロックから割り当てられ、1G バイトのファイルシステムごとに約 1M バイトのサイズ (合計で 64M バイトまで) が割り当てられます。ログは、いっぱいになるとフラッシュされます。また、ファイルシステムがマウント解除されたとき、あるいは `lockfs` コマンドを実行したときにも、ログがフラッシュされます。

1T バイトを超えるファイルシステムの場合、UFS ロギングがデフォルトで有効になります。

UFS ロギングを有効にする必要がある場合は、`/etc/vfstab` ファイル内で `mount` コマンドに `-o logging` オプションを指定する、またはファイルシステムを手動でマウントするときに `mount` コマンドに `-o logging` オプションを指定します。ログは、任意の UFS ファイルシステムで有効にできます。また、`fsdb` コマンドには、UFS ロギングをサポートするための新しいデバッグコマンドが用意されています。

一部のオペレーティングシステムでは、ロギングが有効になっているファイルシステムを「ジャーナル」ファイルシステムと呼びます。

UFS ファイルシステムの計画

ファイルシステムの配置を決めるときには、要求が競合する可能性があることを考える必要があります。次にいくつかの推奨事項を示します。

- 作業負荷を異なる入出力システムやディスクドライブ間でできるだけ均等に分散します。`/export/home` ファイルシステムやスワップ空間を異なるディスク間で均等に分散させます。
- プロジェクトの個々の部分やグループのメンバーを同じファイルシステム内に入れます。
- 1 ディスク当たりのファイルシステム数をできるだけ少なくします。通常、システム (またはブート) ディスク上には、ルート (`/`)、`/usr`、スワップ空間の 3 つのファイルシステムがあります。他のディスクには、1 つまたは多くても 2 つのファイルシステム (1 つはなるべく追加スワップ空間) を作成します。多数の小規模なファイルシステムに分割しすぎるよりもファイルシステム数を少なくして余地を設ける方が、ファイルがフラグメントに分割される可能性が小さくなります。容量の大きいテープドライブを使用し、`ufsdump` コマンドで複数のボリュームを処理できるようにしておけば、大規模なファイルシステムでも簡単にバックアップをと

ることができます。

- 絶えずきわめて小さいファイルを作成するユーザーがいる場合は、i ノード数を増やして別のファイルシステムを作成することを検討します。ただし、ほとんどのサイトでは、類似するタイプのファイルを同じファイルシステム内で保管する必要はありません。

デフォルトのファイルシステムパラメータや、新しい UFS ファイルシステムを作成する手順については、第 39 章を参照してください。

UFS 直接入出力

直接入出力の目的は、大容量入出力処理のスピードを速くすることです。大容量入出力処理では、大規模ファイル (256K バイトを超える) を転送するために、大容量のバッファサイズを使用します。

UFS の直接入出力を使用すると、データベースエンジンなど、独自の内部バッファリングを行うアプリケーションにメリットがあります。Solaris 8 1/01 リリースから開始された UFS の直接入出力は、raw デバイスのアクセス時に見られる同様の入出力の並行処理に対応するよう改善されました。現在では、パフォーマンスがわずかに低下するだけで、ファイルシステムのネーミングや柔軟性がもたらすメリットを受けることができます。データベースの製造元を調べて、その製品構成オプションで UFS の直接入出力を有効にできるかどうかを確認してください。

mount コマンドに `forcedirectio` オプションを使用しても、直接入出力をファイルシステムで有効にできます。直接入出力を有効にしてパフォーマンスが向上するのは、ファイルシステムが大量の連続するデータを転送するときだけです。

`forcedirectio` オプションでファイルシステムをマウントするとき、データはユーザーのアドレス空間とディスクの間で直接伝送されます。直接入出力がファイルシステムで無効な場合、ユーザーのアドレス空間とディスクの間で転送されるデータは、まず、カーネルアドレス空間にバッファされます。

デフォルトでは、UFS ファイルシステムでは直接入出力は行われません。詳細については、`mount_ufs(1M)` のマニュアルページを参照してください。

ファイルシステムのマウントとマウント解除

ファイルシステム上のファイルにアクセスするには、ファイルシステムをマウントする必要があります。ファイルシステムをマウントするときは、ファイルシステムをディレクトリ (マウントポイント) に接続し、システムで使用可能にすることです。ルート (/) ファイルシステムは、常にマウントされています。他のファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりできます。

ファイルシステムをマウントすると、そのファイルシステムがマウントされている間は、マウントポイントのディレクトリ内に実際に存在しているファイルやディレクトリは使用できなくなります。これらのファイルはマウント処理の影響を永続的に受けるわけではなく、ファイルシステムをマウント解除すると再び使用できるようになります。通常は存在はするがアクセスできないファイルは混乱の原因になるので、マウントディレクトリを空にしておきます。

たとえば、次の図は、ルート (/) ファイルシステムと、その下の `sbin`、`etc`、`opt` の各サブディレクトリから始まるローカルファイルシステムを示しています。

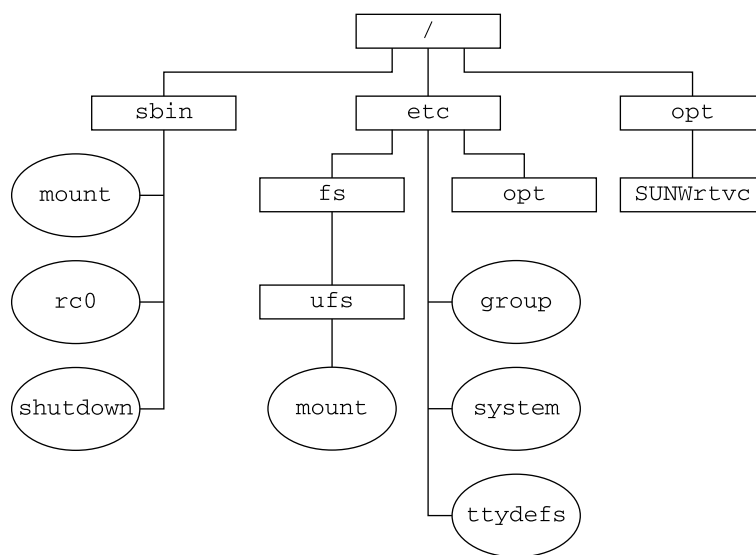
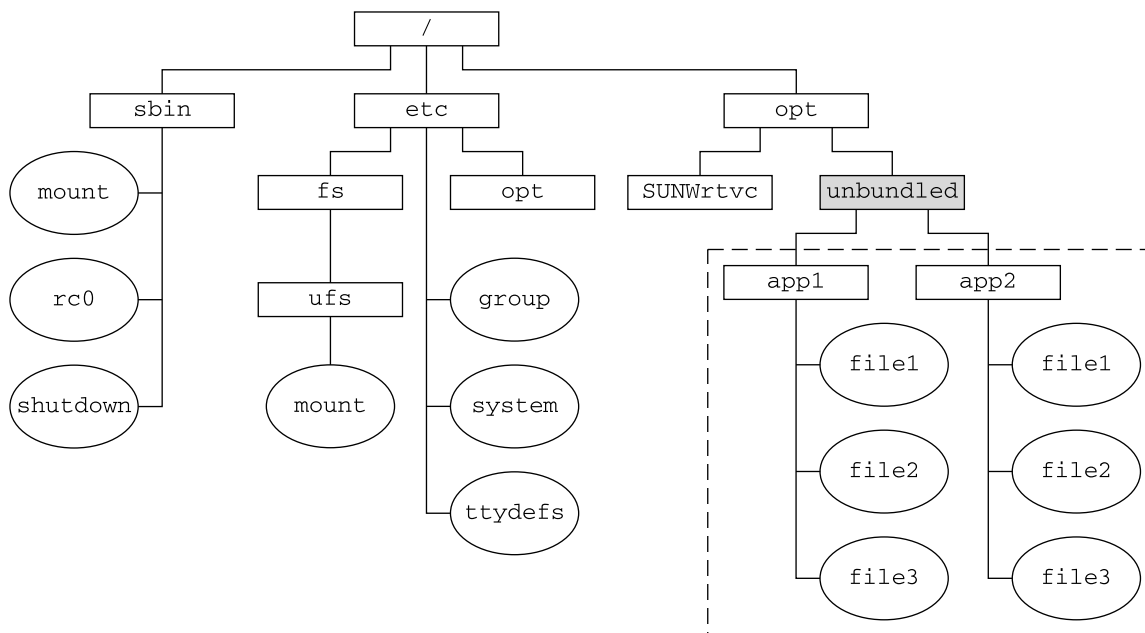


図 38-1 ルート (/) ファイルシステムの例

別パッケージの製品が含まれている `/opt` ファイルシステムからローカルファイルシステムにアクセスするには、次の作業を行います。

- まず、ファイルシステムをマウントするためのマウントポイントとして使用するディレクトリを作成しなければなりません (たとえば、`/opt/unbundled`)。
- マウントポイントを作成すると、`mount` コマンドでファイルシステムをマウントでき、`/opt/unbundled` 内のすべてのファイルとディレクトリにアクセスできるようになります (次の図を参照)。

ファイルシステムのマウント方法については、第 40 章を参照してください。



□ マウントポイント

[] ファイルシステム

図 38-2 ファイルシステムのマウント

マウントされたファイルシステムテーブル

ファイルシステムをマウントまたはマウント解除すると、現在マウントされているファイルシステムのリストを使用して、`/etc/mnttab` (マウントテーブル) ファイルが変更されます。このファイルの内容は `cat` または `more` コマンドを使用して表示できますが、編集することはできません。次にマウントテーブルファイル `/etc/mnttab` の例を示します。

```

$ more /etc/mnttab
/dev/dsk/c0t0d0s0 / ufs rw,intr,largefiles,onerror=panic,suid,dev=2200000 938557523
/proc /proc proc dev=3180000 938557522
fd /dev/fd fd rw,suid,dev=3240000 938557524
mnttab /etc/mnttab mntfs dev=3340000 938557526
swap /var/run tmpfs dev=1 938557526
swap /tmp tmpfs dev=2 938557529
/dev/dsk/c0t0d0s7 /export/home ufs rw,intr,largefiles,onerror=panic,suid,dev=2200007 ...
$
  
```

仮想ファイルシステムテーブル

アクセスするたびにファイルシステムを手動でマウントするのは、時間がかり、またまちがいが起こりやすい作業です。この問題を解決するために、仮想ファイルシステムテーブル (/etc/vfstab ファイル) にファイルシステムのリストとそのマウント方法を指定できます。

/etc/vfstab ファイルは、2つの重要な機能を持っています。

- 1つは、システムブート時に自動的にマウントするファイルシステムを指定できることです。
- もう1つは、マウントポイント名だけでファイルシステムをマウントできることです。これは、/etc/vfstab ファイルにマウントポイントと実際のデバイススライス名とのマッピングを指定することができるためです。

デフォルトの /etc/vfstab ファイルは、システムをインストールするときに作成され、その内容はシステムソフトウェアをインストールするときに行なった選択によって異なります。ただし、システムの /etc/vfstab ファイルはいつでも編集できます。エントリを追加するときに指定する必要がある主な情報は、ファイルシステムが置かれているデバイス、マウントポイントの名前、ファイルシステムのタイプ、システムブート時に自動的にマウントするかどうか (mountall コマンドを使用する)、およびマウントオプションです。

次の例は、/etc/vfstab ファイルの内容を示しています。コメント行は # で始まります。この例は、2つのディスク (c0t0d0 と c0t3d0) を持つシステムの /etc/vfstab ファイルを示しています。

```
$ more /etc/vfstab
#device      device      mount      FS      fsck  mount  mount
#to mount    to fsck     point      type    pass  at boot options
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /
/proc        -           /proc      proc    -     no    -
/dev/dsk/c0t0d0s1 -           -          swap    -     no    -
swap         -           /tmp       tmpfs   -     yes   -
/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /usr       ufs     2     no    -
/dev/dsk/c0t3d0s7 /dev/rdsk/c0t3d0s7 /test     ufs     2     yes   -
$
```

上の例の最後のエントリは、システムブート時に /dev/dsk/c0t3d0s7 スライス上の UFS ファイルシステムが自動的にマウントポイント /test にマウントされることを示しています。ルート (/) と /usr では、mount at boot フィールドの値が no に指定されていることに注意してください。これはこの2つのファイルシステムが、mountall コマンドを実行する前にブートシーケンスの一部としてカーネルによってマウントされるからです。

/etc/vfstab の各フィールドの説明、およびこのファイルの編集/使用方法については、第 40 章を参照してください。

NFS 環境

NFS は、1つのシステム (通常はサーバー) の「リソース」 (ファイルやディレクトリ) をネットワーク上の他のシステムと共有するための分散型ファイルシステムサービスです。たとえば、Sun 以外のアプリケーションやソースファイルを他のシステム上のユーザーと共有できます。

NFS は、リソースの実際の物理的な位置をユーザーが意識しなくてすむようにします。よく使用されるファイルのコピーをシステムごとに配置しなくても、あるシステムのディスク上にコピーを1つ配置することによって NFS は、他のすべてのシステムがそのコピーにネットワークからアクセスできるようにします。NFS の環境では、リモートファイルシステムは、実際にはローカルシステムと区別が付きません。

システムは、ネットワーク上で共有するリソースがあるときに、NFS サーバーになります。サーバーは、現在共有されているリソースとそのアクセス制限 (読み取り / 書き込み、読み取り専用アクセスなど) のリストを管理します。

リソースを共有する場合は、リモートシステムにマウントできるように、そのリソースを使用可能な状態にします。

リソースを共有するには、次の方法があります。

- `share` コマンドまたは `shareall` コマンドを使用する
- `/etc/dfs/dfstab` (分散ファイルシステムテーブル) ファイルにエントリを追加し、システムをリポートする。

リソースを共有する方法については、第 40 章を参照してください。NFS の詳細については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「ネットワークファイルシステムの管理 (概要)」を参照してください。

自動マウントまたは AutoFS

NFS ファイルシステムリソースをマウントするには、自動マウント (または AutoFS) というクライアント側のサービスを使用します。このサービスによってシステムは、ユーザーがアクセスしたときに自動的に NFS リソースをマウントまたはマウント解除できます。ユーザーがファイルシステム内のファイルを使用している間、ファイルシステムリソースはマウントされたままになります。リソースが一定の時間アクセスされなかった場合、リソースは自動的にマウント解除されます。

次に、AutoFS の特徴を示します。

- システムブート時に NFS リソースをマウントする必要がないために、ブート時間が短くなります。
- NFS リソースをマウントまたはマウント解除するために、スーパーユーザーのパスワードを知っている必要はありません。
- NFS リソースは使用されるときにだけマウントされるために、ネットワークトラフィックが軽減されます。

AutoFS サービスは automount ユーティリティによって初期化されます。このコマンドは、システムのブート時に自動的に実行されます。automountd デーモンは永続的に動作し、必要に応じて NFS ファイルシステムをマウントまたはマウント解除します。デフォルトでは、/home ファイルシステムは automount デーモンによってマウントされます。

AutoFS を使用すると、同じファイルシステムを提供するサーバーを指定できます。このような方法では、1つのサーバーがダウンしても、AutoFS が他のマシンからファイルシステムをマウントすることができます。

AutoFS の設定と管理方法については、『Solaris のシステム管理 (IP サービス)』を参照してください。

ファイルシステムのタイプを調べる

ファイルシステムのタイプは、次のいずれかを使用して判断できます。

- 仮想ファイルシステムテーブル (/etc/vfstab ファイル) 内の FS type フィールド
- ローカルファイルシステムの /etc/default/fs ファイル
- NFS ファイルシステムの /etc/dfs/fstypes ファイル

ファイルシステムのタイプを調べる方法

この手順は、ファイルシステムがマウントされているかどうかにかかわらず、使用できます。

grep コマンドを使用して、ファイルシステムのタイプを調べます。

```
$ grep mount-point fs-table
```

mount-point

ファイルシステムタイプを調べるファイルシステムのマウントポイント名を指定する。たとえば、/var ディレクトリ

fs-table

ファイルシステムのタイプを調べるファイルシステムテーブルへの絶対パスを示す。ファイルシステムがマウントされている場合、*fs-table* は /etc/mnttab。マウントされていない場合、*fs-table* は /etc/vfstab

マウントポイントの情報が表示されます。

注 - ディスクスライスの raw デバイス名がわかる場合、fstyp コマンドで、ファイルシステムのタイプを調べることができます (そのディスクスライスにファイルシステムが含まれている場合)。詳細については、fstyp (1M) のマニュアルページを参照してください。

例 — ファイルシステムのタイプを調べる

次の例では、/etc/vfstab を使用して、/export ファイルシステムのタイプを調べます。

```
$ grep /export /etc/vfstab
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /export ufs 2 yes -
$
```

次の例では、/etc/mnttab ファイルを使用して、現在マウントされているフロッピーディスク (vold でマウントされたもの) のファイルシステムのタイプを調べます。

```
$ grep /floppy /etc/mnttab
/vol/dev/diskette0/unnamed_floppy /floppy/unnamed_floppy pcfs rw,
nohidden,nofoldcase,dev=16c0009 89103376
$
```

第 39 章

ファイルシステムの作成 (手順)

この章では、UFS ファイルシステム、一時ファイルシステム (TMPFS)、およびループバックファイルシステム (LOFS) の作成方法について説明します。UFS ファイルシステムについては、`newfs (1M)` コマンドを使用してファイルシステムをハードディスク上に作成する方法を示します。TMPFS と LOFS は仮想ファイルシステムであるため、これらのファイルシステムを「実際に使用する」には、ファイルシステムをマウントします。

この章で説明する手順は次のとおりです。

- 573 ページの「UFS ファイルシステムを作成する方法」
- 575 ページの「TMPFS ファイルシステムを作成する方法」
- 576 ページの「LOFS ファイルシステムを作成する方法」

注 - UFS と DOS のファイルシステムをリムーバブルメディア上に作成する手順については、第 17 章を参照してください。

UFS ファイルシステムの作成

UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクスライスとは物理的なディスクのサブセットで、連続するブロックからなる 1 つの範囲のことです。スライスはスワップ空間などの raw デバイスとして使用することも、ディスクベースのファイルシステムとして使用することもできます。ディスクのフォーマットとディスクのスライスへの分割の詳細については、第 32 章を参照してください。

Solaris ボリュームマネージャなどのボリューム管理製品を使用すると、精錬された (単一のスライスや単一のディスク境界を拡張する) 「ボリューム」を作成できます。ボリュームの使用方法については、『Solaris ボリュームマネージャの管理』を参照してください。

注 - Solaris のデバイス名は、用語「スライス (デバイス名内の文字は s)」を使用して、スライス番号を参照します。スライスは「パーティション」と呼ばれていました。

UFS ファイルシステムは、インストール手順の一部として Solaris オペレーティング環境によって自動的に作成されるので、UFS ファイルシステムを作成しなければならないことはほとんどありません。次の場合には、UFS ファイルシステムを作成する (または作成し直す) 必要があります。

- ディスクを追加または交換する場合
- 既存のパーティション構造を変更する場合
- ファイルシステム全体を復元する場合

`newfs` コマンドを使用するのが、UFS ファイルシステムを作成する標準的な方法です。`newfs` コマンドは `mkfs` コマンドの使いやすいフロントエンドであり、新しいファイルシステムを作成します。`newfs` コマンドのパラメータ (デフォルトでは 1 シリンダ当たりのトラック数や 1 トラック当たりのセクター数など) は、新しいファイルシステムが作成されるディスクのラベルから読み取られます。選択したオプションは、`mkfs` コマンドに渡され、ファイルシステムが作成されます。

newfs コマンドのデフォルトのパラメータ

ディスクスライス上に新しいファイルシステムを作成するには、ほとんどの場合に `newfs` コマンドを使用します。次の表に、`newfs` コマンドで使用するデフォルトのパラメータを示します。

パラメータ	デフォルト値
ブロックサイズ	8K バイト
フラグメントサイズ	1K バイト
最小空き領域	$((64\text{M バイト} / \text{パーティションサイズ}) * 100)$ で算出した値を最も近い整数に切り捨てる。値は、パーティションサイズの 1% から 10% の範囲に制限される
回転待ち	0
最適化タイプ	時間
i ノード数	2K バイトのディスク領域ごとに 1 個

▼ UFS ファイルシステムを作成する方法

1. 次の前提条件を満たしているかどうかを確認します。
 - a. ディスクをフォーマットし、スライスに分割しておかなければならない。
ディスクのフォーマットとスライスへの分割については、第 32 章を参照してください。
 - b. ファイルシステムを格納するスライスのデバイス名を知っていなければならない。
ディスク番号とディスクスライス番号を調べる方法については、第 33 章を参照してください。
 - c. 既存の **UFS** ファイルシステムを作成し直す場合は、そのマウントを解除する。
 - d. スーパーユーザーになるか、同等の役割を引き受けていなければならない。
2. ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdisk/device-name
```

-N	newfs コマンドが mkfs コマンドに渡すパラメータを表示する。ファイルシステムは実際に作成されない。newfs コマンドをテストするのに好ましい方法。
-b size	ファイルシステムのブロックサイズを 1 ブロックあたり 4096 または 8192 バイトで指定する。デフォルトは 8192 バイト。
-i bytes	i ノード 1 個当たりのバイト数を指定する。デフォルトはディスクのサイズによって異なる。詳細については、newfs(1M) のマニュアルページを参照。
device-name	新しいファイルシステムを作成するディスクデバイス名を指定する。

システムから、確認を促すプロンプトが表示されます。



注意 – この手順を実行する前に、スライスのデバイス名が正しく指定されていることを確認してください。間違ったスライスを指定すると、その内容は新しいファイルシステムの作成時に消去されます。そして、システムがパニックを起こす原因となる可能性があります。

3. **UFS** ファイルシステムが作成されていることを確認するには、新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

device-name 引数は、新しいファイルシステムを格納するディスクデバイスの名前を指定します。

`fsck` コマンドは、新しいファイルシステムの整合性をチェックして、問題があれば通知し、問題を修復する前にプロンプトを表示します。`fsck` コマンドの詳細については、第 43 章または `fsck(1M)` のマニュアルページを参照してください。

例 — UFS ファイルシステムを作成する

次の例は、`/dev/rdisk/c0t1d0s7` 上に UFS ファイルシステムを作成する方法を示しています。

```
# newfs /dev/rdisk/c0t1d0s7
/dev/rdisk/c0t1d0s7: 725760 sectors in 720 cylinders of 14 tracks, 72 sectors
354.4MB in 45 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
613376, 629584, 645792, 662000, 678208, 694416, 710624,
#
```

次に進む手順

UFS ファイルシステムをマウントし、使用可能にするには、第 40 章に進みます。

一時ファイルシステム (TMPFS) の作成

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べて読み書きの速度が高速です。TMPFS ファイルシステムを使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドが軽減されるのでシステムのパフォーマンスを向上できます。TMPFS ファイルシステム内のファイルは、リブートまたはマウント解除すると削除されます。

複数の TMPFS ファイルシステムを作成した場合は、すべてのファイルシステムが同じシステムリソースを使用するという点に注意してください。`mount` コマンドの `-o size` オプションを使用して TMPFS のサイズを制限しないと、ある TMPFS ファイルシステムで作成されたファイルが、他の TMPFS ファイルシステムのための領域を使い切ってしまう可能性があります。

詳細については、`tmpfs(7FS)` のマニュアルページを参照してください。

▼ TMPFS ファイルシステムを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、**TMPFS** ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir /mount-point
/mount-point は、TMPFS ファイルシステムがマウントされるディレクトリです。
```

3. **TMPFS** ファイルシステムをマウントします。

```
# mount -F tmpfs [-o size=number] swap mount-point
```

<code>-o size= number</code>	TMPFS ファイルシステムのサイズ制限を M バイト単位で示す。
<code>mount-point</code>	TMPFS ファイルシステムがマウントされるディレクトリを示す。

ブート時にシステムが TMPFS ファイルシステムを自動的にマウントするように設定するには、575 ページの「例 — ブート時に TMPFS ファイルシステムをマウントする」を参照してください。

4. **TMPFS** ファイルシステムが作成されていることを確認します。

```
# mount -v
```

例 — TMPFS ファイルシステムを作成する

次の例は、TMPFS ファイルシステム `/export/reports` を作成およびマウントし、そのサイズを 50M バイトに制限する方法を示しています。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50m swap /export/reports
```

例 — ブート時に TMPFS ファイルシステムをマウントする

ブート時にシステムが自動的に TMPFS ファイルシステムをマウントするように設定するには、`/etc/vfstab` のエントリを追加します。次の例は、システムのブート時に `/export/test` を TMPFS ファイルシステムとしてマウントする `/etc/vfstab` ファイルのエントリを示しています。 `size=number` オプションを指定していないため、`/export/test` の TMPFS ファイルシステムのサイズは利用できるシステムリソースによってのみ制限されます。

```
swap - /export/test tmpfs - yes -
```

/etc/vfstab ファイルの詳細については、583 ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。

ループバックファイルシステム (LOFS) の作成

LOFS ファイルシステムは、既存のファイルシステムへの代替パスを提供する仮想ファイルシステムです。他のファイルシステムを LOFS ループバックファイルシステムにマウントしても、元のファイルシステムは変化しません。

詳細については、`lofs` (7FS) のマニュアルページを参照してください。



注意 - LOFS ファイルシステムは慎重に作成してください。LOFS ファイルシステムは仮想ファイルシステムなので、ユーザーやアプリケーションを混乱させる可能性があります。

▼ LOFS ファイルシステムを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、**LOFS** ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir loopback-directory
```

3. 新しく作成したディレクトリに対して、適切なアクセス権と所有権を設定します。

4. 必要に応じて、**LOFS** ファイルシステムをマウントするマウントポイントを作成します。

```
# mkdir /mount-point
```

5. **LOFS** ファイルシステムをマウントします。

```
# mount -F lofs loopback-directory /mount-point
```

loopback-directory

ループバックマウントポイントにマウントするファイルシステムを指定する

`/mount-point`

LOFS ファイルシステムをマウントするディレクトリを指定する

6. LOFS ファイルシステムがマウントされていることを確認します。

```
# mount -v
```

例 — LOFS ファイルシステムを作成およびマウントする

次の例は、新しいソフトウェアを、実際にはインストールしないで、ループバックファイルシステムとして `/new/dist` ディレクトリに作成、マウント、およびテストする方法を示しています。

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

例 — ブート時に LOFS ファイルシステムをマウントする

ブート時にシステムが自動的に LOFS ファイルシステムをマウントするように設定するには、`/etc/vfstab` ファイルの最後にエントリを追加します。次の例は、ルート (`/`) ファイルシステムの LOFS ファイルシステムを `/tmp/newroot` にマウントする `/etc/vfstab` ファイルのエントリを示しています。

```
/ - /tmp/newroot lofs - yes -
```



注意 - ループバックファイルシステムのエントリは、`/etc/vfstab` ファイル内の最後のエントリでなければなりません。ループバックファイルシステムの `/etc/vfstab` エントリが、そこに組み込まれるファイルシステムよりも前にあると、ループバックファイルシステムをマウントできません。

`/etc/vfstab` ファイルの詳細については、583 ページの「`/etc/vfstab` ファイルのフィールドの説明」を参照してください。

第 40 章

ファイルシステムのマウントとマウント解除 (手順)

この章では、ファイルシステムをマウントしたり、マウント解除する方法について説明します。

この章で説明する手順は次のとおりです。

- 585 ページの「どのファイルシステムがマウントされているかを調べる方法」
- 585 ページの「/etc/vfstab ファイルにエントリを追加する方法」
- 586 ページの「1つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)」
- 588 ページの「UFS ファイルシステムのマウント方法 (mount コマンド)」
- 589 ページの「大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)」
- 590 ページの「NFS ファイルシステムのマウント方法 (mount コマンド)」
- 591 ページの「x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)」
- 593 ページの「ファイルシステムのマウント解除を確認する方法」
- 593 ページの「ファイルシステムを使用中のすべてのプロセスを終了させる方法」
- 594 ページの「1つのファイルシステムをマウント解除する方法」

ファイルシステムのマウントの概要

ファイルシステムを作成したら、そのファイルシステムをシステムで使用できるようにする必要があります。ファイルシステムを使用できるようにするには、マウントします。マウントしたファイルシステムは、システムのディレクトリツリー内の指定したマウントポイントに接続されます。ルート (/) ファイルシステムは、常にマウントされています。

次の表に、ファイルシステムをその用途に応じてマウントするためのガイドラインを示します。

必要なマウントの種類	推奨されるマウント方法
ローカルまたはリモートのファイルシステムをときどきマウントする	コマンド行から手動で <code>mount</code> コマンドを入力する。
ローカルのファイルシステムを頻繁にマウントする	<code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに、自動的にファイルシステムをマウントする。
リモートのファイルシステムを頻繁にマウントする (ホームディレクトリなど)	<ul style="list-style-type: none"> ■ <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザーモードでブートされたときに、自動的にファイルシステムをマウントする。 ■ AutoFS により、ユーザーがディレクトリにアクセスしたとき、あるいはディレクトリから移動したときに、自動的にファイルシステムをマウントまたはマウント解除する。 <p>パフォーマンスを向上させるために、CacheFS ファイルシステムを使用してリモートのファイルシステムをキャッシュに書き込むこともできる。</p>

ファイルシステムを含むメディアは、必要に応じてメディアをドライブに挿入し、`volcheck` コマンドを実行することによってマウントできます。リムーバブルメディアのマウント方法については、第 17 章を参照してください。

ファイルシステムのマウントとマウント解除に使用するコマンド

表 40-1 に、ファイルシステムのマウントとマウント解除に使用する `/usr/sbin` ディレクトリ内のコマンドを示します。

表 40-1 ファイルシステムのマウントとマウント解除に使用するコマンド

コマンド	マニュアルページ	説明
<code>mount</code>	<code>mount (1M)</code>	ファイルシステムとリモートリソースをマウントする。
<code>mountall</code>	<code>mountall (1M)</code>	<code>/etc/vfstab</code> ファイルに指定されているすべてのファイルシステムをマウントする。システムがマルチユーザーモードになるときに、 <code>mountall</code> コマンドが自動的に実行される。

表 40-1 ファイルシステムのマウントとマウント解除に使用するコマンド (続き)

コマンド	マニュアルページ	説明
umount	mount (1M)	ファイルシステムとリモートリソースをマウント解除する。
umountall	mountall (1M)	/etc/vfstab ファイルに指定されているすべてのファイルシステムをマウント解除する。

mount と mountall の各コマンドは、不整合が認められた読み取り / 書き込み用のファイルシステムはマウントしません。mount または mountall コマンドからエラーメッセージが表示される場合は、ファイルシステムをチェックする必要があります。ファイルシステムをチェックする方法については、第 43 章を参照してください。

umount と umountall の各コマンドは、使用中のファイルシステムはマウント解除しません。ファイルシステムは、次の場合に使用中であるとみなされます。

- ユーザーがファイルシステム内のファイルまたはディレクトリにアクセスしている場合
- プログラムがそのファイルシステム上にあるファイルをオープンしている場合
- ファイルシステムが共有されている場合

汎用マウントオプション

次の表に、mount -o オプションで指定できる汎用オプションを示します。複数のオプションを指定する場合は、コンマ (空白を入れない) で区切ります。たとえば、-o ro,nosuid のようになります。

各ファイルシステムタイプで指定可能なマウントオプションのリストについては、各マウントコマンドのマニュアルページ (mount_ufs (1M) など) を参照してください。

表 40-2 -o で指定する汎用マウントオプション

オプション	ファイルシステム	説明
bg fg	NFS	最初のマウントに失敗すると、バックグラウンド (bg) またはフォアグラウンド (fg) で再試行する。このオプションは重要でない vfstab エントリには安全である。デフォルトは fg。
hard soft	NFS	サーバーが応答しない場合の手続きを指定する。soft オプションは、エラーが返されることを示す。hard オプションは、サーバーが応答するまで再試行要求が継続されることを示す。デフォルトは hard。

表 40-2 -o で指定する汎用マウントオプション (続き)

オプション	ファイルシステム	説明
intr nointr	NFS	ハードマウントされたファイルシステムに関する応答を待って実行を停止しているプロセスを、キーボード割り込みで強制終了できるかどうかを指定する。デフォルトは intr (割り込み可能)。
largefiles nolargefiles	UFS	2G バイトを超えるファイルを持つことができるようにする。largefiles オプションでマウントされたファイルシステムは、2G バイトを超えるファイルを格納できる (必須ではない)。nolargefiles オプションを指定した場合、UFS ファイルシステムは Solaris 2.6 (またはその互換バージョン) が動作しているシステムにはマウントできない。デフォルトは largefiles。
logging nologging	UFS	<p>ファイルシステムのロギングを有効または無効にする。UFS ロギングとは、トランザクション (完全な UFS 処理を構成する変更) をログに保存してから、そのトランザクションを UFS ファイルシステムに適用するプロセスである。ロギングを使用すると、UFS ファイルシステムの整合性を保つことができる。つまり、fsck を省略できることを意味する。fsck を省略すると、システムがクラッシュしたとき、あるいはシステムをクリーンな状態でシャットダウンできなかったとき、システムをリブートする時間を省ける。</p> <p>ログはファイルシステムの空きブロックから、1G バイトのファイルシステムごとに約 1M バイトのサイズ (合計で 64M バイトまで) が割り当てられる。デフォルトは nologging。</p>
atime noatime	UFS	ファイルのアクセス時間更新を抑制する。ただし、最後にファイルの状態が変わった時間または最後にファイルが変更された時間に対する更新が同時に行われる場合を除く。詳細については、stat(2) のマニュアルページを参照。このオプションによって、アクセス時間が重要でないファイルシステム (Usenet ニューススプールなど) でのディスクに対する動作が減る。デフォルトでは、通常アクセス時間 (atime) が記録される。

表 40-2 -o で指定する汎用マウントオプション (続き)

オプション	ファイルシステム	説明
remount	すべて	既にマウントされているファイルシステムに関連付けられているマウントオプションを変更する。通常このオプションは、ro オプション以外のオプションと共に使用できる。ただし、変更できるマウントオプションは、ファイルシステムタイプによって異なる。
retry= <i>n</i>	NFS	マウント処理に失敗した場合に再試行する。 <i>n</i> は再試行回数。
ro rw	CacheFS、NFS、PCFS、UFS、HSFS	読み取り / 書き込み (rw) または読み取り専用 (ro) を指定する。このオプションを指定しない場合のデフォルトは rw。HSFS のデフォルトのオプションは ro。
suid nosuid	CacheFS、HSFS、NFS、UFS	setuid 実行を許可または禁止する。デフォルトは、setuid 実行を許可する。

/etc/vfstab ファイルのフィールドの説明

/etc/vfstab ファイル内のエントリには、次の表に示すように7つのフィールドがあります。

表 40-3 /etc/vfstab ファイルのフィールドの説明

フィールド名	説明
device to mount	このフィールドは、次のいずれかを指定する。 <ul style="list-style-type: none"> ■ ローカル UFS ファイルシステム用のブロックデバイス名 (/dev/dsk/c0t0d0s0 など) ■ リモートファイルシステム用のリソース名 (myserver:/export/home など)。NFS の詳細については、『Solaris のシステム管理 (IP サービス)』を参照 ■ スワップ空間用のスライスのブロックデバイス名 (/dev/dsk/c0t3d0s1 など) ■ 仮想ファイルシステムタイプ用のディレクトリ
device to fsck	「device to mount」フィールドで指定した UFS ファイルシステムに対応する raw (キャラクタ型) デバイス名 (/dev/rdisk/c0t0d0s0 など)。このフィールドによって、fsck コマンドが使用する raw インタフェースが決まる。読み取り専用ファイルシステムやリモートファイルシステムなど、適用できるデバイスがない場合は、ダッシュ (-) を使用する。
mount point	デフォルトのマウントポイントディレクトリ (/usr など) を指定する。

表 40-3 /etc/vfstab ファイルのフィールドの説明 (続き)

フィールド名	説明
FS type	ファイルシステムのタイプを指定する。
fsck pass	<p>fsck コマンドがファイルシステムをチェックするか決めるために使用するパス番号。このフィールドでダッシュ (-) を指定すると、ファイルシステムはチェックされない。</p> <p>このフィールドに 0 が指定されている場合、UFS ファイルシステムはチェックされないが、UFS 以外のファイルシステムはチェックされる。このフィールドに 0 より大きい値が指定されている場合、UFS ファイルシステムは常にチェックされる。</p> <p>このフィールドに 1 が指定されている場合、すべてのファイルシステムは vfstab ファイル内の順番どおりに 1 つずつチェックされる。このフィールドに 1 より大きな値が指定され、さらに preen (修復) オプション (-o p) が指定されている UFS ファイルシステムが複数ある場合、効率を最大限に高めるために、fsck コマンドは複数のディスク上のファイルシステムを自動的に並行してチェックする。それ以外の場合、このフィールドの値は意味を持たない。</p>
mount at boot	システムのブート時にファイルシステムが mountall コマンドによって自動的にマウントされるかどうかを yes または no で設定する。このフィールドは AutoFS とは連動していないので注意すること。ルート (/)、/usr、/var のファイルシステムは最初は vfstab ファイルからマウントされない。これらのファイルシステムおよび /proc や /dev/fd などのような仮想ファイルシステムの場合、このフィールドは常に no に設定しなければならない。
mount options	ファイルシステムのマウントに使用されるオプションを (空白を入れずに) コンマで区切ったリスト。オプションなしを示すにはダッシュ (-) を使用する。汎用マウントオプションについては、表 40-2 を参照。

注 - /etc/vfstab ファイル内の各フィールドには必ずエントリが必要です。フィールドに値を指定しない場合は、必ずダッシュ (-) を入力してください。ダッシュを入力しないと、システムが正常にブートしない可能性があります。同様に、フィールドの値に空白文字を使用しないでください。

ファイルシステムのマウント

次の節では、/etc/vfstab ファイルにエントリを追加するか、コマンド行から mount コマンドを使用してファイルシステムをマウントする方法について説明します。

/etc/vfstab フィールドのエントリについては、表 40-3 を参照してください。

注 - ルート (/) ファイルシステムは、ブートプロセスの過程でカーネルによって読み取り専用としてマウントされます。そのため、remount オプション (および、remount と一緒に使用できるオプション) だけが /etc/vfstab ファイルのルート (/) エントリでは有効です。

例 — /etc/vfstab ファイルにエントリを追加する

次の例は、ディスクスライス /dev/dsk/c0t3d0s7 を UFS ファイルシステムとして、マウントポイントディレクトリ /files1 にマウントする方法を示しています。「device to fsck」として raw キャラクタ型デバイス /dev/rdisk/c0t3d0s7 を指定します。「fsck pass」の値が 2 なので、ファイルシステムは順不同でチェックされます。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
#
/dev/dsk/c0t3d0s7 /dev/rdisk/c0t3d0s7 /files1  ufs     2     yes    -
```

次の例は、システム pluto 上のディレクトリ /export/man を、NFS ファイルシステムとしてマウントポイント /usr/man にマウントする方法を示しています。ファイルシステムが NFS であるため、「device to fsck」や「fsck pass」は指定されません。この例では、「mount options」は ro (読み取り専用) と soft になっています。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
pluto:/export/man -                /usr/man nfs     -     yes    ro,soft
```

次の例は、ルート (/) ファイルシステムをループバックマウントポイント /tmp/newroot にマウントする方法を示しています。LOFS ファイルシステムをマウントするときは、LOFS ファイルシステム内に入るファイルシステムを先にマウントし、その後で LOFS をマウントします。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
#
/                -                /tmp/newroot lofs -     yes    -
```

▼ 1 つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. /etc/vfstab ファイル内に列挙されているファイルシステムをマウントします。

```
# mount /mount-point
```

`/mount-point` は、`/etc/vfstab` ファイル内の「mount point」または「device to mount」フィールドのエントリを指定します。通常は、マウントポイントを指定する方が簡単です。

例 — 1つのファイルシステムをマウントする (`/etc/vfstab` ファイル)

次の例は、`/etc/vfstab` ファイル内に列挙されているファイルシステム `/usr/dist` をマウントする方法を示しています。

```
# mount /usr/dist
```

例 — すべてのファイルシステムをマウントする (`/etc/vfstab` ファイル)

次の例は、`mountall` コマンドを実行したときに、すでにファイルシステムがマウントされている場合に表示されるメッセージを示します。

```
# mountall
/dev/rdsk/c0t0d0s7 already mounted
mount: /tmp already mounted
mount: /dev/dsk/c0t0d0s7 is already mounted, /export/home is busy,
       or the allowable number of mount points has been exceeded
```

マウントが実行される前に、「device to fsck」エントリがあるすべてのファイルシステムがチェックされ、必要であれば修正されます。

次の例は、`/etc/vfstab` ファイル内に列挙されているすべてのローカルシステムをマウントする方法を示しています。

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=38c0000 on Sun Feb  2 18:20:07 2003
/etc/mnttab on mnttab read/write/setuid/dev=3980000 on Sun Feb  2 ...
/dev/fd on fd read/write/setuid/dev=39c0000 on Sun Feb  2 18:20:10 2003
/var/run on swap read/write/setuid/xattr/dev=1 on Sun Feb  2 18:20:11 2003
/tmp on swap read/write/setuid/xattr/dev=2 on Sun Feb  2 18:20:15 2003
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr...
/datab on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/ ...
```

次の例は、`/etc/vfstab` ファイル内に列挙されているすべてのリモートファイルシステムをマウントする方法を示しています。

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
```

```

/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=38c0000 on Sun Feb  2 18:20:07 2003
/etc/mnttab on mnttab read/write/setuid/dev=3980000 on Sun Feb  2 ...
/dev/fd on fd read/write/setuid/dev=39c0000 on Sun Feb  2 18:20:10 2003
/var/run on swap read/write/setuid/xattr/dev=1 on Sun Feb  2 18:20:11 2003
/tmp on swap read/write/setuid/xattr/dev=2 on Sun Feb  2 18:20:15 2003
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr...
/datab on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/ ...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr ...

```

▼ UFS ファイルシステムのマウント方法 (mount コマンド)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. UFS ファイルシステムをマウントします。

```
# mount [-o mount-options] /dev/dsk/device-name /mount-point
```

<i>-o mount-options</i>	UFS ファイルシステムのマウントに使用できるマウントオプションを指定する。オプションの詳細については、表 40-2 または <code>mount_ufs(1M)</code> のマニュアルページを参照。
<i>/dev/dsk/device-name</i>	ファイルシステムが含まれているディスクスライス用のディスクデバイス名 (<code>/dev/dsk/c0t3d0s7</code> など) を指定する。ディスクのスライス情報については、481 ページの「ディスクスライス情報を表示する方法」を参照。
<i>/mount-point</i>	ファイルシステムをマウントするディレクトリを指定する。

例 — UFS ファイルシステムをマウントする (mount コマンド)

次の例は、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントする方法を示しています。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

例 — ログイングを有効にして UFS ファイルシステムをマウントする (mount コマンド)

UFS ログイングによってファイルシステムの整合性が保たれるために、システムのリブート時間が大幅に短縮されます。次の例は、ログイングを有効にして、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントする方法を示しています。

```
# mount -o logging /dev/dsk/c0t3d0s7 /files1
```

▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)

ファイルシステムをマウントするときには、`largefiles` オプションがデフォルトで選択されるため、2G バイトを超えるファイルを作成できます。大規模ファイルを作成した後で、`nolargefiles` オプションを指定してファイルシステムを再マウントするには、あるいは Solaris 2.6 およびその互換バージョンを実行するシステム上にマウントするには、大規模ファイルをすべて削除し、`fsck` コマンドを実行して状態を「`nolargefiles`」にリセットしなければなりません。

以下の手順では、ファイルシステム用のエントリが `/etc/vfstab` ファイルにあるものとします。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。
3. ファイルシステム内に大規模ファイルが存在しないことを確認します。

```
# cd /mount-point
# find . -xdev -size +20000000 -exec ls -l {} \;
```

`/mount-point` は、大規模ファイルがあるかどうかをチェックするファイルシステムのマウントポイントを指定します。
4. 大規模ファイルが当該ファイルシステム内に存在する場合は、必要に応じてそのファイルを削除するか、他のファイルシステムに移動します。
5. ファイルシステムをマウント解除します。

```
# umount /mount-point
```
6. ファイルシステムの状態をリセットします。

```
# fsck /mount-point
```

7. **nolargefiles** オプションを指定してファイルシステムを再マウントします。

```
# mount -o nolargefiles /mount-point
```

例 — 大規模ファイルを持たないファイルシステムをマウントする (mount コマンド)

次の例は、/datab ファイルシステムをチェックし、nolargefiles オプションを指定して再マウントする方法を示しています。

```
# cd /datab
# find . -xdev -size +2000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

▼ NFS ファイルシステムのマウント方法 (mount コマンド)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. リソース (ファイルまたはディレクトリ) がサーバーから使用可能かどうかを確認します。

NFS ファイルシステムをマウントするには、share コマンドを使用し、サーバー上のリソースを使用可能にしておかなければなりません。リソースの共有方法については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「NFS サービスについて」を参照してください。

4. NFS ファイルシステムをマウントします。

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

-o *mount-options*

NFS ファイルシステムのマウントに使用できるマウントオプションを指定する。汎用マウントオプションのリストについては、表 40-2 を参照。オプションについては、mount_nfs (1M) のマニュアルページを参照。

<i>server:/directory</i>	共有するリソースを持つサーバーのホスト名と、マウントするファイルまたはディレクトリへのパスを指定する。
<i>/mount-point</i>	ファイルシステムをマウントするディレクトリを指定する。

例 — NFS ファイルシステムをマウントする (mount コマンド)

次の例は、サーバー `pluto` の `/export/packages` ディレクトリを `/mnt` にマウントする方法を示しています。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)

次の手順で、PCFS (DOS) ファイルシステムをハードディスクからマウントします。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. PCFS ファイルシステムをマウントします。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
```

<code>-o rw ro</code>	PCFS ファイルシステムを読み取り / 書き込み (rw) または読み取り専用 (ro) にマウントできることを指定する。このオプションを指定しない場合のデフォルトは rw
<code>/dev/dsk/device-name</code>	ディスク全体のデバイス名を指定する (/dev/dsk/c0t0d0p0 など)
<code>logical-drive</code>	DOS の論理ドライブ名 (c から z) またはドライブ番号 (1 から 24) を指定する。ドライブ c はドライブ 1 に相当し、ディスク上の基本 DOS スライスを表す。他のすべてのドライブ名やドライブ番号は、拡張 DOS スライス内の DOS 論理ドライブを表す
<code>/mount-point</code>	ファイルシステムをマウントするディレクトリを指定する

「`device-name`」と「`logical-drive`」は、コロンで区切る必要があります。

x86: 例 — ハードディスクから PCFS (DOS) ファイルシステムをマウントする (mount コマンド)

次の例は、基本 DOS スライス内の論理ドライブを /pcfs/c ディレクトリにマウントする方法を示しています。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例は、拡張 DOS スライス内の最初の論理ドライブを /mnt に読み取り専用としてマウントする方法を示しています。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

ファイルシステムのマウント解除

ファイルシステムをマウント解除すると、ファイルシステムがそのマウントポイントから削除され、そのエントリが /etc/mnttab ファイルから削除されます。マウントされているファイルシステム上では、一部のファイルシステム管理作業を実行できません。次の場合には、ファイルシステムをマウント解除する必要があります。

- ファイルシステムが不要になった場合、または新しいソフトウェアが入ったファイルシステムに交換された場合
- fsck コマンドを使用してファイルシステムを検査し、修復する必要がある場合
fsck コマンドの詳細については、第 43 章を参照してください。

ファイルシステムの完全バックアップを実行する前に、マウント解除しておくといでしょう。バックアップの実行の詳細については、第 47 章を参照してください。

注 - 各ファイルシステムは、ファイルシステムのシャットダウン手順の一部として自動的にマウント解除されます。

umount -f オプションを使用すると、非常時に使用中のファイルシステムを強制的にマウント解除できます。ファイルを開いた状態でファイルシステムをマウント解除すると、データが失われる可能性があるため、非常時以外はこの操作を行わないようにしてください。このオプションは、UFS と NFS のファイルシステムでのみ使用できます。

ファイルシステムをマウント解除する場合の前提条件

ファイルシステムをマウント解除する場合の前提条件は次のとおりです。

- スーパーユーザーになるか、同等の役割を引き受けていなければならない。
- ファイルシステムは、マウント解除が可能な状態でなければならない。使用中のファイルシステムはマウント解除できません。ユーザーがそのディレクトリ内にいるとき、プログラムがそのファイルシステム上のファイルを開いているとき、または共有されているときには、ファイルシステムは使用中とみなされます。次の方法でファイルシステムをマウント解除が可能な状態にできます。
 - 別のファイルシステム内のディレクトリにカレントディレクトリを変更する。
 - システムからログアウトする。
 - `fuser` コマンドを使用して、そのファイルシステムを使用中のすべてのプロセスを表示し、必要に応じて終了させる。詳細については、593 ページの「ファイルシステムを使用中のすべてのプロセスを終了させる方法」を参照。

他のユーザーが使用しているファイルシステムをマウント解除する必要があるときは、各ユーザーに通知してください。
- ファイルシステムの共有を解除する。ファイルシステムの共有を解除する方法については、`unshare(1M)` のマニュアルページを参照してください。

ファイルシステムのマウント解除を確認する方法

ファイルシステムをマウント解除したことを確認するには、`mount` コマンドからの出力を調べます。

```
$ mount | grep unmounted-file-system
$
```

▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. どのプロセスを終了させるかがわかるように、ファイルシステムを使用中のすべてのプロセスを表示します。

```
# fuser -c [ -u ] /mount-point
```

<code>-c</code>	ファイルシステムのマウントポイントとなっているファイルと、マウントされているファイルシステム内のファイルが表示される。
<code>-u</code>	プロセス ID ごとにユーザーのログイン名が表示される。
<code>/mount-point</code>	プロセスを終了させるファイルシステムの名前を指定する。

3. ファイルシステムを使用しているすべてのプロセスを終了させます。

```
# fuser -c -k /mount-point
```

ファイルシステムを使用している各プロセスに SIGKILL が送信されます。

注 - ユーザーのプロセスを終了させるときには、必ず事前に警告してください。

4. ファイルシステムを使用しているプロセスがないことを確認します。

```
# fuser -c /mount-point
```

例 — あるファイルシステムを使用中のすべてのプロセスを終了させる

次の例は、`/export/home` ファイルシステムを使用中のプロセス `4006c` を終了させる方法を示しています。

```
# fuser -c /export/home
/export/home:      4006c
# fuser -c -k /export/home
/export/home:      4006c
# fuser -c /export/home
/export/home:
```

▼ 1つのファイルシステムをマウント解除する方法

次の手順に従って、ファイルシステム (ルート (`/`)、`/usr`、`/var` を除く) をマウント解除します。

注 - ルート(`/`)、`/usr`、`/var` の各ファイルシステムは、システムが機能するのに必要であるため、シャットダウン中でなければマウント解除できません。

1. 593 ページの「ファイルシステムをマウント解除する場合の前提条件」の前提条件を満たしているかどうかを確認します。

2. ファイルシステムをマウント解除します。

```
# umount /mount-point
```

`/mount-point` は、マウント解除するファイルシステムの名前を示します。次のいずれかを指定できます。

- ファイルシステムがマウントされているディレクトリ名
- ファイルシステムのデバイス名パス
- NFS ファイルシステムのリソース
- LOFS ファイルシステムのループバックディレクトリ

例 — 1つのファイルシステムをマウント解除する

次の例は、ローカルのホームディレクトリからファイルシステムをマウント解除する方法を示しています。

```
# umount /export/home
```

次の例は、ローカルディスクの7番目のスライス上のファイルシステムをマウント解除する方法を示しています。

```
# umount /dev/dsk/c0t0d0s7
```

次の例は、`/export` ファイルシステムを強制的にマウント解除する方法を示しています。

```
# umount -f /export  
#
```

次の例は、`/etc/vfstab` ファイル内のすべてのファイルシステム (ルート (/)、`/proc`、`/var`、`/usr` を除く) をマウント解除する方法を示しています。

```
# umountall
```

使用中のファイルシステムを除く、すべてのファイルシステムがマウント解除されます。

第 41 章

CacheFS ファイルシステムの使用 (手順)

この章では、CacheFS™ ファイルシステムの設定および管理の方法について説明します。

この章で説明する作業マップは、次のとおりです。

- 597 ページの「CacheFS ファイルシステムの概観 (作業マップ)」
- 600 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」
- 606 ページの「CacheFS ファイルシステムの保守 (作業マップ)」
- 612 ページの「CacheFS ファイルシステムのバックアップ (作業マップ)」
- 621 ページの「CacheFS の統計情報の収集 (作業マップ)」

CacheFS に関するエラーの障害追跡については、618 ページの「cachefspack エラーの障害追跡」を参照してください。

CacheFS ファイルシステムの概観 (作業マップ)

次の作業マップは、CacheFS ファイルシステムを使用するためのすべての作業を確認するために使用してください。このマップに記載された各作業は、CacheFS ファイルシステムの作成とマウント、キャッシュのバックアップと保守などの一連の付加された作業を指しています。

作業	説明	参照先
1. CacheFS ファイルシステムを作成してマウントする	キャッシュを作成し、ファイルシステムをそのキャッシュにマウントする。	600 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」

作業	説明	参照先
2. CacheFS ファイルシステムを保守する	キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムを表示および変更する。	606 ページの「CacheFS ファイルシステムの保守 (作業マップ)」
3. (省略可能) CacheFS ファイルシステムをバックおよびバック解除する	キャッシュをバックしてバックングリストを使用するかどうかを決める。 キャッシュをバックすると、キャッシュ内の特定のファイルおよびディレクトリが常に更新されるようになる。	612 ページの「CacheFS ファイルシステムのバックング (作業マップ)」
4. CacheFS の統計情報を収集する	キャッシュの性能や適切なキャッシュサイズを決める。	621 ページの「CacheFS の統計情報の収集 (作業マップ)」

CacheFS ファイルシステムの概要

CacheFS ファイルシステムは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティを改善する汎用ファイルシステムキャッシングメカニズムです。CacheFS ファイルシステムは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ファイルシステムはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。

CacheFS ファイルシステムの機能

CacheFS ファイルシステムをクライアントシステム上に作成すると、キャッシュに書き込んだファイルシステムをクライアントがネットワークを経由しなくても、ローカルにアクセスできます。次の図は、CacheFS ファイルシステムの使用に関連する構成要素の関係を示しています。

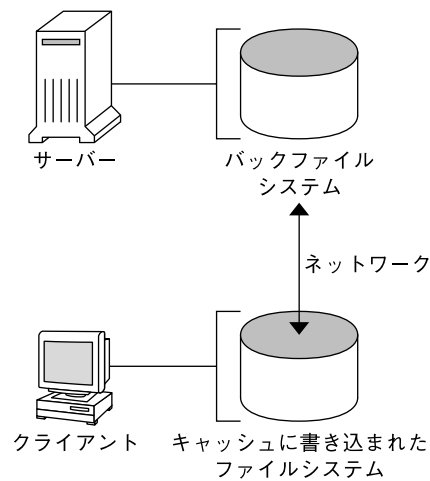


図 41-1 CacheFS ファイルシステムの機能

「バック」ファイルシステムとは、キャッシュにマウントされるように指定したファイルシステムです。通常、これは NFS または HSFS (High Sierra File System) ファイルシステムです。ユーザーがバックファイルシステムの一部であるファイルにアクセスしようとする、そのファイルはキャッシュに書き込まれます。「フロント」ファイルシステムとは、キャッシュにマウントされ、ローカルのマウントポイントからアクセスされるファイルシステムです。フロントファイルシステムのタイプは、UFS でなければなりません。

ユーザーにとって、CacheFS ファイルシステムのファイルに初めてアクセスするときは低速に思われますが、同じファイルを続けて使用していると高速になります。

CacheFS ファイルシステムの構造と動作

各キャッシュには、キャッシュの構造とその動作を決定する 1 組のパラメータが付いています。各パラメータは、以下の表に示すデフォルト値に設定されています。デフォルト値は、フロントファイルシステム全体をキャッシュに使用するように指定しますが、これはファイルシステムをキャッシュに書き込む場合の推奨方法です。

表 41-1 CacheFS ファイルシステムのパラメータとそのデフォルト値

CacheFS ファイルシステムのパラメータ	デフォルト値	定義
maxblocks	90%	CacheFS ファイルシステムがフロントファイルシステム内で要求できる最大ブロック数を設定する

表 41-1 CacheFS ファイルシステムのパラメータとそのデフォルト値 (続き)

CacheFS ファイルシステムのパラメータ	デフォルト値	定義
minblocks	0%	CacheFS ファイルシステムがフロントファイルシステム内で要求できる最小ブロック数を設定する
threshblocks	85%	CacheFS ファイルシステムが minblocks で指定したより多数のブロックを要求する前に、フロントファイルシステム内で使用可能でなければならないブロック数を設定する
maxfiles	90%	CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最大数 (ファイル数) を設定する
minfiles	0%	CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最小数を設定する
threshfiles	85%	CacheFS ファイルシステムが minfiles で指定したより多くのファイルを要求する前に、フロントファイルシステム内で使用可能でなければならない i ノード数を設定する

通常、これらのパラメータ値を変更する必要はありません。最適のキャッシュ動作が得られるデフォルト値に設定されています。ただし、キャッシュに使用されないフロントファイルシステム内に空き空間があり、それを他のファイルシステムに使用する場合は、maxblocks と maxfiles の値を変更することをお勧めします。この変更を行うには、cfsadmin コマンドを使用します。たとえば、次のようになります。

```
$ cfsadmin -o maxblocks=60
```

CacheFS ファイルシステムの作成とマウント (作業マップ)

次の表の手順を使用して、CacheFS ファイルシステムを作成およびマウントしてください。

作業	説明	参照先
1. キャッシュに書き込むファイルシステムを共有する	キャッシュに書き込むファイルシステムが共有されていることを確認する。	share (1M) のマニュアルページ

作業	説明	参照先
2. キャッシュを作成する	<code>cfsadmin</code> コマンドを使用してキャッシュを作成する。	601 ページの「キャッシュを作成する方法」
3. ファイルシステムをキャッシュにマウントする	次のいずれかの方法を使用して、ファイルシステムをキャッシュにマウントする。 <code>mount</code> コマンドで、CacheFS ファイルシステムをマウントする <code>/etc/vfstab</code> ファイルを編集して、CacheFS ファイルシステムをマウントする AutoFS を使用して、キャッシュされたファイルシステムをマウントする	602 ページの「CacheFS ファイルシステムのマウント方法 (<code>mount</code>)」 604 ページの「CacheFS ファイルシステムのマウント方法 (<code>/etc/vfstab</code>)」 605 ページの「CacheFS ファイルシステムのマウント方法 (AutoFS)」

▼ キャッシュを作成する方法

1. クライアントシステムでスーパーユーザーになります。
2. キャッシュを作成します。

```
# cfsadmin -c /cache-directory
```

`cache-directory` は、キャッシュがあるディレクトリの名前を示します。

詳細については、`cfsadmin(1M)` のマニュアルページを参照してください。

注 - キャッシュを作成し終わったら、キャッシュディレクトリ内で処理を実行しないでください。何か実行すると、CacheFS ソフトウェア内で矛盾が生じる可能性があります。

例 — キャッシュを作成する

次の例は、デフォルトのキャッシュパラメータ値を使用して、`/local/mycache` ディレクトリ内にキャッシュを作成する方法を示しています。

```
# mkdir /local
# cfsadmin -c /local/mycache
```

ファイルシステムをキャッシュにマウントする

キャッシュにマウントされるファイルシステムを指定して、ユーザーがそのファイルシステム内のファイルにローカルにアクセスできるようにします。実際には、各ファイルは、ユーザーがアクセスするまではキャッシュに書き込まれません。

次の表に、CacheFS ファイルシステムをマウントする 3 つの方法を示します。

CacheFS ファイルシステムのマウント方法	CacheFS マウント方法の使用頻度
mount コマンドを使用する方法	同じファイルにアクセスするには、システムをリブートするたびに mount コマンドを使用する
/etc/vfstab ファイルを編集する方法	1 度だけですむ。/etc/vfstab ファイルの内容は、システムをリブートした後も変更されずに残る
AutoFS を使用する方法	1 度だけですむ。AutoFS のマップは、システムをリブートした後も変更されずに残る

いずれかの方法を選択してファイルシステムをマウントしてください。

共有可能なファイルシステムしかマウントできません。ファイルシステムの共有については、share (1M) のマニュアルページを参照してください。

注 - CacheFS ファイルシステムでは、ルート (/) と /usr のファイルシステムをキャッシュに書き込むことはできません。

▼ CacheFS ファイルシステムのマウント方法 (mount)

1. クライアントシステムでスーパーユーザーになります。
2. 必要に応じて、マウントポイントを作成します。

```
# mkdir /mount-point
```

マウントポイントはどこからでも作成できますが、UFS ファイルシステムでなければなりません。次の手順のように、mount コマンドで CacheFS オプションを使用すると、作成するマウントポイントが指定したキャッシュディレクトリ内のキャッシュに書き込まれるように指定できます。

3. ファイルシステムをキャッシュにマウントします。

```
# mount -F cachefs -o backfstype=fstype,cachedir=/cache-directory [,options] /back-filesystem /mount-point
```

<i>fstype</i>	バックファイルシステムのファイルシステムタイプ (NFS または HSFS)
<i>/cache-directory</i>	キャッシュがある UFS ディレクトリの名前。これは、601 ページの「キャッシュを作成する方法」でキャッシュを作成するときの指定と同じ
<i>options</i>	ファイルシステムをキャッシュにマウントするときに追加できる他のマウントオプション。CacheFS mount オプションの詳細については、mount_cachefs(1M) のマニュアルページを参照
<i>/back-filesystem</i>	キャッシュにマウントするバックファイルシステムのマウントポイント。バックファイルシステムが NFS ファイルシステムである場合は、ファイルシステムのマウント元となるサーバーのホスト名と、キャッシュにマウントするファイルシステム名 (コロンで区切る) を指定する必要がある。たとえば、 <i>merlin:/data/abc</i>
<i>/mount-point</i>	ファイルシステムのマウント先となるディレクトリ

4. 作成したキャッシュが実際にマウントされたかどうかを確認します。

```
# cachefsstat /mount-point
```

/mount-point は作成した CacheFS ファイルシステムです。

たとえば、次のようになります。

```
# cachefsstat /docs
```

```
/docs
```

```
      cache hit rate:   100% (0 hits, 0 misses)
consistency checks:   1 (1 pass, 0 fail)
      modifies:        0
garbage collection:   0
```

ファイルシステムがキャッシュにマウントされなかった場合は、次のようなエラーメッセージが表示されます。

```
# cachefsstat /mount-point
```

```
cachefsstat: mount-point: not a cachefs mountpoint
```

cachefsstat コマンドの詳細については、622 ページの「CacheFS の統計情報の収集」を参照してください。

例 — CacheFS ファイルシステムをマウントする (mount)

次の例は、NFS ファイルシステム *merlin:/docs* を */docs* という CacheFS ファイルシステムとして、*/local/mycache* というキャッシュにマウントする方法を示しています。

```
# mkdir /docs
```

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

次の例は、Solaris 9 SPARC CD (HSFS ファイルシステム) を /cfssrc という CacheFS ファイルシステムとして使用できるようにする方法を示しています。CDには書き込めないで、引数 ro を指定して CacheFS ファイルシステムを読み取り専用にします。この例では、vold を実行していないものとします。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /sol9
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/sol9 /dev/dsk/c0t6d0s0 /cfssrc
# ls /cfssrc
Copyright Solaris_9
```

次の例は、vold を実行しながら、Solaris 9 SPARC CD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/sol_9_sparc/s0 /vol/dev/dsk/c0t2d0/sol_9_sparc/s0 /cfssrc
```

次の例は、vold を実行しながら、CD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/epson /vol/dev/dsk/c0t2d0/epson /drvrs
```

次の例では、demandconst オプションを使用して、NFS CacheFS ファイルシステム /docs の整合性チェックを指定します。/docs のバックファイルシステムは merlin:/docs です。詳細については、608 ページの「CacheFS ファイルシステムの整合性チェック」を参照してください。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ CacheFS ファイルシステムのマウント方法 (/etc/vfstab)

1. クライアントシステムでスーパーユーザーになります。
2. エディタを使用して、マウントするファイルシステムを /etc/vfstab ファイル内で指定します。
下の例を参照してください。
/etc/vfstab ファイルの詳細については、583 ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。
3. CacheFS ファイルシステムをマウントします。

```
# mount /mount-point
```

あるいはシステムをリブートします。

例 — CacheFS ファイルシステムをマウントする (/etc/vfstab)

次の例は、キャッシュされたディレクトリ /opt/cache にマウントされるリモートシステム starbug からの /data/abc ディレクトリ用の /etc/vfstab エントリを示しています。

```
#device          device          mount      FS      fsck  mount  mount
#to mount        to fsck         point      type    pass  at boot options
#
starbug:/data/abc /local/abc     /opt/cache cachefs 7    yes   local-access,bg,
nosuid,demandconst,backfstype=nfs,cachedir=/opt/cache
```

▼ CacheFS ファイルシステムのマウント方法 (AutoFS)

自動マウントマップ内で `-fstype=cachefs` マウントオプションを指定して、AutoFS によってファイルシステムをキャッシュにマウントします。CacheFS のマウントオプション (`backfstype` や `cachedir` など) も、自動マウントマップ内で指定します。自動マウントマップの詳細については、`automount (1M)` のマニュアルページを参照してください。

1. クライアントシステムでスーパーユーザーになります。

2. エディタを使用して、次の 1 行を `auto_direct` マップに追加します。

```
/mount-point -fstype=cachefs,cachedir=/directory,backfstype=nfs
server:/file-system
```

3. エディタを使用して、次の 1 行を `auto_master` マップに追加します。

```
/-
/- エントリによって、auto_direct マップをチェックするように指示します。
```

4. システムをリブートします。

5. エントリが正しく作成されたか確認するには、次のようにキャッシュにマウントしたファイルシステムにカレントディレクトリを変更して内容を表示します。

```
# cd /filesystem
# ls
```

AutoFS とマップの編集方法については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「autofs 管理作業の概要」を参照してください。

例 — CacheFS ファイルシステムをマウントする (AutoFS)

次の `auto_direct` エントリは、CacheFS ファイルシステムを `/docs` ディレクトリに自動的にマウントします。

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

CacheFS ファイルシステムの保守 (作業マップ)

CacheFS ファイルシステムを設定し終わったら、保守作業を多少行う必要があります。CacheFS ファイルシステムの保守作業を行う必要がある場合は、次の表の手順 (オプション) に従ってください。

作業	説明	参照先
1. CacheFS ファイルシステムの変更	キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムの動作を変更する。	607 ページの「CacheFS ファイルシステムの変更」
2. CacheFS ファイルシステム情報の表示	<code>cfsadmin</code> コマンドを使用して、CacheFS ファイルシステムに関する情報を表示する。	608 ページの「CacheFS ファイルシステムに関する情報を表示する方法」
3. 整合性チェックの実行	<code>cfsadmin</code> コマンドを使用して必要に応じて整合性チェックを実行する。	609 ページの「必要に応じてキャッシュの整合性チェックを指定する方法」
4. CacheFS ファイルシステムの削除	<code>umount</code> コマンドと <code>cfsadmin</code> コマンドを使用して、CacheFS ファイルシステムを削除する。	609 ページの「CacheFS ファイルシステムを削除する方法」
5. CacheFS ファイルシステムの整合性の検査	<code>fsck_cachefs</code> コマンドを使用して、CacheFS ファイルシステムの整合性を検査する。	611 ページの「CacheFS ファイルシステムの整合性をチェックする方法」

CacheFS ファイルシステムの保守

この節では、CacheFS ファイルシステムの保守方法について説明します。

/etc/vfstab ファイルを使用してファイルシステムをマウントしている場合は、このファイル内でファイルシステムオプションを編集してキャッシュを変更します。AutoFS を使用している場合は、AutoFS マップ内でファイルシステムオプションを編集してキャッシュを変更します。

CacheFS ファイルシステムの変更

キャッシュ内でファイルシステムを変更する場合は、キャッシュを削除してから作成し直す必要があります。また、ファイルシステムの共有方法とアクセス方法によっては、マシンをシングルユーザーモードでリブートしなければならない場合があります。

次の例では、キャッシュが削除されてから再び作成され、ファイルシステム /docs に指定された demandconst オプションを使用して再びマウントされます。

```
# shutdown -g30 -y
.
.
.
Type Cntrl-d to proceed with normal startup,
(or give root password for system maintenance):
# enter password:
.
.
.
(ここで、キャッシュがあるファイルシステムで
fsck コマンドを実行するようメッセージが表示される。)

# fsck /local
# mount /local
# cfsadmin -d all /local/mycache
# cfsadmin -c /local/mycache
# init 6
.
.
.
console login:
password:
# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1,demandconst
merlin:/docs /docs
#
```

▼ CacheFS ファイルシステムに関する情報を表示する方法

1. クライアントシステムでスーパーユーザーになります。
2. 指定したキャッシュにマウントされたすべてのファイルシステムに関する情報を表示します。

```
# cfsadmin -l /cache-directory
/cache-directory は、キャッシュがあるディレクトリの名前です。
```

例 — CacheFS ファイルシステムに関する情報を表示する

次の例は、/local/mycache キャッシュディレクトリに関する情報を示しています。この例では、/docs ファイルシステムが /local/mycache にキャッシュされません。最終行には、CacheFS ファイルシステムの名前が表示されます。

```
# cfsadmin -l /local/mycache
cfsadmin: list cache FS information
  maxblocks      90%
  minblocks      0%
  threshblocks   85%
  maxfiles       90%
  minfiles       0%
  threshfiles    85%
  maxfilesize    3MB
merlin:_docs:_docs
#
```

CacheFS ファイルシステムの整合性チェック

CacheFS ソフトウェアでは、キャッシュされたディレクトリとファイルが最新の状態に保たれることを保証するために、キャッシュに格納されているファイルの整合性を定期的にチェックします。整合性をチェックするために、CacheFS ソフトウェアでは現在の変更時刻を前回の変更時刻と比較します。変更時刻が異なる場合は、そのディレクトリまたはファイルに関するすべてのデータと属性がキャッシュから消去されます。そして、バックファイルシステムから新しいデータと属性が取り出されます。

必要に応じて行う整合性チェック

整合性チェックは、`-o demandconst` オプションを使用してマウントされたファイルシステムに対して明示的に要求した場合のみ実行できます。このオプションでファイルシステムをキャッシュにマウントした場合は、`-s` オプションを指定した `cfsadmin` コマンドを実行して整合性チェックを要求します。デフォルトでは、

ファイルがアクセスされるたびに、各ファイルの整合性がチェックされます。ファイルがアクセスされなければ、チェックは実行されません。-o demandconst オプションを使用すると、整合性チェックによってネットワークがいっぱいになるという事態を回避できます。

詳細については、mount_cachefs(1M) のマニュアルページを参照してください。

▼ 必要に応じてキャッシュの整合性チェックを指定する方法

1. クライアントシステムでスーパーユーザーになります。
2. ファイルシステムをキャッシュにマウントし、キャッシュの整合性チェックを指定します。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/directory,demandconst
server:/file-system /mount-point
```

3. 特定の **CacheFS** ファイルシステムに対する整合性チェックを開始します。

```
# cfsadmin -s /mount-point
```

▼ CacheFS ファイルシステムを削除する方法

1. クライアントシステムでスーパーユーザーになります。
2. **CacheFS** ファイルシステムをマウント解除します。

```
# umount /mount-point
```

/mount-point は、削除する CacheFS ファイルシステムを示します。

3. **CacheFS** ファイルシステムの名前 (キャッシュ ID) を判別します。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
maxblocks      90%
minblocks       0%
threshblocks   85%
maxfiles        90%
minfiles         0%
threshfiles     85%
maxfilesize     3MB
cache-ID
#
```

4. 指定したキャッシュから **CacheFS** ファイルシステムを削除します。

```
# cfsadmin -d cache-ID /cache-directory
```

<i>cache-ID</i>	CacheFS ファイルシステム名。この名前は、 <code>cfsadmin -l</code> コマンドからの出力の最終行に表示される。詳細については、608 ページの「CacheFS ファイルシステムに関する情報を表示する方法」を参照。 <i>cache-ID</i> に <code>all</code> を指定すると、特定のキャッシュに書き込まれた CacheFS ファイルシステムをすべて削除できる。
<i>/cache-directory</i>	キャッシュがあるディレクトリ。

5. ファイルシステムが削除されたことを確認します。

直前に削除したファイルシステムのキャッシュ ID が、`cfsadmin -l` の出力に表示されなくなります。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
  maxblocks      90%
  minblocks      0%
  threshblocks   85%
  maxfiles       90%
  minfiles       0%
  threshfiles    85%
  maxfilesize    3MB
#
```

このコマンド出力に指定されるフィールドについては、`cfsadmin(1M)` のマニュアルページを参照してください。

6. `fsck -F cachefs` コマンドを実行して、キャッシュのリソースアカウントを更新します。

詳細については、611 ページの「CacheFS ファイルシステムの整合性をチェックする方法」を参照してください。

例 — CacheFS ファイルシステムを削除する

次の例は、キャッシュからファイルシステムを削除する方法を示しています。

```
# umount /cfssrc
# cfsadmin -l /cfssrc
# cfsadmin -d _dev_dsk_c0t6d0s0:_cfssrc
# cfsadmin -l
```

▼ CacheFS ファイルシステムの整合性をチェックする方法

`fsck` コマンドを使用して、CacheFS ファイルシステムの整合性を確認します。何も操作しなくても、`fsck` コマンドの CacheFS バージョンによって問題が自動的に解決されます。`fsck` コマンドはブート時またはファイルシステムのマウント時に自動的に実行されるため、CacheFS ファイルシステムに対して `fsck` コマンドを手作業で実行する必要はありません。整合性を手作業で確認する場合は、次の手順を使用できます。

詳細については、`fsck_cachefs(1M)` のマニュアルページを参照してください。

1. クライアントシステムでスーパーユーザーになります。
2. 指定されたキャッシュ内でファイルシステムをチェックします。

```
# fsck -F cachefs [-m -o noclean] /cache-directory
```

<code>-m</code>	<code>fsck</code> コマンドに CacheFS ファイルシステムをチェックさせるが、修復は行わない。
-----------------	--

<code>-o noclean</code>	CacheFS ファイルシステムに対するチェックのみを実行させる。修復は行わない。
-------------------------	---

<code>/cache-directory</code>	キャッシュがあるディレクトリの名前を指定する。
-------------------------------	-------------------------

例 — CacheFS ファイルシステムの整合性をチェックする

次の例は、`/local/mycache` キャッシュに書き込まれているファイルシステムをチェックする方法を示しています。

```
# fsck -F cachefs /local/mycache  
#
```

CacheFS ファイルシステムのバックアップ (作業マップ)

次の作業マップでは、CacheFS ファイルシステムのバックアップに関連する手順について説明します。これらの手順はすべてオプションです。

作業	説明	参照先
ファイルをキャッシュにバックアップする	キャッシュにロードされるファイルとディレクトリを識別し、それらをバックアップする。バックアップは、これらのファイルのコピーがキャッシュで利用できることを保証する。	613 ページの「キャッシュにファイルをバックアップする方法」
バックアップリストを作成する	キャッシュにバックアップするファイルを個々に指定しない場合は、バックアップリストを作成する。	615 ページの「バックアップリストを作成する方法」
バックアップリストを使ってファイルをキャッシュにバックアップする	キャッシュにバックアップするファイルが記載されているバックアップリストの名前を指定する。	616 ページの「バックアップリストを使ってファイルをキャッシュにバックアップする方法」
キャッシュからファイルまたはバックアップリストのバックアップを解除する	不要になったファイルをキャッシュから削除する。	617 ページの「キャッシュからファイルまたはバックアップリストのバックアップを解除する方法」
バックアップされたファイルの情報を表示する	バックアップされたファイルに関する情報 (バックアップ状況など) を表示する。	614 ページの「バックアップされたファイルの情報を表示する方法」

CacheFS ファイルシステムのバックアップ

一般的な使い方では、設定が終わると、CacheFS ソフトウェアは、ユーザーが要求しなくても自動的に適切な動作をします。最近使用されたファイルがキャッシュされます。「バックアップ」機能を使用するとキャッシュ内で、特定のファイルまたはディレクトリを常に最新の状態で保持できるので、より積極的にキャッシュを管理できます。

cachefspack コマンドを使用すると、キャッシュにロードするファイルとディレク
トリーを指定できます。このコマンドは、これらのファイルのコピーがキャッシュで利
用できることを保証します。

パッキングリストには、特定のファイル名やディレクトリー名が入っています。他の
パッキングリストを入れることも可能です。この機能によって、たくさんの項目を
キャッシュにパックする場合に、個々のファイルやディレクトリーを指定する手間が省
けます。

次のように、`-h` オプションを使用すると、cachefspack のすべてのオプションの簡
単なヘルプ情報を出力できます。

```
$ cachefspack -h
Must select 1 and only 1 of the following 5 options
-d Display selected filenames
-i Display selected filenames packing status
-p Pack selected filenames
-u Unpack selected filenames
-U Unpack all files in directory 'dir'
-f Specify input file containing rules
-h Print usage information
-r Interpret strings in LIST rules as regular expressions
-s Strip './' from the beginning of a pattern name
-v Verbose option
files - a list of filenames to be packed/unpacked
```

キャッシュにファイルをパックする方法

ファイルをキャッシュにパックするには、cachefspack コマンドを使用します。

```
$ cachefspack -p filename
```

<code>-p</code>	ファイルまたは複数のファイルをパックすることを示す。こ のオプションはデフォルト。
<code>filename</code>	キャッシュにパックするファイルまたはディレクトリーの名前 を指定する。ディレクトリーを指定すると、そのサブディレク トリーもすべてパックされる。詳細については、cachefspack (1M) のマニュアルページを参照。

例 — ファイルをキャッシュにパックする

次の例は、`projects` ファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects
```

次の例は、3つのファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects updates master_plan
```

次の例は、ディレクトリがキャッシュにパックされることを示しています。

```
$ cachefspack -p /data/abc/bin
```

パックされたファイルの情報を表示する方法

パックされたファイル情報を表示するには、`cachefspack -i` コマンドを使用します。

```
$ cachefspack -i[v] filename
```

<code>-i</code>	パックされたファイルの情報を表示することを示す。
<code>-v</code>	詳細表示オプション。
<code>cached-filename-or-directory</code>	情報を表示するファイル名またはディレクトリ名を指定する。

例 — パックされたファイルの情報を表示する

次の例は、`doc_file` ファイルが正常にパックされていることを示しています。

```
$ cachefspack -i doc_file
cachefspack: file doc_file marked packed YES, packed YES
```

次の例では、`/data/abc` ディレクトリに `bin` サブディレクトリがあります。`bin` サブディレクトリには、`big`、`medium`、および `small` という3つのファイルがあります。`big` と `small` ファイルはパックするように指定されていますが、パックされていません。`medium` ファイルは正常にパックされています。

```
$ cd /data/abc
$ cachefspack -i bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO
cachefspack: file /bin/medium marked packed YES,
packed YES
cachefspack: file /bin/small marked packed YES,
packed NO
.
.
.
```

`-iv` オプションを指定した場合は、指定したファイルまたはディレクトリがキャッシュからフラッシュされたかどうかに関する追加情報が表示されます。たとえば、次のようになります。

```
$ cd /data/bin
$ cachefspack -iv bin
.
```

```
.
.
cachefspack: file /bin/big marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/medium marked packed YES,
packed YES, nocache NO
cachefspack: file /bin/small marked packed YES,
packed NO
nocache NO
.
.
.
```

上記例の最後の行は、ディレクトリの内容がキャッシュからフラッシュされていないことを示しています。

パッキングリストの使用

`cachefspack` コマンドの機能の1つに、パッキングリストを作成するという機能があります。

パッキングリストには、キャッシュにパックするファイルやディレクトリが入っています。ディレクトリがパッキングリストに入っている場合、そのサブディレクトリとファイルもすべてパックされます。

この機能によって、キャッシュにパックする各ファイルをそれぞれ指定するという手間が省けます。

パッキングリストを作成する方法

パッキングリストを作成するには、`vi` などのエディタを使用して、ファイルを開きます。パッキングリストファイルの書式は、`filesync` コマンドで使用する書式と同じです。詳細については、`filesync(1)` のマニュアルページを参照してください。

パッキングリストには、次の2つの機能があります。

- パッキングリスト内のファイルを文字通りのファイル名ではなく、正規表現として識別することができるため、ファイル名を個別に指定する必要がありません。
- 所有するファイルだけがパックされるようにすることで、共有ディレクトリのファイルをパックできます。

これらの機能の使い方については、`cachefspack(1M)` のマニュアルページを参照してください。

例 — パッキングリストを作成する

次の例は、パッキングリストファイルの内容を示しています。

```
BASE /home/ignatz
LIST plans
LIST docs
IGNORE *.ps
```

- BASE 文は、パックする項目が存在するディレクトリのパスを指定しています。
- 2つの LIST 文は、当刻ディレクトリ中のパックされるファイルを指定しています。
- IGNORE 文は、パックしないファイルタイプ (ここでは .ps) を指定しています。

パッキングリストを使ってファイルを キャッシュにパックする方法

パッキングリストのファイルをパックするには、次のように `cachefspack -f` コマンドを使用します。

```
$ cachefspack -f packing-list
```

<code>-f</code>	パッキングリストを使用することを示す。
<code>packing-list</code>	パッキングリスト名を指定する。

例 — パッキングリストを使ってファイルをキャッシュに パックする

この例では、`list.pkg` ファイルを `cachefspack` コマンドのパッキングリストとして使用しています。

```
$ cachefspack -f list.pkg
```

キャッシュからファイルまたはパッキングリスト のパッキングを解除する

キャッシュからファイルを削除、つまりそのパッキングを解除する場合があります。他よりも優先度が高い一部のファイルまたはディレクトリが存在する場合があるため、重要でないファイルのパッキングを解除する必要があります。たとえば、あるプロジェクトを終了して、そのプロジェクトに関連するファイルをアーカイブしたと仮定します。次は、新しいプロジェクト、つまり新しいファイルのセットで作業することになります。

キャッシュからファイルまたはパッキングリストのパッキングを解除する方法

キャッシュからファイルまたはパッキングリストのパッキングを解除するには、`cachefspack` コマンドの `-u` または `-U` オプションを使用します。

```
$ cachefspack -u filename | -U cache-directory
```

<code>-u</code>	ファイルまたは複数のファイルのパッキングを解除することを示す。このオプションを使用する場合は、ファイル名を指定しなければならない。
<code>filename</code>	キャッシュからパッキングを解除するファイルまたはパッキングリストの名前を示す。
<code>-U</code>	キャッシュ中のすべてのファイルのパッキングを解除することを示す。

`cachefspack` コマンドの詳細については、マニュアルページを参照してください。

例 — キャッシュからファイルまたはパッキングリストのパッキングを解除する

次の例は、キャッシュから `/data/abc/bin/big` ファイルのパッキングが解除されることを示しています。

```
$ cachefspack -u /data/abc/bin/big
```

次の例は、キャッシュからいくつかのファイルのパッキングが解除されることを示しています。

```
$ cd /data/abc/bin/big  
$ cachefspack -u big small medium
```

次の例は、パッキングリスト (ファイルのディレクトリへのパスを含むファイル) のパッキングを解除する方法を示しています。

```
$ cachefspack -uf list.pkg
```

次の例は、キャッシュディレクトリ中のすべてのファイルのパッキングが解除されるように `-U` オプションを使用する方法を示しています。

```
$ cachefspack -U /local/mycache
```

ファイルシステムを1つもマウントしていないキャッシュのパッキングは解除できません。`-U` オプションに対して、マウント済みファイルシステムを持っていないキャッシュを指定した場合、次のような出力が表示されます。

```
$ cachefspack -U /local/mycache  
cachefspack: Could not unpack cache /local/mycache, no mounted
```

filesystems in the cache.

cachefspack エラーの障害追跡

cachefspack コマンドを使用すると、次のようなエラーメッセージが表示されることがあります。

cachefspack: *pathname*

- ディレクトリをオープンできません。アクセス権がありません。

原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

解決法

適切なアクセス権を取得してください。

cachefspack: *pathname*

- ディレクトリをオープンできません。

ファイルまたはディレクトリがありません。

原因

正しいファイルまたはディレクトリがありません。

解決法

入力ミスがないか確認してください。

cachefspack: *pathname* - ディレクトリをオープンできません。

NFS のファイルハンドルが無効です。

原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

cachefspack: *pathname* - ディレクトリをオープンできません。

システムコールに割り込みがかかりました。

原因

コマンドの実行中に間違って Control + C を押した可能性があります。

解決法

このコマンドを再度実行してください。

cachefspack: *pathname* - ディレクトリをオープンできません。I/O エラー

原因

ハードウェアの障害の可能性があります。

解決法

ハードウェアの接続を確認してください。

cachefspack: ディレクトリのオープンに失敗しました。

原因

正しいファイルまたはディレクトリがありません。ファイル書式中の BASE コマンドの後に指定したパスが、ディレクトリではなくファイルになっている可能性があります。指定するパスはディレクトリでなければなりません。

解決法

入力ミスがないか確認してください。ファイル書式中の BASE コマンドの後に指定したパスを確認してください。ファイルではなくディレクトリが指定されていることを確認してください。

cachefspack: 共有オブジェクトを得られません。

原因

実行可能ファイルが壊れているか、その形式を認識できません。

解決法

実行可能ファイルを交換してください。

cachefspack: *filename*

- ファイルを pack できません。アクセス権がありません。

原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

解決法

適切なアクセス権を取得してください。

cachefspack: *filename*

- ファイルを pack できません。

ファイルまたはディレクトリがありません。

原因

正しいファイルまたはディレクトリがありません。

解決法

入力ミスがないか確認してください。

cachefspack: *filename* - ファイルを pack できません。

NFS のファイルハンドルが無効です。

原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

cachefspack: *filename* - ファイルを pack できません。

システムコールに割り込みがかかりました。

原因

コマンドの実行中に間違って Control + C を押した可能性があります。

解決法

このコマンドを再度実行してください。

cachefspack: *filename* - ファイルを pack できません。I/O エラー

原因

ハードウェアの障害の可能性があります。

解決法

ハードウェアの接続を確認してください。

cachefspack: *filename* - ファイルを pack できません。

デバイス上に十分な領域がありません。

原因

キャッシュのディスク容量が不足しています。

解決法

ディスク容量を増やしてキャッシュのサイズを大きくする必要があります。

cachefspack: *filename*

- ファイルを unpack できません。アクセス権がありません。

原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

解決法

適切なアクセス権を取得してください。

cachefspack: *filename* - ファイルを unpack できません。

ファイルまたはディレクトリがありません。

原因

正しいファイルまたはディレクトリがありません。

解決法

入力ミスがないか確認してください。

cachefspack: *filename* - ファイルを unpack できません。

NFS のファイルハンドルが無効です。

原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

cachefspack: *filename* - ファイルを unpack できません。

システムコールに割り込みがかかりました。

原因

コマンドの実行中に間違っ Control + C を押した可能性があります。

解決法

このコマンドを再度実行してください。

cachefspack: *filename* - ファイルを unpack できません。I/O エラー

原因

ハードウェアの障害の可能性があります。

解決法

ハードウェアの接続を確認してください。

cachefspack: `d`, `i`, `p`, `u` オプションのどれか 1 つを指定できます。

原因

コマンドに対して上記オプションのうち複数のオプションが指定されています。

解決法

オプションは 1 つだけ選択してください。

cachefspack: 環境変数が見つかりません。

原因

構成ファイル中で \$ で指定されている環境変数を設定していません。

解決法

環境変数を適切な場所に定義してください。

cachefspack: LIST コマンドをスキップします - 動作中の base はありません

原因

LIST コマンドが構成ファイル内にありますが、対応する BASE コマンドがありません。

解決法

BASE コマンドを定義してください。

CacheFS の統計情報の収集 (作業マップ)

次の作業マップは、CacheFS の統計情報の収集に関連する手順を示しています。この表に記載されている手順はすべてオプションです。

作業	説明	参照先
ログgingsの設定	cachefslog コマンドを使用して、CacheFS ファイルシステムのログgingsを設定する。	623 ページの「CacheFS ログgingsを設定する方法」

作業	説明	参照先
ログファイルの検索	cachefslog コマンドを使用してログファイルの位置を特定する。	624 ページの「CacheFS ログファイルの場所を調べる方法」
ロギングの停止	cachefslog コマンドを使用してロギングを停止する。	624 ページの「CacheFS ロギングを停止する方法」
キャッシュサイズの表示	cachefswssize コマンドを使用してキャッシュサイズを表示する。	625 ページの「作業セット (キャッシュ) のサイズを表示する方法」
キャッシュ統計情報の表示	cachefsstat コマンドを使用して統計情報を表示する。	626 ページの「CacheFS 統計情報を表示する方法」

CacheFS の統計情報の収集

CacheFS の統計情報を収集すると、次の作業を行うことができます。

- 適切なキャッシュサイズを判断する。
- キャッシュのパフォーマンスを監視する。

これらの統計情報を使用すると、キャッシュサイズと望ましいパフォーマンスを選択して調整できます。

CacheFS 統計情報コマンドには次のものがあります。

コマンド	マニュアルページ	説明
cachefslog	cachefslog (1M)	ログファイルの位置を指定します。また、このコマンドでは、統計情報が現在どこに記録されているかが表示されるので、ロギングを中止できます。
cachefswssize	cachefswssize (1M)	ログファイルを解釈して推奨キャッシュサイズを表示します。

コマンド	マニュアルページ	説明
<code>cachefsstat</code>	<code>cachefsstat (1M)</code>	特定のファイルシステム、またはすべての CacheFS ファイルシステムに関する統計情報を表示します。このコマンドの出力に含まれる情報は、キャッシュから直接取り出されます。

注 – CacheFS 統計情報コマンドは、どのディレクトリから実行してもかまいません。ただし、`cachefswssize` コマンドを実行するには、スーパーユーザーにならなければなりません。

CacheFS 統計の累計は、ログファイルの作成時から始まります。作業時間が終わったら、`cachefslog -h` コマンドを使用してロギングを停止してください。手順については、624 ページの「CacheFS ロギングを停止する方法」を参照してください。

CacheFS 統計情報コマンドを使用する前に、次の操作を実行する必要があります。

- `cfsadmin` コマンドを使用してキャッシュを設定する。
- 作成するログファイルに統計情報を収集できるように、適切な時間を決定する。この時間は、一般的な作業間隔に等しくする必要がある。たとえば、1 日、1 週間、1 カ月などにする。
- ログファイルの位置またはパスを選択する。ログファイルが大きくなっても対応できる程度の領域があることを確認する。ログファイルに統計情報を収集できる時間を長くするほど、大きな領域が必要になる。

注 – 次の手順は推奨する順序を示しています。異なる順序で作業してもかまいません。

CacheFS ロギングを設定する方法

1. ロギングを設定します。

```
$ cachefslog -f log-file-path /mount-point
```

`-f`

ロギングを設定する。

`log-file-path`

ログファイルの位置を指定する。ログファイルは、`vi` などのエディタで作成する標準ファイル。

/mount-point 統計情報を収集するマウントポイント (CacheFS ファイルシステム) を指定する。

2. ログファイルを正しく設定したかどうかを確認します。

```
$ cachefslog /mount-point
```

例 — CacheFS ロギングを設定する

次の例は、`/var/tmp/samlog` ログファイルを設定して、`/home/sam` ディレクトリに関する統計情報を収集する方法を示しています。

```
$ cachefslog -f /var/tmp/samlog /home/sam  
/var/tmp/samlog: /home/sam
```

CacheFS ログファイルの場所を調べる方法

オプションを指定せずに `cachefslog` コマンドを使用して、特定のマウントポイントに対するログファイルの場所を調べることもできます。

```
$ cachefslog /mount-point
```

/mount-point は、統計情報を表示する CacheFS ファイルシステムを示します。

次の例は、ログファイルが設定されている場合の表示を示します。ログファイルは、`/var/tmp/stufflog` にあります。

```
$ cachefslog /home/stuff  
/var/tmp/stufflog: /home/stuff
```

次の例は、指定したファイルシステムのログファイルが設定されていないことを示しています。

```
$ cachefslog /home/zap  
not logged: /home/zap
```

CacheFS ロギングを停止する方法

ロギングを停止するには、`cachefslog -h` オプションを使用します。

```
$ cachefslog -h /mount-point
```

次の例は、`/home/stuff` に対するロギングを停止する方法を示しています。

```
$ cachefslog -h /home/stuff  
not logged: /home/stuff
```


上記の例と異なるシステム応答が表示される場合は、ロギングが正常に停止されていません。正しいログファイル名とマウントポイントを指定したかどうかを確認してください。

作業セット (キャッシュ) のサイズを表示する方法

キャッシュサイズを増やすべきかどうかを確認できます。または、特定のマウントポイントに関して前回 `cachefswssize` コマンドを使用した後の作業に基づいて、理想的なキャッシュサイズを決定することもできます。

1. クライアントシステムでスーパーユーザーになります。
2. 現在のキャッシュサイズとロギングされた最大キャッシュサイズを表示します。

```
# cachefswssize log-file-path
```

詳細については、`cachefswssize (1M)` のマニュアルページを参照してください。

例 — 作業セット (キャッシュ) のサイズを表示する

次の例で、「end size」とは `cachefswssize` コマンドを実行した時点のキャッシュサイズです。「high water size」とは、ロギングが発生した時間枠内のキャッシュの最大サイズです。

```
# cachefswssize /var/tmp/samlog

/home/sam
    end size: 10688k
  high water size: 10704k

/
    end size: 1736k
  high water size: 1736k

/opt
    end size: 128k
  high water size: 128k

/nfs/saturn.dist
    end size: 1472k
  high water size: 1472k

/data/abc
    end size: 7168k
  high water size: 7168k

/nfs/venus.svr4
    end size: 4688k
  high water size: 5000k
```

```
/data
  end size: 4992k
  high water size: 4992k

total for cache
  initial size: 110960k
  end size: 30872k
  high water size: 30872k
```

CacheFS の統計情報の表示

特定の CacheFS ファイルシステムに関する情報を表示できます。次の表は、統計情報の出力時に表示される用語を示しています。

表 41-2 CacheFS 統計情報の用語

用語	説明
キャッシュのヒット率	キャッシュのヒット率対ミスヒット率の比と、それに続く実際のヒット数とミスヒット数。キャッシュヒットは、ユーザーがファイル操作を実行したいときに、そのファイルが実際にはキャッシュ内にあると発生する。キャッシュのミスヒットは、ファイルがキャッシュにないときに発生する。サーバーにかかる負荷は、キャッシュのミスヒット数、整合性チェック数、および変更数の合計である
整合性チェック	実行された整合性チェックの回数、合格回数、不合格回数
変更数 (modifies)	変更操作の回数。書き込みや作成など

CacheFS 統計情報を表示する方法

`cachefsstat` コマンドを使用して統計情報を表示します。この操作はいつでも実行できます。たとえば、ロギングを設定しなくても統計情報を表示できます。

```
$ cachefsstat /mount-point
```

`/mount-point` は、統計情報を表示する CacheFS ファイルシステムを示します。

マウントポイントを指定しなければ、マウントされているすべての CacheFS ファイルシステムに関する統計情報が表示されます。

詳細については、`cachefsstat (1M)` のマニュアルページを参照してください。

例 — CacheFS 統計情報を表示する

この例は、キャッシュされたファイルシステム `/home/sam` に関する統計情報の表示方法を示しています。

```
$ cachefsstat /home/sam
  cache hit rate: 73% (1234 hits, 450 misses)
  consistency checks: 700 (650 pass, 50 fail)
    modifies: 321
garbage collection: 0
```


第 42 章

追加スワップ空間の構成 (手順)

この章では、Solaris リリースをインストールした後で追加のスワップ空間を構成するためのガイドラインと手順について説明します。

この章で説明する手順は次のとおりです。

- 636 ページの「スワップファイルを作成して使用可能にする方法」
- 638 ページの「不要になったスワップ空間を削除する方法」

この章の内容は次のとおりです。

- 629 ページの「スワップ空間について」
- 631 ページの「スワップ空間の追加が必要かどうかを調べる方法」
- 632 ページの「スワップ空間の割り当て方法」
- 633 ページの「スワップ空間の計画」
- 634 ページの「スワップリソースの監視」
- 635 ページの「スワップ空間の追加」

スワップ空間について

システム管理者は、次の事柄を決定する上で SunOS のスワップ機構を理解しておく必要があります。

- スワップ空間の要件
- スワップ空間と TMPFS ファイルシステムとの関係
- スワップ空間に関連するエラーメッセージからの復元

スワップ空間と仮想メモリー

Solaris ソフトウェアは、一時記憶域にファイルシステムではなくディスクスライスを使用します。これらのスライスを「スワップ」スライスと呼びます。スワップスライスは、システムの物理メモリーが不足し現在のプロセスを処理することができないときに、仮想メモリー記憶域として使用されます。

仮想メモリーシステムは、ディスク上のファイルの物理コピーをメモリー内の仮想アドレスに対応付けます。これらのマッピングに関するデータが入った物理メモリーページは、ファイルシステム内の通常ファイルまたはスワップ空間から読み直されます。メモリーをバックアップしているディスク空間に割り当てられる ID はわからないため、スワップ空間から読み直されたメモリーは *anonymous* メモリーと呼ばれます。

Solaris 環境には、「仮想スワップ空間」という概念が導入されています。これは、*anonymous* メモリーページとこれらのページを実際にバックアップする物理記憶域（またはディスク上にとられたバックアップ用のスワップ空間）の間に位置する層です。システムの仮想スワップ空間は、すべての物理（ディスク上にとられたバックアップ用のスワップ空間）スワップ空間と現在使用可能な物理メモリーの一部の合計に等しくなります。

仮想スワップ空間の長所は次のとおりです。

- 仮想スワップ空間が物理（ディスク）記憶域に対応していなくてもかまわないので、大きな物理スワップ空間を確保する必要がなくなる。
- SWAPFS という疑似ファイルシステムが、*anonymous* メモリーページのアドレスを提供する。SWAPFS はメモリーページの割り当てを制御するので、ページに対する処理を柔軟に決定できる。たとえば、ディスク上にとられたバックアップ用のスワップ記憶域のページ要件を変更できる。

スワップ空間と TMPFS ファイルシステム

Solaris 環境では、TMPFS ファイルシステムは `/etc/vfstab` ファイル内のエントリによって自動的に稼働されます。TMPFS ファイルシステムは、ファイルとそれに関連付けられた情報をディスクではなくメモリー（`/tmp` ディレクトリ）に格納するので、これらのファイルへのアクセスが高速になります。この機能によって、コンパイラや DBMS 製品のように `/tmp` の使用量の大きいアプリケーションの場合は、パフォーマンスが大幅に改善されます。

TMPFS ファイルシステムは、システムのスワップリソースから `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` ディレクトリ内の領域を使い果たすと、スワップ空間も使い果たしたことになります。したがって、`/tmp` ディレクトリの使用量が大きいアプリケーションの場合は、スワップ空間の使用状況を監視しなければ、システムがスワップ空間を使い果たす可能性があります。

TMPFS を使用したいがスワップリソースが限られている場合は、次の方法を使用してください。

- サイズオプション (-o size) を指定して TMPFS ファイルシステムをマウントし、TMPFS が使用できるスワップリソースを制御する。
- コンパイラの TMPDIR 環境変数を使用して、より大きな別のディレクトリを指す。

コンパイラの TMPDIR 変数を使用すると、コンパイラが /tmp ディレクトリを使用するかどうかだけが制御される。この変数は、他のプログラムによる /tmp ディレクトリの使用には影響しない。

ダンプデバイスとしてのスワップ空間

通常、ダンプデバイスとは、システムクラッシュダンプ情報を格納するために予約されているディスク領域のことです。デフォルトでは、システムのスワップ空間は適切なスワップパーティションになるように設定されます。可能であれば、クラッシュダンプの信頼性を高めたり、システム障害が発生した後のリポート時間を短縮したりする代わりに、代替ディスクパーティションを「専用ダンプデバイス」として設定してください。専用ダンプデバイスの設定は、dumpadm コマンドで行えます。詳細については、『Solaris のシステム管理 (上級編)』の「システムクラッシュ情報の管理 (手順)」を参照してください。

Solaris ボリュームマネージャなどのボリュームマネージャを使用してディスクを管理している場合は、専用ダンプデバイスを Solaris ボリュームマネージャの管理下に置かないように設定してください。スワップ領域は、Solaris ボリュームマネージャの管理下に保管することをお勧めします。ただし、使いやすさとパフォーマンスの理由から、Solaris ボリュームマネージャの管理下で動作しないディスクを専用ダンプデバイスとして設定してください。

スワップ空間の追加が必要かどうかを調べる方法

スワップ空間の追加が必要かどうかを調べるには、swap -l コマンドを使用します。

たとえば、次の swap -l コマンドの出力は、このシステムのスワップ空間がほぼ完全に使い果たされたか、割り当て率が 100% に達していることを示しています。

```
% swap -l
swapfile          dev      swaplo blocks   free
/dev/dsk/c0t0d0s1 136,1    16 1638608    88
```

システムのスワップ空間の割り当て率が 100% に達すると、アプリケーションのメモリーページが一時的にロックしてしまいます。アプリケーションエラーは発生しませんが、システムパフォーマンスが低下する可能性があります。

システムにスワップ空間を追加する方法については、636 ページの「スワップファイルを作成して使用可能にする方法」を参照してください。

スワップ関連のエラーメッセージ

次のメッセージは、アプリケーションがさらに `anonymous` メモリを取得しようとしたが、バックアップ用のスワップ空間が残っていなかったことを示します。

```
application is out of memory
```

```
malloc error 0
```

```
messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]  
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

TMPFS 関連のエラーメッセージ

次のメッセージは、ファイルに書き込むときにページを割り当てることができない場合に表示されます。この問題は、TMPFS が許容限度を超えて書き込もうとしたときや、現在実行されているプログラムが大量のメモリーを使用している場合に発生することがあります。

```
directory: File system full, swap space limit exceeded
```

次のメッセージは、TMPFS が新しいファイルやディレクトリの作成中に物理メモリーを使い果たしたことを意味します。

```
directory: File system full, memory allocation failed
```

TMPFS 関連のエラーメッセージから回復する方法については、TMPFS (7FS) のマニュアルページを参照してください。

スワップ空間の割り当て方法

最初に、スワップ空間は Solaris インストールプロセスの一部として割り当てられます。インストールプログラムによるディスクスライスの自動レイアウトを選択し、スワップファイルのサイズを手作業で変更しなければ、Solaris インストールプログラムはデフォルトのスワップ領域 (512M バイト) を割り当てます。

Solaris 9 リリースよりインストールプログラムは利用可能な最初のディスクシリンダ (通常はシリンダ 0) にスワップ空間を割り当てるようになりました。この配置によって、デフォルトのディスクレイアウト時にはルート (/) ファイルシステムに最大のスワップ空間を割り当てることができ、アップグレード時にはルート (/) ファイルシステムを拡張できます。

スワップ空間の割り当て方法については、633 ページの「スワップ空間の計画」を参照してください。

スワップファイルを作成すると、スワップ空間をシステムに追加できます。スワップファイルの作成方法については、635 ページの「スワップ空間の追加」を参照してください。

/etc/vfstab ファイル

システムをインストールすると、スワップスライスとスワップファイルが /etc/vfstab ファイル内に列挙されます。スワップスライスとスワップファイルは、システムのブート時に /sbin/swapadd スクリプトによって有効になります。

/etc/vfstab ファイル内のスワップデバイスエントリには、次の情報が入っています。

- スワップスライスまたはスワップファイルのフルパス名
- スワップのファイルシステムタイプ

スワップファイルが入っているファイルシステムは、スワップファイルが有効になる前にマウントしておかなければなりません。このため、/etc/vfstab ファイル内で、ファイルシステムをマウントするエントリが、スワップファイルを有効にするエントリより前に入っていることを確認してください。

スワップ空間の計画

スワップ空間のサイズを決定する上で最も重要な要素は、システムのソフトウェアアプリケーションの要件です。たとえば、コンピュータ支援設計シミュレータ、データベース管理製品、トランザクションモニター、地質分析システムなどの大型アプリケーションは、200 ~ 1000M バイトのスワップ空間を消費することがあります。

アプリケーションのスワップ空間の要件については、アプリケーションベンダーに問い合わせてください。

アプリケーションベンダーに問い合わせてもスワップ空間の要件を判断できない場合は、次のシステムタイプ別のガイドラインに従ってスワップ空間を割り当ててください。

システムタイプ	スワップ空間のサイズ	専用ダンプデバイスのサイズ
約 4G バイトの物理メモリーを備えたワークステーション	1G バイト	1G バイト
約 8G バイトの物理メモリーを備えた中型のサーバー	2G バイト	2G バイト
約 16 ~ 128G バイトの物理メモリーを備えたハイエンドサーバー	4G バイト	4G バイト

一般的なガイドラインのほかに、次の場合のスワップ空間やディスク領域の割り当ても検討してください。

- 専用ダンプデバイス
- 大型アプリケーション (コンパイラなど) が /tmp ディレクトリを使用するかどうかを決定する。次に、TMPFS によって使用される追加のスワップ空間を割り当てる。TMPFS については、630 ページの「スワップ空間と TMPFS ファイルシステム」を参照

スワップリソースの監視

/usr/sbin/swap コマンドを使用してスワップ領域を管理します。2つのオプション -l と -s は、スワップリソースに関する情報を表示します。

swap -l コマンドを使用すると、システムのスワップ領域を確認できます。有効になっているスワップデバイスやファイルは、swapfile カラムの下に表示されます。

```
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1      16 1638608 1600528
```

swap -s コマンドを使用すると、スワップリソースを監視できます。

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

used 値と available 値の合計は、システム上の合計スワップ空間に等しくなります。これには、物理メモリーの一部とスワップデバイス (またはファイル) が含まれます。

使用可能なスワップ空間と使用済みスワップ空間の容量 (swap -s の出力内) を使用して、時間経過に伴うスワップ空間の使用状況を監視できます。システムパフォーマンスが適正であれば、swap -s を使用するとどの程度のスワップ空間が使用可能であ

るかがわかります。システムパフォーマンスが低下したときは、使用可能なスワップ空間の容量をチェックして減少していないかどうかを調べてください。これによって、システムに対するどのような変更が原因でスワップ空間の使用量が増大したかを識別できます。

このコマンドを使用するときには、カーネルとユーザープロセスが物理メモリーをロックして解除するたびに、スワップに使用できる物理メモリーの容量が動的に変化するので注意してください。

注 - `swap -l` コマンドではスワップ空間が 512 バイトのブロック数として表示され、`swap -s` コマンドでは 1024 バイトのブロック数として表示されます。 `swap -l` の場合はスワップ空間の計算に物理メモリーが含まれないので、`swap -l` で表示されたブロック数を合計して K バイト数に換算すると、(`swap -s` で出力される) `used` と `available` の値の合計よりも少なくなります。

表 42-1 に、`swap -s` コマンドの出力とその説明を示します。

表 42-1 `swap -s` コマンドの出力

キーワード	説明
<code>bytes allocated</code>	現在バックアップ用の記憶域 (ディスク上にとられたバックアップ用のスワップ空間) として使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数
<code>reserved</code>	現在は割り当てられていないが、後から使用できるようにメモリーによって回収されるスワップ空間の合計容量を表す 1024 バイトのブロック数
<code>used</code>	割り当て済みまたは予約済みのスワップ空間の合計容量を表す 1024 バイトのブロック数
<code>available</code>	後から予約や割り当てに使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数

スワップ空間の追加

システム構成を変更して新しいソフトウェアパッケージをインストールした後に、スワップ空間を追加しなければならないことがあります。その場合に望ましいのは、`mkfile` コマンドと `swap` コマンドを使用して、既存の UFS または NFS ファイルシステムの一部を補助スワップ空間として指定する方法です。次の節で説明するように、これらのコマンドを使用すると、ディスクをパーティションに分割し直さなくても、スワップ空間を追加できます。

スワップ空間を追加するもう 1 つの方法は、既存のディスクをパーティションに分割し直すか、別のディスクを追加することです。ディスクをパーティションに分割し直す方法については、第 32 章を参照してください。

スワップファイルの作成

次の手順でスワップファイルを作成します。

- `mkfile` コマンドを使用してスワップファイルを作成する。
- `swapon` コマンドを使用してスワップファイルを有効にする。
- システムのブート時に自動的に有効になるように、スワップファイルのエントリを `/etc/vfstab` ファイルに追加する。

mkfile コマンド

`mkfile` コマンドは、NFS のマウント済みスワップ空間またはローカルスワップ空間に使用できるファイルを作成します。スティッキビットが設定され、ファイルは 0 が埋め込まれます。スワップファイルのサイズは、バイト数 (デフォルト) として指定するか、接尾辞 `k`、`b`、`m` を使用して、それぞれ K バイト数、ブロック数、M バイト数として指定できます。

次の表に、`mkfile` コマンドのオプションを示します。

表 42-2 `mkfile` コマンドのオプション

オプション	説明
<code>-n</code>	空のファイルを作成する。サイズは表示されるが、データが書き込まれるまでディスクブロックは割り当てられない。
<code>-v</code>	作成されたファイル名とサイズが表示される。



注意 `--n` オプションは、NFS スワップファイルの作成時のみ使用してください。

▼ スワップファイルを作成して使用可能にする方法

1. スーパーユーザーになります。

`root` の権限がなくてもスワップファイルを作成できます。しかし、スワップファイルが誤って上書きされないように、`root` を所有者にしておくといでしょう。

2. 必要に応じて、スワップファイル用のディレクトリを作成します。

3. スワップファイルを作成します。

```
# mkfile nnn[k|b|m] filename
```

指定したサイズ「*nnn*」(K バイト、バイト、または M バイト) とファイル名でスワップファイルが作成されます。

4. スワップファイルを有効にします。

```
# /usr/sbin/swap -a /path/filename
```

絶対パス名を使用してスワップファイルを指定しなければなりません。スワップファイルが追加され、ファイルシステムがマウント解除されるか、またはシステムがリブートされるまで使用可能になります。プロセスまたはプログラムがスワップファイルにスワップしているときは、ファイルシステムのマウントを解除できないことに注意してください。

5. 次のように、ファイルのフルパス名を指定し、ファイルシステムのタイプとして **swap** を指定して、スワップファイルのエントリを **/etc/vfstab** ファイルに追加します。

```
/path/filename - - swap - no -
```

6. スワップファイルが追加されたことを確認するには、次のように入力します。

```
$ /usr/sbin/swap -l
```

例 — スワップファイルを作成して使用可能にする

次の例は、`/files/swapfiles` という 100M バイトのスワップファイルを作成する方法を示しています。

```
# mkdir /files
# mkfile 100m /files/swapfile
# swap -a /files/swapfile
# vi /etc/vfstab
(エントリがスワップファイル用に追加される):
/files/swapfile - - swap - no -
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1    16 1638608 1600528
/files/swapfile   -      16 204784 204784
```

スワップファイルの削除

不要になったスワップ空間は、削除することができます。

▼ 不要になったスワップ空間を削除する方法

1. スーパーユーザーになります。

2. スワップ空間を削除します。

```
# /usr/sbin/swap -d /path/filename
```

スワップファイル名が削除されるので、スワッピングに使用できなくなります。
ファイルそのものは削除されません。

3. `/etc/vfstab` ファイルを編集して、スワップファイルのエントリを削除します。

4. 他の目的に使用できるようにディスク領域を復元します。

```
# rm /path/filename
```

スワップ空間がファイルの場合は削除します。また、スワップ空間が別のスライスに入っていて、不要なことがわかっている場合は、新しいファイルシステムを作成してマウントします。

ファイルシステムのマウントについては、第 40 章を参照してください。

5. 削除したスワップファイルが使用できなくなっていることを確認します。

```
# swap -l
```

例 — 不要になったスワップ空間を削除する

次の例は、`/files/swapfile` スワップファイルを削除する方法を示しています。

```
# swap -d /files/swapfile
# (削除したスワップエントリを /etc/vfstab ファイルから削除する)
# rm /files/swapfile
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/dsk/c0t0d0s1 136,1      16 1638608 1600528
```

第 43 章

UFS ファイルシステムの整合性 チェック (手順)

この章では、UFS ファイルシステムの整合性チェックに関する概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 649 ページの「ファイルシステムのチェックが必要かを調べる方法」
- 650 ページの「ファイルシステムを対話式でチェックする方法」
- 651 ページの「UFS ファイルシステムを修復する方法」
- 653 ページの「不正なスーパーブロックを復元する方法」

この章の内容は次のとおりです。

- 639 ページの「ファイルシステムの整合性」
- 640 ページの「ファイルシステムの状態はどのように記録されるか」
- 642 ページの「fsck コマンドでチェックして修復される内容」
- 649 ページの「UFS ファイルシステムを対話式でチェックして修復する」
- 653 ページの「不正なスーパーブロックの復元」
- 655 ページの「fsck コマンドの構文とオプション」

fsck のエラーメッセージについては、『Solaris のシステム管理 (上級編)』の「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

この章で参照される UFS ファイルシステム構造の内容については、第 44 章を参照してください。

ファイルシステムの整合性

UFS ファイルシステムは、一連の内部テーブルを基にして使用済み i ノード、使用可能ブロックを特定します。これらの内部テーブルがディスク上のデータと正しく同期していないと、整合性が失われ、ファイルシステムの修復が必要になります。

次のような原因でオペレーティングシステムが異常終了すると、ファイルシステムの整合性が失われることがあります。

- 電源障害
- 不注意によるシステム電源の切断
- 正しいシャットダウン手順以外の方法によるシステム電源の切断
- カーネル内のソフトウェアエラー

ファイルシステムの整合性が失われることは重大ですが、頻繁に起きるものではありません。システムをブートすると、ファイルシステムの整合性チェックが (`fsck` コマンドを使用して) 自動的に実行されます。ほとんどの場合は、このファイルシステムのチェックによって問題が修復されます。

`fsck` コマンドは、ファイルシステム上に配置されているが参照不可能なファイルとディレクトリを `lost+found` ディレクトリに入れます。参照不可能なファイルとディレクトリの名前として `i` ノード番号が割り当てられます。`lost+found` ディレクトリが存在しない場合は、`fsck` コマンドによって作成されます。`lost+found` ディレクトリ内の領域が足りない場合は、`fsck` コマンドによってそのサイズが拡張されます。

`i` ノードについては、667 ページの「`i` ノード」を参照してください。

ファイルシステムの状態はどのように記録されるか

`fsck` コマンドは、スーパーブロックに格納された状態フラグを使用して、ファイルシステムの状態を記録します。また、このフラグを使用して、ファイルシステムの整合性をチェックする必要があるか判断します。このフラグは、ブート時に `/sbin/rcS` スクリプトによって使用されるか、`fsck -m` コマンドによって使用されます。`fsck -m` コマンドの結果を無視する場合は、状態フラグの設定に関係なく、すべてのファイルシステムをチェックできます。

スーパーブロックについては、666 ページの「スーパーブロック」を参照してください。

表 43-1 に状態フラグの値とその説明を示します。

表 43-1 ファイルシステムの状態フラグの値

状態フラグの値	説明
FSACTIVE	ファイルシステムのマウント後、変更されると、状態フラグが FSACTIVE に設定される。ファイルシステムの整合性が失われている可能性がある。変更後のメタデータがディスクに書き込まれるまでは、ファイルシステムに FSACTIVE マークが付けられる。ファイルシステムが正常にマウント解除されると、状態フラグは FSCLEAN に設定される。FSACTIVE フラグが設定されたファイルシステムは、整合性が失われている可能性があるの で、fsck コマンドでチェックしなければならない。
FSBAD	ルート (/) ファイルシステムが、FSCLEAN でも FSSTABLE でもない状態のときにマウントされると、状態フラグが FSBAD に設定される。カーネルが、このファイルシステムの状態を FSCLEAN または FSSTABLE に変更することはない。ブートの処理の一部として、ルート (/) ファイルシステムに FSBAD フラグが設定された場合、ルートファイルシステムは読み取り専用としてマウントされる。ルートの raw デバイスに対して fsck コマンドを実行できる。その後で、ルート(/)ファイルシステムを読み取り / 書き込み用にマウントし直す。
FSCLEAN	ファイルシステムが正しくマウント解除された場合は、状態フラグが FSCLEAN に設定される。FSCLEAN 状態フラグが設定されているファイルシステムは、システムのブート時にチェックされない。
FSLOG	UFS ログングを有効にしてファイルシステムがマウントされた場合は、状態フラグが FSLOG に設定される。システムのブート時、状態フラグが FSLOG のファイルシステムはチェックされない。
FSSTABLE	ファイルシステムはマウントされている (またはされた) が、前回のチェックポイント (sync または fsflush) 以後に変更がなかった。チェックポイントは、通常は 30 秒ごとに発生する。たとえば、カーネルはファイルシステムがアイドル状態かどうかを定期的にチェックし、アイドル状態であれば、スーパーブロック内の情報をディスクにフラッシュさせて FSSTABLE マークを設定する。システムがクラッシュした場合、ファイルシステムの構造は正しいが、少量のデータが失われている可能性がある。FSSTABLE マークが付いたファイルシステムは、マウント前のチェックをスキップできる。ファイルシステムの状態が FSCLEAN、FSSTABLE、または FSLOG のいずれでもない場合は、mount コマンドによってファイルシステムを読み取り / 書き込み用にマウントできない。

次の表に、fsck コマンドを使用して、初期状態に基づいて状態フラグを変更する方法を示します。

表 43-2 fsck による状態フラグの変更内容

初期状態 : fsck の実行前	fsck の実行後	
	エラーなし	すべてのエラーを修正済み エラーが未修正
不明	FSSTABLE	FSSTABLE 不明

表 43-2 fsck による状態フラグの変更内容 (続き)

初期状態: fsck の実行前	fsck の実行後		
	エラーなし	すべてのエラーを修正済み	エラーが未修正
FSACTIVE	FSSTABLE	FSSTABLE	FSACTIVE
FSSTABLE	FSSTABLE	FSSTABLE	FSACTIVE
FSCLEAN	FSCLEAN	FSSTABLE	FSACTIVE
FSBAD	FSSTABLE	FSSTABLE	FSBAD
FSLOG	FSLOG	FSLOG	FSLOG

fsck コマンドでチェックして修復される内容

この節では、ファイルシステムの通常の処理中に発生する問題、原因、fsck コマンド(チェックおよび修復ユーティリティ)で検出される問題、およびそれらの修正方法について説明します。

不整合が発生する原因

就業日には毎日多数のファイルが作成、変更、または削除されます。ファイルが変更されるたびに、オペレーティングシステムは一連のファイルシステムの更新処理を実行します。これらの更新処理がディスクに確実に書き込まれると、ファイルシステムの整合性が保たれます。

ユーザープログラムが書き込みなどの、ファイルシステムを変更する処理を実行すると、書き込まれるデータはまずカーネルのインコアバッファーにコピーされます。一般に、ディスクの更新は非同期に処理されます。このため、ユーザープロセスは、書き込みシステムコールが値を返した後すぐに処理を続けることができますが、実際のデータの書き込みは、ずいぶん後に実行されることもあります。したがって、ディスク上にあるファイルシステムは、インコア情報で表されるファイルシステムの状態から常に遅延することになります。

別の目的にバッファーが必要になったり、カーネルが fsflush デーモンを自動的に(30 秒間隔で)実行すると、インコア情報を反映するようにディスク情報が更新されます。システムがインコア情報を書き込まずに停止すると、ディスク上のファイルシステムの整合性がなくなります。

ファイルシステムの整合性は、さまざまな原因で失われることがあります。最も一般的な原因は、オペレータのエラーとハードウェア障害です。

システムを正しくシャットダウンしなかったり、マウントされているファイルシステムが正しくオフラインにされないと、「クリーンでない停止」が原因で問題が発生することがあります。クリーンでないシャットダウンを防ぐには、システムをシャットダウンしたり、ディスクをドライブから物理的に取り出したり、ディスクをオフライン状態にしたりする前に、ファイルシステムの現在の状態をディスクに書き込まなければなりません。つまり、「同期」させなければなりません。

また、ハードウェアの欠陥が原因で整合性が失われることもあります。ディスクドライブ上ではいつでもブロックが損傷する可能性があり、ディスクコントローラが正常に機能しなくなる可能性があります。

整合性がチェックされる UFS 構成要素

この節では、UFS ファイルシステムの構成要素、つまりスーパーブロック、シリンダグループブロック、i ノード、間接ブロック、データブロックに `fsck` コマンドが適用する整合性チェックの種類について説明します。

UFS ファイルシステム構造については、666 ページの「UFS ファイルシステムのシリンダグループの構造」を参照してください。

スーパーブロックのチェック

スーパーブロックには集計情報が格納されており、UFS ファイルシステム内で最も破損しがちな構成要素です。ファイルシステムの i ノードやデータブロックが変更されるたびに、スーパーブロックも変更されます。CPU が停止した場合、直前のコマンドが `sync` コマンドでなければ、スーパーブロックはほぼ確実に破損しています。

スーパーブロックの不整合は、次の面からチェックされます。

- ファイルシステムのサイズ
- i ノード数
- 空きブロック数
- 空き i ノード数

ファイルシステムのサイズと i ノードリストのサイズのチェック

ファイルシステムのサイズは、スーパーブロックと i ノードリストに使用されるブロック数よりも大きくなければなりません。i ノード数は、ファイルシステムの最大許容数よりも小さくなければなりません。i ノードは、ファイルに関するすべての情報を表します。ファイルシステムのサイズとレイアウト情報は、`fsck` コマンドにとって最も重要な情報部分です。これらのサイズはファイルシステムの作成時に静的に決められるため、実際にチェックする方法はありません。ただし、`fsck` コマンドを使用してサイズが妥当な範囲内にあるかどうかはチェックできます。ファイルシステムの他のすべてのチェックを行うには、これらのサイズが正確でなければなりません。`fsck` コマンドが一次スーパーブロックの静的パラメータ内に不正な情報を検出すると、オペレータに代替スーパーブロックの位置を指定するように促します。

UFS ファイルシステム構造の詳細については、666 ページの「UFS ファイルシステムのシリンダグループの構造」を参照してください。

空きブロック数のチェック

空きブロック数は、シリンダグループのブロックマップに格納されます。fsck コマンドは、空きマーク付きのすべてのブロックがファイルによって使用されていないかどうかをチェックします。すべてのブロックをチェックし終わると、fsck コマンドは空きブロック数と i ノードによって使用されるブロック数の合計がファイルシステム内の合計ブロック数に等しくなるかどうかをチェックします。ブロックマップ内に間違いがあると、fsck コマンドはブロックが割り当てられている状態のまま構築し直します。

スーパーブロック内の集計情報には、ファイルシステム内の空きブロックの合計数のカウントが入っています。fsck コマンドは、このブロック数をファイルシステム内で見つかった空きブロック数と比較します。数が一致しなければ、fsck コマンドはスーパーブロック内の空きブロック数を実際の空きブロック数で置き換えます。

空き i ノード数のチェック

スーパーブロック内の集計情報には、ファイルシステム内の空き i ノード数が入っています。fsck コマンドは、この i ノード数をファイルシステム内で見つかった空き i ノード数と比較します。数が一致しなければ、fsck はスーパーブロック内の空き i ノード数を実際の空き i ノード数で置き換えます。

i ノード

i ノードリストは、i ノード 2 から順番にチェックされます (i ノード 0 と i ノード 1 は予約済み)。各 i ノードの不整合は、次の面からチェックされます。

- 形式とタイプ
- リンク数
- 重複ブロック
- 不正なブロック番号
- i ノードのサイズ

i ノードの形式とタイプ

各 i ノードには、そのタイプと状態を記述するモードのワードが入っています。i ノードには、次の 9 つのタイプがあります。

- 通常ファイル
- ディレクトリ
- ブロック型特殊ファイル
- キャラクタ型特殊ファイル

- FIFO (名前付きパイプ)
- シンボリックリンク
- シャドウ (ACL で使用される)
- 属性ディレクトリ
- ソケット

i ノードの状態は、次の3つに分かれています。

- 割り当て済み
- 未割り当て
- 不完全に割り当て済み

ファイルシステムが作成されると、一定数の i ノードが確保されますが、必要になるまでは割り当てられません。割り当て済みの i ノードとは、ファイルを指す i ノードです。未割り当ての i ノードは、ファイルを指さないのが空のはずです。不完全に割り当て済みの状態は、i ノードが正しくフォーマットされていないことを意味します。たとえば、ハードウェア障害が原因で i ノードに不正なデータが書き込まれると、i ノードは不完全に割り当て済みの状態になることがあります。fsck コマンドが実行できる唯一の修正動作は、その i ノードを消去することです。

リンク数のチェック

各 i ノードには、そこにリンクされているディレクトリエントリ数が入っています。fsck コマンドは、ルートディレクトリから順番にディレクトリ構造全体を検査し、i ノードごとに実際のリンク数を計算して、各 i ノードのリンク数を検査します。

i ノードに格納されているリンク数が fsck コマンドによって判断された実際のリンク数と一致しない場合は、次の3つの状況が考えられます。

- 格納されたリンク数が 0 でなく、実際のリンク数が 0 の場合
この状況は、i ノードにリンクされているディレクトリエントリが存在しない場合に発生することがあります。この場合、fsck コマンドはリンクされていないファイルを lost+found ディレクトリに入れます。
- 格納されたリンク数が 0 でなく、実際のリンク数も 0 でないが、2つのリンク数が等しくない場合
この状況は、ディレクトリエントリが追加または削除されたが、i ノードが更新されていない場合に発生することがあります。この場合、fsck コマンドは格納されたリンク数を実際のリンク数で置き換えます。
- 格納されたリンク数が 0 で実際のリンク数が 0 でない場合
この場合、fsck コマンドは i ノードのリンク数を実際のリンク数に変更します。

重複ブロックのチェック

各 i ノードには、それが使用するすべてのブロックのリスト、またはリストを指すポインタ (間接ブロック) が入っています。間接ブロックは i ノードによって所有されるので、間接ブロックの整合性が失われると、それを所有する i ノードが直接影響を受けます。

fsck コマンドは、i ノードから使用される各ブロック番号を、割り当て済みブロックのリストと比較します。別の i ノードからすでにブロック番号が使用されていると、そのブロック番号は重複ブロックのリストに入れられます。それ以外の場合は、割り当て済みブロックのリストが更新され、ブロック番号が追加されます。

重複ブロックがあると、fsck コマンドは再び i ノードリストを調べて、各重複ブロックを使用する他の i ノードを検索します。i ノード内に大量の重複ブロックが入っている場合は、ファイルシステムに間接ブロックが正しく書き込まれていない可能性があります。どの i ノードにエラーがあるかを正確に判断することはできません。fsck コマンドは、保持する i ノードと消去する i ノードを選択するように促すプロンプトを表示します。

不正なブロック番号のチェック

fsck コマンドは、i ノードから使用される各ブロック番号をチェックして、その値が最初のデータブロック番号よりも大きく、ファイルシステム内の最後のデータブロック番号より小さいかどうかを調べます。ブロック番号がこの範囲に含まれない場合は、不正なブロック番号と見なされます。

間接ブロックがファイルシステムに正しく書き込まれていないことが原因で、i ノード内に不正なブロック番号が発見されることがあります。fsck コマンドはその i ノードの消去を促すプロンプトを表示します。

i ノードサイズのチェック

各 i ノードには、参照するデータブロック数が入っています。実際のデータブロック数は、割り当て済みのデータブロック数と間接ブロック数の合計です。fsck コマンドはデータブロック数を計算し、そのブロック数を i ノードから使用されるブロック数と比較します。i ノードに不正なブロック数が入っていると、fsck コマンドはその修正を促すプロンプトを表示します。

各 i ノードには、64 ビットのサイズフィールドがあります。このフィールドは、i ノードに関連付けられたファイル内の文字数 (データバイト数) を示します。i ノードのサイズフィールドに整合性があるかどうかは、サイズフィールド内の文字数を使用して、i ノードに関連付けるべきブロック数を計算し、その結果を i ノードから使用される実際のブロック数と比較して概算でチェックされます。

間接ブロック

間接ブロックは i ノードによって所有されます。したがって、間接ブロック内の整合性が失われると、それを所有する i ノードが影響を受けます。不整合は、次の面からチェックされます。

- すでに別の i ノードから使用されているブロック
- ファイルシステムの範囲に含まれないブロック番号

また、間接ブロックに対しても整合性チェックが実行されます。

データブロック

i ノードは、3 種類のデータブロックを直接または間接に参照できます。参照されるブロックは、すべて同じ種類でなければなりません。次の 3 種類のデータブロックがあります。

- プレーンデータブロック
- シンボリックリンクデータブロック
- ディレクトリデータブロック

プレーンデータブロックには、ファイルに格納される情報が入っています。シンボリックリンクデータブロックには、シンボリックリンクに格納されるパス名が入っています。ディレクトリデータブロックには、ディレクトリエントリが入っています。fsck コマンドはディレクトリデータブロックの妥当性しかチェックできません。

ディレクトリは、i ノードの mode フィールド内のエントリによって通常ファイルと区別されます。ディレクトリに関連付けられたデータブロックには、ディレクトリエントリが入っています。ディレクトリデータブロックの不整合は、次の面からチェックされます。

- 未割り当ての i ノードを指すディレクトリ内の i ノード番号
- ファイルシステム内の i ノード番号より大きいディレクトリ内の i ノード番号
- 「.」と「..」ディレクトリには許されないディレクトリ内の i ノード番号
- ファイルシステムから切り離されたディレクトリ

未割り当てディレクトリのチェック

ディレクトリデータブロック内の i ノード番号が未割り当て i ノードを指す場合、fsck コマンドはそのディレクトリエントリを削除します。この状況は、新しいディレクトリエントリが入っているデータブロックが変更されて書き出されたが、i ノードが書き込まれていない場合に発生します。また、警告なしに CPU が停止された場合にも発生します。

不正な i ノード番号のチェック

ディレクトリエントリの i ノード番号が i ノードリストの最後を超える位置を指す場合、fsck コマンドはそのディレクトリエントリを削除します。この状況は、不正なデータがディレクトリのデータブロックに書き込まれると発生します。

不正な「.」と「..」エントリ

「.」ディレクトリの i ノード番号は、ディレクトリデータブロックの最初のエントリでなければなりません。また、それ自体を参照しなければなりません。つまり、その値はディレクトリデータブロックの i ノード番号に等しくなければなりません。

「..」ディレクトリの i ノード番号は、ディレクトリデータブロックの第 2 のエントリでなければなりません。その値は、親ディレクトリの i ノード番号 (または、ディレクトリがルートディレクトリの場合は、それ自体の i ノード番号) に等しくなければなりません。

「.」と「..」ディレクトリのiノード番号が不正であれば、fsck コマンドは正しい値に置き換えます。ディレクトリへのハードリンクが複数個ある場合は、最初に見つかったハードリンクが「..」が指す実際の親であると見なされます。この場合、fsck コマンドは他の名前を削除するように促すプロンプトを表示します。

切り離されたディレクトリ

fsck コマンドは、ファイルシステム全体で参照関係をチェックします。ファイルシステムにリンクされていないディレクトリが見つかったら、fsck コマンドはそのディレクトリをファイルシステムの lost+found ディレクトリにリンクします。この状況は、iノードがファイルシステムに書き込まれたが、それに対応するディレクトリデータブロックが書き込まれていない場合にも発生することがあります。

通常データブロック

通常ファイルに関連付けられたデータブロックには、ファイルの内容が入っています。fsck コマンドは、通常ファイルのデータブロックの内容が有効かどうかはチェックしません。

fsck 要約メッセージ

fsck コマンドを対話式で実行して正常に終了すると、次のようなメッセージが表示されます。

```
# fsck /dev/rdisk/c0t0d0s7
** /dev/rdisk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)
#
```

fsck 出力の最後の行は、ファイルシステムについて次のような情報を記述します。

# files	使用中のiノード数
# used	使用中のフラグメント数
# free	未使用のフラグメント数
# frags	未使用の非ブロックフラグメント数

# blocks	未使用の完全ブロック数
% fragmentation	断片化の比率。ファイルシステム内の空きフラグメント × 100 / 全フラグメント

フラグメントについては、670 ページの「フラグメントサイズ」を参照してください。

UFS ファイルシステムを対話式でチェックして修復する

次の場合には、ファイルシステムを対話式でチェックする必要があります。

- マウントできない場合
- 使用中に不整合が発生する場合

使用中のシステムの整合性が失われると、コンソールウィンドウにエラーメッセージが表示されたり、システムがクラッシュしたりすることがあります。

`fsck` コマンドを使用する前に、655 ページの「`fsck` コマンドの構文とオプション」および『*Solaris* のシステム管理 (上級編)』の「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

▼ ファイルシステムのチェックが必要かを調べる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ファイルシステムがマウントされている場合、マウントを解除します。

```
# umount /mount-point
```

3. ファイルシステムをチェックします。

```
# fsck -m /dev/rdsk/device-name
```

指定したファイルシステムのスーパーブロック内の状態フラグがチェックされて、ファイルシステムがクリーンであるか、チェックする必要があるかどうか判断されます。

デバイス引数を省略すると、`/etc/vfstab` ファイル内で「`fsck pass`」の値が 0 より大きいすべての UFS ファイルシステムがチェックされます。

例 — ファイルシステムのチェックが必要かを調べる

次の例では、ファイルシステムのチェックが必要なことを示しています。

```
# fsck -m /dev/rdisk/c0t0d0s6
** /dev/rdisk/c0t0d0s6
ufs fsck: sanity check: /dev/rdisk/c0t0d0s6 needs checking
```

▼ ファイルシステムを対話式でチェックする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ルート (/) と /usr 以外のローカルファイルシステムをマウント解除します。

```
# umountall -l
```

3. ファイルシステムをチェックします。

```
# fsck
```

/etc/vfstab ファイル内で、「fsck pass」フィールドのエントリが 0 より大きいすべてのファイルシステムがチェックされます。また、fsck コマンドの引数として、マウントポイントディレクトリや /dev/rdisk/device-name も指定できます。整合性が失われている場合には、そのことを示すメッセージが表示されます。

1 つまたは複数の UFS ファイルシステムを対話式でチェックしながら、エラーメッセージのプロンプトに応答する方法については、『Solaris のシステム管理 (上級編)』の「UFS ファイルシステムの不整合解決 (手順)」を参照してください。



注意 – マウントされているファイルシステム上で fsck コマンドを実行すると、fsck コマンドによってなんらかの変更が行われた場合にシステムがクラッシュする可能性があります。ただし、シングルユーザーモードで fsck コマンドを実行してファイルシステムを修復する場合などは除きます。

4. エラーを修正し終わったら、**fsck** と入力して **Return** キーを押します。
fsck コマンドは、一度の実行ですべてのエラーを修正できないことがあります。「FILE SYSTEM STATE NOT SET TO OKAY」というメッセージが表示される場合は、fsck コマンドを使って、修正作業を繰り返します。fsck では修正できない場合は、652 ページの「fsck コマンドで修復できない UFS ファイルシステムの修正」を参照してください。
5. **lost+found** ディレクトリに保存されているファイルの名前を変更して移動します。
fsck コマンドによって lost+found ディレクトリに入れられた各ファイルの名前は、その i ノード番号を使用して変更されます。可能であれば、ファイル名を変更し、ファイルが含まれるディレクトリに移動してください。grep コマンドを使用して各ファイル中の語句を探したり、file コマンドを使用してファイルタイプ

を識別できる場合もあります。ディレクトリ全体が `lost+found` ディレクトリに書き込まれている場合の方が、復帰先のディレクトリを調べて、移動することは容易です。

例 — ファイルシステムを対話式でチェックする

次の例は、`/dev/rdisk/c0t0d0s6` ファイルシステムをチェックし、不正なブロック数を訂正する方法を示しています。

```
# fsck /dev/rdisk/c0t0d0s6
checkfileys: /dev/rdisk/c0t0d0s6
** Phase 1 - Check Block and Sizes
INCORRECT BLOCK COUNT I=2529 (6 should be 2)
CORRECT? y

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Cylinder Groups
929 files, 8928 used, 2851 free (75 frags, 347 blocks, 0.6%
fragmentation)
/dev/rdisk/c0t0d0s6 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
```

UFS ファイルシステムの修復

`fsck -o p` コマンド (`p` は修復用) は、UFS ファイルシステムをチェックし、通常は予期しないシステムのシャットダウンによって発生する問題を自動的に修正します。オペレータの介入が必要な問題が発見されると、このコマンドは即座に終了します。このコマンドによって、ファイルシステムを並列にチェックすることも可能です。

システムがクリーンな状態でシャットダウンしなかった後のファイルシステムの修復にも、`fsck -o p` コマンドを実行することができます。このモードでは、`fsck` コマンドはクリーンフラグを調べずに完全チェックを実行します。これらの処理は、`fsck` コマンドを対話式で実行した場合の処理のサブセットです。

▼ UFS ファイルシステムを修復する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. UFS ファイルシステムをマウント解除します。

```
# umount /mount-point
```

3. 修復オプションを指定して UFS ファイルシステムをチェックします。

```
# fsck -o p /dev/rdisk/device-name
```

fsck コマンドの引数として /mount-point または /dev/rdisk/device-name を使用すると、個々のファイルシステムを修復できます。

例 — UFS ファイルシステムを修復する

次の例は、/usr ファイルシステムを修復する方法を示しています。

```
# fsck -o p /usr
```

fsck コマンドで修復できない UFS ファイルシステムの修正

あるパスで問題が訂正されたために、前のパスで見つからなかった問題が表面化する場合があるので、fsck コマンドを何度か実行してファイルシステムを修正しなければならないことがあります。fsck コマンドはクリーンになるまで動作を続けるわけではないので、手作業で再実行しなければなりません。

fsck コマンドで表示される情報に注目してください。問題を解決する上で参考になります。たとえば、メッセージは損傷したディレクトリを指している場合があります。そのディレクトリを削除すると、fsck コマンドが問題なく実行されるようになる場合もあります。

それでも fsck コマンドでファイルシステムを修復できない場合は、fsdb、ff、clri、または ncheck コマンドを使用し、間違いを指定して修正します。これらのコマンドの使用方法については、fsdb(1M)、ff(1M)、clri(1M)、ncheck(1M) の各マニュアルページを参照してください。最終的には、ファイルシステムを作成し直し、その内容をバックアップメディアから復元せざるを得ない場合があります。

ファイルシステム全体を復元する方法については、第 49 章を参照してください。

ファイルシステムを完全に修復できないが、読み取り専用としてマウントできる場合は、cp、tar、または cpio コマンドを使用して、データのすべてまたは一部をファイルシステムから取り出してください。

問題の原因がハードウェア上のディスクエラーであれば、ファイルシステムを作成し直して復元する前に、ディスクをフォーマットし直して再びスライスに分割しなければならない場合があります。一般に、ハードウェアエラーが発生すると、さまざまなコマンドで同じエラーが繰り返し表示されます。format コマンドはディスク上の不良ブロックを使用しないようにします。ただし、ディスクの損傷が致命的な場合、フォーマットし直した後も問題が解決されないことがあります。format コマンドの使用方法については、format(1M) のマニュアルページを参照してください。新しいディスクのインストール方法については、第 34 章または第 35 章を参照してください。

不正なスーパーブロックの復元

ファイルシステムのスーパーブロック内のデータが破壊された場合は、復元しなければなりません。スーパーブロックが不正なときには、`fsck` コマンドからメッセージが表示されます。幸い、スーパーブロックのコピーがファイルシステム内に格納されています。`fsck -o b` コマンドを使用すると、スーパーブロックをそのいずれかのコピーで置き換えることができます。

スーパーブロックの詳細については、666 ページの「スーパーブロック」を参照してください。

ルート (/) ファイルシステム内のスーパーブロックが損傷し、修復できない場合は、次のどちらかの操作を実行します。

- システムをインストールし直す。
- ネットワークまたはローカルの CD からブートし、以下の手順を実行する。それらの手順がうまくいかない場合は、`newfs` コマンドを使ってルート (/) ファイルシステムを作成し直し、バックアップコピーから復元する。

▼ 不正なスーパーブロックを復元する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 不正なスーパーブロックがルート (/) または `/usr` ファイルシステム内にあるかどうかを調べ、次のどちらかの操作を実行します。
 - a. 不正なスーパーブロックがルート (/) または `/usr` ファイルシステム内にある場合は、システムをいったん停止し、ネットワークまたはローカル接続された CD からブートします。

ローカル接続された CD からブートする場合は、次のコマンドを使用します。

```
ok boot cdrom -s
```

ブートサーバーまたはインストールサーバーがすでに設定済みのネットワークからブートする場合は、次のコマンドを使用します。

```
ok boot net -s
```

システムを停止する方法については、198 ページの「SPARC: 復元を目的としてシステムを停止する方法」または 213 ページの「x86: 復元を目的としてシステムを停止する方法」を参照してください。

- b. 不正なスーパーブロックがルート (/) または `/usr` ファイルシステム内にはない場合は、損傷したファイルシステム以外のディレクトリに移動し、ファイルシステムをマウント解除します。

```
# umount /mount-point
```



注意 - 次の手順では、必ず `newfs -N` オプションを使用してください。-N オプションを省略すると、新しい空のファイルシステムが作成されます。

3. `newfs -N` コマンドを使用して、スーパーブロックの値を表示します。

```
# newfs -N /dev/rdisk/device-name
```

このコマンドの出力には、`newfs` コマンドによってファイルシステムが作成されたときに、スーパーブロックのコピーとして使用されることになったブロック番号が表示されます。カスタマイズされたファイルシステムを作成する方法については、669 ページの「カスタムファイルシステムパラメータ」を参照してください。

4. `fsck` コマンドを使用して、代替スーパーブロックを指定します。

```
# fsck -F ufs -o b=block-number /dev/rdisk/device-name
```

`fsck` コマンドは、指定された代替スーパーブロックを使用して、一次スーパーブロックを復元します。いつでも代替ブロックとして 32 を試すことができます。また、`newfs -N` コマンドで表示された代替ブロックを使用することもできます。

例 — 不正なスーパーブロックを復元する

次の例は、スーパーブロックのコピー 5264 を復元する方法を示しています。

```
# newfs -N /dev/rdisk/c0t3d0s7
/dev/rdisk/c0t3d0s7: 163944 sectors in 506 cylinders of 9 tracks, 36 sectors
83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
super-block backups (for fsck -b #) at:
 32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,
47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,
93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
140528, 145760, 150992, 156224, 161456,
# fsck -F ufs -o b=5264 /dev/rdisk/c0t3d0s7
Alternate superblock location: 5264.
** /dev/rdisk/c0t3d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)
/dev/rdisk/c0t3d0s7 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
#
```

fsck コマンドの構文とオプション

fsck コマンドは、ファイルシステム内の不整合をチェックして修復します。オプションを指定しないで fsck コマンドを実行した場合は、修復が行われる前に確認を求めるプロンプトが表示されます。このコマンドには、次の4つのオプションがあります。

コマンドとオプション	説明
fsck -m	ファイルシステムがマウント可能かどうかのチェックを行う
fsck -y	すべての修復に対して yes の応答が指定されたものとして処理を行う
fsck -n	すべての修復に対して no の応答が指定されたものとして処理を行う
fsck -o p	確認を求めるプロンプトを表示することなく、ファイルシステムを修復し、想定される(軽微な)不整合箇所をすべて修正するが、重大な問題が見つかったら終了する

第 44 章

UFS ファイルシステム (参照情報)

この章の内容は次のとおりです。

- 657 ページの「ルート (/) と /usr ファイルシステムのデフォルトディレクトリ」
- 665 ページの「プラットフォームに依存するディレクトリ」
- 666 ページの「UFS ファイルシステムのシリンダグループの構造」
- 669 ページの「カスタムファイルシステムパラメータ」
- 673 ページの「カスタマイズされたファイルシステムを作成するためのコマンド」

ルート (/) と /usr ファイルシステムのデフォルトディレクトリ

/kernel ディレクトリには、プラットフォームに依存しないオブジェクト (プラットフォームに依存しないカーネル `genunix` も含む) だけが入っています。プラットフォームに依存するディレクトリ `/platform` と `/usr/platform` については、表 44-3 を参照してください。

次の表に、ルート (/) ファイルシステムに含まれているデフォルトのディレクトリの説明を示します。

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ

ディレクトリ	説明
/	ファイルシステムの名前空間全体のルート
/dev	論理デバイスファイルの一次位置
/dev/cfg	物理 <code>ap_id</code> へのシンボリックリンク

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
/dev/cua	uucp 用のデバイスファイル
/dev/dsk	ブロックディスクデバイス
/dev/fbs	フレームバッファのデバイスファイル
/dev/fd	ファイル記述子
/dev/md	ボリューム管理デバイス名
/dev/printers	USB プリンタデバイスファイル
/dev/pts	pty スレーブデバイス
/dev/rdisk	raw ディスクデバイス
/dev/rmt	raw テープデバイス
/dev/sad	STREAMS Administrative Driver のエントリポイント
/dev/sound	オーディオデバイスとオーディオデバイス制御ファイル
/dev/swap	デフォルトのスワップデバイス
/dev/term	シリアルデバイス
/devices	物理デバイスファイル
/etc	ホスト固有のシステム管理構成ファイルとデータベース
/etc/acct	アカウントの構成情報
/etc/apache	Apache の構成ファイル
/etc/cron.d	cron の構成情報
/etc/default	各種プログラムのデフォルト情報
/etc/dfs	エクスポートされるファイルシステムの構成情報
/etc/dhcp	DHCP (Dynamic Host Configuration Protocol) の構成ファイル
/etc/dmi	Solstice Enterprise Agents の構成ファイル
/etc/fn	フェデレーテッドネーミングサービスと x.500 のサポートファイル
/etc/fs	ファイルシステムタイプ別に編成されたバイナリ
/etc/ftpd	ftpd の構成ファイル
/etc/gss	GSS (Generic Security Service) アプリケーションプログラミングインタフェースの構成ファイル

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
/etc/gtk	GNOME (GNU Network Object Model Environment) の構成ファイル
/etc/inet	インターネットサービスの構成ファイル
/etc/init.d	実行レベルを変更するためのスクリプト
/etc/iplanet	iPlanet (Sun ONE) の構成ファイル
/etc/krb5	Kerberos の構成ファイル
/etc/lib	/usr が利用できないときに必要な動的リンクライブラリ
/etc/l1c2	論理リンク制御 (l1c2) ドライバの構成ファイル
/etc/lp	プリンタサブシステムの構成情報
/etc/lu	Solaris Live Upgrade の構成ファイル
/etc/lvm	Solaris ボリュームマネージャの構成ファイル
/etc/mail	メールサブシステムの構成情報
/etc/nca	Solaris NCA (Network Cache and Accelerator) の構成ファイル
/etc/net	TI (トランスポート独立) ネットワークサービスの構成情報
/etc/nfs	NFS サーバーロギングの構成ファイル
/etc/openwin	OpenWindows の構成ファイル
/etc/opt	オプションパッケージの構成情報
/etc/ppp	Solaris PPP の構成ファイル
/etc/rc0.d	実行レベル 0 を開始または停止したときに起動されるスクリプト
/etc/rc1.d	実行レベル 1 を開始または停止したときに起動されるスクリプト
/etc/rc2.d	実行レベル 2 を開始または停止したときに起動されるスクリプト
/etc/rc3.d	実行レベル 3 を開始または停止したときに起動されるスクリプト
/etc/rcS.d	システムをシングルユーザーモードにするためのスクリプト
/etc/rcm	再構成マネージャ (RCM) のカスタムスクリプト用のディレクトリ
/etc/rpcsec	NIS+ 認証の構成ファイルが含まれていることがある

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
/etc/saf	サービスアクセス機能ファイル (FIFO など)
/etc/security	BSM (Basic Security Module) の構成ファイル
/etc/sfw	Samba の構成ファイル
/etc/skel	新規ユーザーアカウントのデフォルトプロファイルスクリプト
/etc/smartcard	Solaris SmartCards の構成ファイル
/etc/snmp	Solstice Enterprise Agents の構成ファイル
/etc/ssh	ssh (secure shell) の構成ファイル
/etc/sysevent	syseventd の構成ファイル
/etc/tm	商標ファイル。内容はブート時に表示される
/etc/usb	USB の構成情報
/etc/uucp	uucp 構成情報
/etc/wrsm	WRSM (WCI Remote Shared Memory) の構成情報
/export	共有ファイルシステム (ユーザーのホームディレクトリやクライアントファイルシステムなど) 用のデフォルトのディレクトリ
/home	スタンドアロンシステム上にあるユーザーのホームディレクトリ用のデフォルトのディレクトリまたはマウントポイント。AutoFS の動作中、このディレクトリには新しいエンタリを作成できない
/kernel	プラットフォームに依存しない読み込み可能なカーネルモジュールのディレクトリ。ブートプロセスの一部として必要。プラットフォームに依存しないコアカーネル /kernel/genunix の汎用部分を含む。/platform ディレクトリと /usr/platform ディレクトリの構造については、表 44-3 を参照
/mnt	ファイルシステムの一般的な一次マウントポイント
/opt	追加アプリケーションパッケージ用のデフォルトディレクトリまたはマウントポイント
/platform	サポートされているプラットフォームのファイル。詳細については、表 44-3 を参照
/proc	プロセス情報
/sbin	ブートプロセスと手作業によるシステム障害の回復に使用される重要な実行可能プログラム

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
/tmp	一次ファイル。内容はブートシーケンス中に消去される
/usr	/usr ファイルシステムのマウントポイント。詳細については、表 44-2 を参照
/var	常に変化するファイル (一時ファイル、ログファイル、状態ファイルなど) 用のディレクトリ
/var/adm	システムのログファイルとアカウントングファイル
/var/apache	Apache Web サーバー用のスクリプト、アイコン、ログ、キャッシュページ
/var/audit	BSM (Basic Security Module) の監査ファイル
/var/crash	カーネルクラッシュダンプのデフォルトの格納場所
/var/cron	cron のログファイル
/var/dmi	Solstice Enterprise Agents のデスクトップ管理インタフェースの実行時構成要素
/var/dt	dtlogin の構成ファイル
/var/inet	IPv6 ルーターの状態ファイル
/var/krb5	Kerberos のデータベースとログファイル
/var/ld	実行時リンカーの構成ファイル
/var/ldap	LDAP クライアントの構成ファイル
/var/log	システムログファイル
/var/lp	ラインプリンタサブシステムのログ情報
/var/mail	ユーザーのメールが保管されるディレクトリ
/var/news	コミュニティサービスメッセージ。これらのメッセージは USENET 方式のニュースとは異なる
/var/nfs	NFS サーバーのログファイル
/var/nis	NIS+ データベース
/var/ntp	NTP (Network Time Protocol) サーバーの状態ディレクトリ
/var/opt	ソフトウェアパッケージ関連の各種ファイルのサブツリーのルート
/var/preserve	vi と ex のバックアップファイル
/var/run	一時的な (つまり、システムをリブート後に残る必要がない) システムファイル。TMPFS マウントされたディレクトリ

表 44-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
/var/sadm	ソフトウェアパッケージ管理ユーティリティで管理されるデータベース
/var/saf	saf (サービスアクセス機能) のログファイルとアカウントファイル
/var/samba	Samba のログファイルとロックファイル
/var/snmp	SNMP の状態と構成情報
/var/spool	スプール化された一時ファイルのディレクトリ
/var/spool/clientmqueue	Sendmail のクライアントファイル
/var/spool/cron	cron と at のスプールファイル
/var/spool/locks	スプールロックファイル
/var/spool/lp	ラインプリンタのスプールファイル
/var/spool/mqueue	配信用に待ち行列に入れられたメール
/var/spool/pkg	スプール化されたパッケージ
/var/spool/print	LP 印刷サービスのクライアント側要求格納域
/var/spool/samba	Samba の印刷待ち行列
/var/spool/uucp	待ち行列に入っている uucp のジョブ
/var/spool/uucppublic	uucp によって格納されるファイル
/var/statmon	ネットワーク状態監視ファイル
/var/tmp	一時ファイルのディレクトリ。ブートシーケンス中には消去されない
/var/uucp	uucp のログファイルと状態ファイル
/var/yp	NIS データベース

次の表に、/usr ファイルシステムに含まれているデフォルトのディレクトリの説明を示します。

表 44-2 /usr ファイルシステムのデフォルトディレクトリ

ディレクトリ	説明
4lib	SunOS 4.1 バイナリ互換パッケージライブラリ
5bin	/usr/bin ディレクトリへのシンボリックリンク
x	/usr/openwin ディレクトリへのシンボリックリンク

表 44-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
adm	/var/adm ディレクトリへのシンボリックリンク
apache	Apache の実行可能プログラム、ロード可能モジュール、マニュアル
aset	ASET (Automated Security Enhancement Tools) のプログラムとファイル用のディレクトリ
bin	標準的なシステムコマンド用のディレクトリ
ccs	C 言語処理系のプログラムとライブラリ
demo	デモのプログラムとデータ
dict	/usr/share/lib/dict ディレクトリへのシンボリックリンク。UNIX の spell プログラムが使用する辞書が入っている
dt	CDE ソフトウェア用のディレクトリまたはマウントポイント
games	空のディレクトリ。SunOS 4.0-4.1 ソフトウェアで使用されていた
include	C プログラム用などのヘッダーファイル
iplanet	Directory Server の実行可能プログラム、ロード可能モジュール、マニュアル
j2se	Java 2 SDK の実行可能プログラム、ロード可能モジュール、マニュアル
java *	Java のプログラムとライブラリが入っているディレクトリ
kernel	その他のカーネルモジュール
kvm	廃止または互換性がなくなる可能性あり
lib	各種プログラムのライブラリ、アーキテクチャ依存データベース、またはユーザーが直接呼び出さないバイナリ
local	サイトのローカルコマンド
mail	/var/mail ディレクトリへのシンボリックリンク
man	/usr/share/man ディレクトリへのシンボリックリンク
net	ネットワークリスナーサービス用のディレクトリ
news	/var/news ディレクトリへのシンボリックリンク
oasys	FMLI (Form and Menu Language Interpreter) 実行環境用のファイル
old	段階的に使用されなくなっているプログラム

表 44-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
openwin	OpenWindows ソフトウェアのディレクトリまたはマウントポイント
perl5	perl 5 のプログラムとマニュアル
platform	サポートされているプラットフォームのファイル。詳細については、表 44-3 を参照
preserve	/var/preserve ディレクトリへのシンボリックリンク
proc	proc ツール用のディレクトリ
pub	オンラインマニュアルページと文字処理用のファイル
sadm	システム管理に関連する各種ファイルとディレクトリ
sbin	システム管理用の実行可能プログラム
sbin/install.d	JumpStart のカスタムのスクリプトと実行可能プログラム
sbin/static	/usr/bin と /usr/sbin から選択したプログラムの静的リンクバージョン
sbin/sparc7 と sparc9	コマンドの 32 ビットバージョンと 64 ビットバージョン
sfw	GNU と公開されているソースの実行可能プログラム、ライブラリ、マニュアル
share	アーキテクチャに依存しない共有可能ファイル
share/admserv5.1	iPlanet Console and Administration Server 5.0 のマニュアル
share/audio	オーディオファイルのサンプル
share/ds5	Sun ONE Directory Server 5.1 のマニュアル
share/lib	アーキテクチャに依存しないデータベース
share/man	Solaris のマニュアルページ
share/src	カーネル、ライブラリ、ユーティリティのソースコード
snadm	システム管理とネットワーク管理に関するプログラムとライブラリ
spool	/var/spool ディレクトリへのシンボリックリンク
src	/usr/share/src ディレクトリへのシンボリックリンク
tmp	/usr/var/tmp ディレクトリへのシンボリックリンク
ucb	UCB 互換パッケージのバイナリ
ucbinclude	UCB 互換パッケージのヘッダーファイル
ucblib	UCB 互換パッケージのライブラリ

表 44-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

ディレクトリ	説明
vmsys	FACE (Framed Access Command Environment) プログラム用のディレクトリ
xpg4	POSIX 準拠ユーティリティ用のディレクトリ

プラットフォームに依存するディレクトリ

表 44-3 に、/platform ディレクトリと /usr/platform ディレクトリに入っているすべてのプラットフォームに依存するオブジェクトを示します。

表 44-3 /platform と /usr/platform ディレクトリ

ディレクトリ	説明
/platform	ルート (/) ファイルシステムに存在すべき一連のディレクトリが、サポートされるプラットフォームごとに 1 ディレクトリずつ入っている。
/platform/*/kernel	プラットフォームに依存するカーネル構成要素が入っている。プラットフォームに依存するコアカーネルであるファイル <code>unix</code> も含む。詳細については、 <code>kernel (1M)</code> のマニュアルページを参照。
/usr/platform	ルート (/) ファイルシステムに存在する必要がない、プラットフォームに依存するオブジェクトが入っている。
/usr/platform/*/lib	/usr/lib ディレクトリ中のオブジェクトに類似した、プラットフォームに依存するオブジェクトが入っている。
/usr/platform/*/sbin	/usr/sbin ディレクトリ中のオブジェクトに類似した、プラットフォームに依存するオブジェクトが入っている。

UFS ファイルシステムのシリンダグループの構造

UFS ファイルシステムを作成すると、ディスクスライスは、1 つまたは複数の連続するディスクシリンダから構成される「シリンダグループ」に分割されます。シリンダグループはさらにアドレス指定可能なブロックに分割され、このブロックによって、シリンダグループ内のファイルの構造が編成され、制御されます。各種のブロックは、ファイルシステム内で特定の機能を持っています。UFS ファイルシステムには、次の 4 種類のブロックがあります。

ブロックの種類	格納されている情報の種類
ブートブロック	システムのブート時に使用される情報
スーパーブロック	ファイルシステムに関する大部分の情報
i ノード	ファイルに関する名前以外のすべての情報
記憶域またはデータブロック	各ファイルのデータ

次の節では、これらのブロックの編成と機能について説明します。

ブートブロック

ブートブロックには、システムのブート時に使用されるオブジェクトが格納されます。ファイルシステムがブートに使用されなければ、ブートブロックは空白のままです。ブートブロックは最初のシリンダグループ (シリンダグループ 0) にのみ表示され、スライス内の最初の 8K バイトです。

スーパーブロック

スーパーブロックには、次のようなファイルシステムに関する大部分の情報が格納されます。

- ファイルシステムのサイズと状態
- ラベル。ファイルシステム名とボリューム名を含む
- ファイルシステムの論理ブロックのサイズ
- 最終更新日時
- シリンダグループのサイズ
- シリンダグループ内のデータブロック数
- 集計データブロック

- ファイルシステムの状態
- 最後のマウントポイントのパス名

スーパーブロックには重要なデータが入っているので、ファイルシステムの作成時には複数のスーパーブロックが作成されます。

集計情報ブロックは、スーパーブロック内に保管されます。複製されませんが、通常はシリンダグループ 0 内で最初のスーパーブロックといっしょにグループ化されます。集計ブロックには、ファイルシステムの使用時に発生した変化が記録されます。さらに、ファイルシステム内の i ノード数、ディレクトリ数、フラグメント数、および記憶ブロック数が表示されます。

i ノード

i ノードには、ファイルに関して名前以外のすべての情報が入っており、ディレクトリ内に保管されます。i ノードは 128 バイトです。i ノード情報はシリンダ情報ブロック内に保管され、次の情報が入っています。

- ファイルのタイプ
 - 通常ファイル
 - ディレクトリ
 - ブロック型特殊ファイル
 - キャラクタ型特殊ファイル
 - FIFO (名前付きパイプとも呼ぶ)
 - シンボリックリンク
 - ソケット
 - その他の i ノード - 属性ディレクトリとシャドウファイル (ACL 用)
- ファイルのモード (読み取り権 - 書き込み権 - 実行権のセット)
- ファイルへのハードリンク数
- ファイルの所有者のユーザー ID
- ファイルが属するグループ ID
- ファイル内のバイト数
- 15 個のディスクブロックアドレスの配列
- ファイルの最終アクセス日時
- ファイルの最終変更日時
- ファイルの作成日時

15 個のディスクアドレス (0 から 14 まで) の配列は、ファイルの内容が格納されるデータブロックを指します。最初の 12 個は直接アドレスです。つまり、ファイルの内容のうち最初の 12 個の論理記憶ブロックを直接指します。ファイルが論理ブロック 12 個分より大きい場合は、13 番目のアドレスは間接ブロックを指します。間接ブロックには、ファイルの内容ではなく直接ブロックのアドレスが入っています。

14 番目のアドレスは、二重間接ブロックを指します。二重間接ブロックには、間接ブロックのアドレスが入っています。15 番目のアドレスは三重間接アドレス用です。次の図に、i ノードから始まるこのアドレスブロックチェーンを示します。

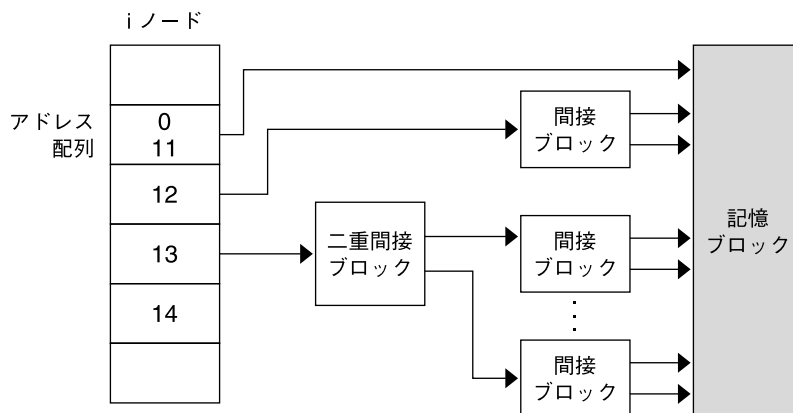


図 44-1 UFS ファイルシステムのアドレスチェーン

データブロック

ファイルシステムに割り当てられた残りの領域には、データブロック (記憶ブロックともいう) が入っています。これらのデータブロックのサイズは、ファイルシステムの作成時に決定されます。デフォルトでは、データブロックは2つのサイズ、つまり8K バイトの論理ブロックサイズと1K バイトのフラグメントサイズで割り当てられます。

通常ファイルの場合、データブロックにはファイルの内容が入っています。ディレクトリの場合、データブロックにはディレクトリ内のファイルの i ノード番号とファイル名を示すエントリが入っています。

空きブロック

現在、i ノード、間接アドレスブロック、または記憶ブロックとして使用されていないブロックには、シリンダグループマップ内で空きを示すマークが付けられます。また、このマップはフラグメントを追跡し、断片化によるディスクパフォーマンスの低下を防止します。

UFS ファイルシステムの内容の概念を理解しやすいように、次の図に、一般的な UFS システム内の一連のシリンダグループを示します。

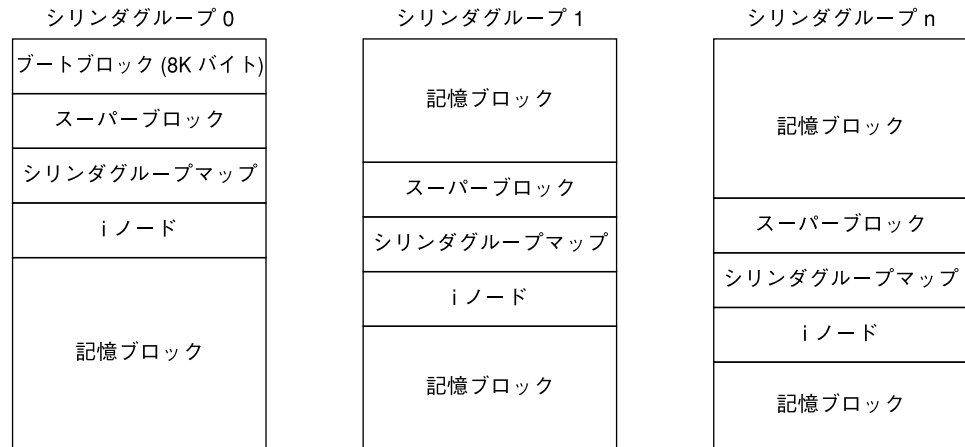


図 44-2 一般的な UFS ファイルシステム

カスタムファイルシステムパラメータ

`newfs` コマンドによって割り当てられるデフォルトのファイルシステムパラメータを変更しようとする前に、各パラメータについて理解しておく必要があります。この節では、次の各パラメータについて説明します。

- 論理ブロックサイズ
- フラグメントサイズ
- 最小空き領域
- 回転待ち
- 最適化タイプ
- i ノード数 (ファイルの数)

論理ブロックサイズ

論理ブロックサイズは、UNIX カーネルがファイルの読み書きに使用するブロックのサイズです。一般に、論理ブロックサイズは物理ブロックサイズとは異なります。物理ブロックサイズは、通常は 512K バイトで、ディスクコントローラが読み書きできる最小ブロックのサイズです。

論理ブロックサイズは、デフォルトでシステムのページサイズに設定されます。UFS ファイルシステムの場合、このデフォルト論理ブロックサイズは 8192 バイト (8K バイト) です。UFS ファイルシステムでは、ブロックサイズとして 4096 バイトまたは 8192 バイト (4K または 8K バイト) がサポートされます。論理ブロックの推奨サイズは 8K バイトです。

SPARC のみ - sun4u プラットフォームで指定できるブロックサイズは 8192 バイトだけです。

システムに最善の論理ブロックサイズを選択するには、必要なパフォーマンスと使用可能容量を検討してください。ほとんどの UFS システムでは、8K バイトのファイルシステムが最高のパフォーマンスを発揮し、ディスクパフォーマンスと一次メモリーやディスク上の領域の使用量が適切なバランスに保たれます。

原則として、効率を高めるには、ほとんどのファイルがきわめて大きいファイルシステムには大きめの論理ブロックサイズを使用します。ほとんどのファイルがきわめて小さいファイルシステムには、小さめの論理ブロックサイズを使用します。ファイルシステム上で `quot -c file-system` コマンドを使用すると、ファイルの分散に関する詳細なレポートをブロックサイズ別に表示できます。

ただし、通常は、ファイルシステムの作成時に設定されたページサイズが最適です。

フラグメントサイズ

ファイルが作成または拡張されると、論理ブロック全体または「フラグメント」と呼ばれる部分のディスク容量が割り当てられます。ファイルのためにディスク容量が必要になると、まずブロック全体が割り当てられ、次に残りの部分にブロックのうち 1 つまたは複数のフラグメントが割り当てられます。小型ファイルの場合、割り当てはフラグメントから始まります。

ブロック全体ではなく、そのフラグメントを割り当てることができるので、ブロック内の未使用のホールによって生じるディスク容量の「断片化」が減少し、容量の節約になります。

UFS ファイルシステムを作成するときに、「フラグメントサイズ」を定義します。デフォルトのフラグメントサイズは 1K バイトです。各ブロックは、1 個、2 個、4 個、または 8 個のフラグメントに分割できます。この場合、フラグメントサイズは 8192 バイトから 512 バイトまでです (4K バイトのファイルシステムのみ)。実際には、下限はディスクのセクターサイズ、通常は 512 バイトに連動します。

マルチテラバイトのファイルシステムの場合、フラグメントサイズはファイルシステムのブロックサイズに等しくなります。

注- フラグメントサイズの上限は論理ブロックサイズに等しくなります。この場合、フラグメントは存在しないこととなります。容量よりも速度を重視する場合、きわめて大型のファイルがあるファイルシステムには、この構成が最適なことがあります。

フラグメントサイズを選択するときには、処理時間と容量を取捨選択してください。フラグメントサイズが小さければ容量の節約になりますが、割り当てには時間がかかります。原則として、格納効率を高めるには、ほとんどのファイルが大型のファイルシステムには、大きめのフラグメントサイズを使用します。ほとんどのファイルが小型のファイルシステムには、小さめのフラグメントサイズを使用します。

最小空き容量

「最小空き容量」とは、ファイルシステムの作成時に予約分として保持されるディスク容量の割合です。デフォルトの予約分は、 $((64\text{M バイト}/\text{パーティションサイズ}) * 100)$ で算出し、その値は最も近い整数に切り捨てられ、ディスク容量の 1% から 10% の範囲に制限されます。

ファイルシステム内の空き容量が少なくなるほど、アクセス速度が低下するので、空き容量は重要です。十分な空き容量があれば、UFS ファイルシステムは効率よく動作します。ファイルシステムがいっぱいになって、使用可能なユーザー領域を使い果たすと、スーパーユーザー以外は予約済みの空き容量にアクセスできなくなります。

`df` などのコマンドは、最小空き容量として割り当て済みの分を差し引いて、ユーザーに使用可能な容量をパーセントで表示します。コマンドでファイルシステム内のディスク容量の 100 パーセント以上が使用中であると表示される場合は、予約分の一部がルートに使用されています。

ユーザーに制限を適用する場合に、各ユーザーが使用可能な容量には予約分の空き容量は含まれません。`tunefs` コマンドを使用すると、既存のファイルシステムの最小空き容量の値を変更できます。

回転待ち

最新のディスクでは、このパラメータをもう使用しません。使用する必要がある場合は、ファイルシステムの作成時に指定されたデフォルト値を使用することをお勧めします。

最適化のタイプ

「最適化のタイプ」パラメータの設定には、「*space*」と「*time*」があります。

- *space* – 最適化のタイプに *space* を選択すると、断片化を最小限度に抑え、ディスクの使用状況が最適化されるようにディスクブロックが割り当てられる。
- *time* – 最適化のタイプに *time* を選択すると、配置はあまり重視されず、できるだけ高速になるようにディスクブロックが割り当てられる。十分な空き領域があれば、それほど細かく断片化しなくても、比較的簡単にディスクブロックを効率よく割り当てることができる。デフォルトは *time*。

既存のファイルシステムの場合は、`tunefs` コマンドを使用して最適化タイプのパラメータ値を変更できる。

詳細については、`tunefs (1M)` のマニュアルページを参照してください。

i ノード数 (ファイルの数)

「i ノード数」パラメータによって、ファイルシステム内で保持できるファイル数が決まります。ファイルごとに i ノードが 1 つあります。「i ノード 1 個あたりのバイト数」によって、ファイルシステムの作成時に作成される i ノードの総数が決まります。これは、ファイルシステムの合計サイズを、i ノード 1 個あたりのバイト数で割った値です。i ノードが割り当てられたら、ファイルシステムを作成し直さないかぎり、その数は変更できません。

i ノード 1 個あたりのデフォルトのバイト数は 2048 バイト (2K バイト) で、これは各ファイルの平均サイズが 2K バイト以上であることを想定しています。ファイルシステムが 1G バイトを超える場合、次の公式が使用されます。

ファイルシステムのサイズ	i ノードごとのバイト数
1G バイト以下	2048
2G バイト未満	4096
3G バイト未満	6144
3G バイト以上 1T バイト以下	8192
1T バイト超	1048576

多数のシンボリックリンクを持つファイルシステムでは、平均ファイルサイズを小さくすることができます。ファイルシステムに多数の小型ファイルが格納される場合は、このパラメータに小さい値を与えてもかまいません。ただし、i ノード数が少ないために i ノードが不足するよりも、多すぎる方が好ましいことを留意してください。i ノード数が少なすぎると、実際には空のディスクスライス上でも最大ファイル数に達してしまうことがあります。

UFS ファイルとファイルシステムの最大サイズ

UFS ファイルシステムの最大サイズは、およそ 16T バイトで、使用できる領域は、そこから約 1 パーセントのオーバーヘッドを引いたサイズです。「空白」ファイルの論理サイズは 1T バイトです。ただし、ファイルに格納できる実際のデータ容量は、1T バイトから約 1 パーセントのファイルシステムオーバーヘッドを引いたサイズです。

UFS サブディレクトリの最大数

UFS ファイルシステム内の 1 ディレクトリあたりのサブディレクトリの最大数は 32,767 です。この制限はあらかじめ定義されたものであり、変更できません。

カスタマイズされたファイルシステムを作成するためのコマンド

この節では、カスタマイズされたファイルシステムの作成に使用する次の 2 つのコマンドについて説明します。

- `newfs`
- `mkfs`

`newfs` コマンドの構文、オプション、引数

`newfs` コマンドは、ファイルシステムの作成に使用する `mkfs` コマンドの簡便バージョンです。

構文は次のとおりです。

```
/usr/sbin/newfs [-Nv] [mkfs_options] raw_device
```

表 44-4 に、`newfs` コマンドのオプションと引数を示します。

表 44-4 newfs コマンドのオプションと引数

オプション	説明
-N	ファイルシステムの作成に使用されるファイルシステムパラメータが表示されるが、実際には作成されない。このオプションでは、既存のファイルシステムの作成に使用されたパラメータは表示されない。
-T	ファイルシステムのサイズが最終的に 1T バイトを超えることが可能となるように、ファイルシステムのパラメータが設定される。このオプションを指定すると、 <i>fragsize</i> が <i>bsize</i> と同じ値に設定され、 <i>-i</i> オプションによる指定がなければ、 <i>nbpi</i> は 1M バイトに設定される。 <i>-f</i> オプションまたは <i>-i</i> オプションを使って指定された <i>fragsize</i> 値または <i>nbpi</i> 値がこのオプションの値と矛盾した場合、指定された <i>fragsize</i> 値または <i>nbpi</i> 値は無視される。
-v	mkfs コマンドに渡されるパラメータが表示される。
<i>mkfs-options</i>	後続のオプション (<i>-s size</i> から <i>-C maxcontig</i> まで) を使用して mkfs コマンドのパラメータが設定される。それらのオプションは、mkfs コマンドに渡される順番に記述されている。各オプションは、空白で区切る。
<i>-s size</i>	ファイルシステムのセクター数。デフォルトは、ディスクラベルから自動的に判別される。
<i>-t ntrack</i>	ディスク上の 1 シリンダあたりのトラック数。デフォルトはディスクラベルから判別される。
<i>-b bsize</i>	データ転送に使用される論理ブロックのバイト数。サイズとして 4096 または 8192 バイト (4K または 8K バイト) を指定する。デフォルトは 8192 バイト (8K バイト)。
<i>-f fragsize</i>	ファイルに割り当てられるディスク容量の最小バイト数。フラグメントサイズを、512 バイトから 8192 バイトまでの 2 の乗数単位で指定する。デフォルトは 1024 バイト (1K バイト)。 1T バイトを超えるファイルシステム、または <i>-T</i> オプションを指定して作成されたファイルシステムの場合、 <i>fragsize</i> は強制的にブロックサイズ (<i>bsize</i>) と同じ値に設定される。
<i>-c cgsize</i>	1 シリンダグループあたりのディスクシリンダ数。デフォルト値を計算するには、ファイルシステム内のセクター数を 1G バイト内のセクター数で割り、その結果に 32 を掛ける。デフォルト値の範囲は 16 から 256 まで。
<i>-m free</i>	空きディスク領域の最小許容率。デフォルトの予約分は、 $((64M \text{ バイト} / \text{パーティションサイズ}) * 100)$ で算出した値は最も近い整数に切り捨てられ、ディスク容量の 1% から 10% の範囲に制限される。

表 44-4 newfs コマンドのオプションと引数 (続き)

オプション	説明
-r rpm	1 分当たりのディスクの回転数。この設定はドライバまたはデバイスに固有である。ドライブが回転数を報告できる場合、mkfs コマンドは報告された値を使用する。そうでない場合、デフォルトは 3600。このパラメータは、mkfs コマンドに渡される前に 1 秒当たりの回転数に変換される。
-i nbpi	作成できる i ノードの計算に使用される i ノード 1 個当たりのバイト数。デフォルト値については、672 ページの「i ノード数 (ファイルの数)」を参照。
-o opt	ディスクブロックをファイルに割り当てるときに使用される最適化のタイプ。opt には time または space を指定する。デフォルトは time。
-a apc	不良ブロックを配置するために予約される 1 ディスクシリンダ (SCSI デバイスのみ) の代替ブロック数。デフォルトは 0 (ゼロ)。
-d gap	(回転待ち) CPU がデータ転送を完了し、同じディスクシリンダ上で次のデータ転送を開始するまでにかかる予想最小時間 (ミリ秒)。デフォルトは 0 (ディスクドライブのトラック先読みバッファリングを考慮)。
-n nrpos	シリンダグループを分割するさまざまな回転位置の数。デフォルトは 8。
-C maxcontig	あるファイルに属し、回転待ちが挿入される前に連続して割り当てられる最大ブロック数。デフォルトはドライブごとに異なる。内部 (トラック) バッファを持たないドライブ (または、内部バッファが存在することを示していないドライブまたはコントローラ) の場合、デフォルトは 1。バッファを持つドライブの場合、デフォルトは 7。 このパラメータは、次のようにする必要がある。 $blocksize \times maxcontig \leq maxphys$ でなければならない <i>maxphys</i> は、入出力サブシステムが満たせる最大ブロック転送サイズ (バイト数) を指定する読み取り専用のカーネル変数である。この制限は、newfs や mkfs コマンドではなく mount コマンドによって適用される。 また、このパラメータはクラスタ化も制御する。 <i>rotdelay</i> の値に関係なく、 <i>maxcontig</i> が 1 より大きいときのみクラスタ化できる。クラスタ化すると、入出力が高速になる。詳細については、tunefs (1M) のマニュアルページを参照。
raw_device	ファイルシステムを入れるパーティションの特殊文字 (raw) デバイスファイル名。この引数は必須。

例 — newfs コマンドのオプションと引数

次の例は、`-N` オプションを使用して、バックアップスーパーブロックなどのファイルシステム情報を表示する方法を示しています。

```
# newfs -N /dev/rdisk/c0t0d0s0
/dev/rdisk/c0t0d0s0: 37260 sectors in 115 cylinders of 9 tracks, 36 sectors
                    19.1MB in 8 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
superblock backups (for fsck -b #) at:
    32, 5264, 10496, 15728, 20960, 26192, 31424, 36656,
#
```

汎用 mkfs コマンド

汎用 `mkfs` コマンドは、ファイルシステム専用の `mkfs` コマンドを呼び出して、指定したディスクスライス上で指定したタイプのファイルシステムを作成させます。`mkfs` コマンドは各種のファイルシステムに対応していますが、実際には UFS、UDFS、PCFS の各ファイルシステムの作成に使用します。他のタイプのファイルシステムを作成するには、ファイルシステム専用の `mkfs` コマンドを使用するためのソフトウェアを作成する必要があります。通常、`mkfs` コマンドは直接実行しません。`mkfs` コマンドは、`newfs` コマンドによって呼び出されます。

汎用 `mkfs` コマンドは、`/usr/sbin` ディレクトリに入っています。引数とオプションについては、`mkfs(1M)` のマニュアルページを参照してください。

第 45 章

ファイルとファイルシステムのバックアップおよび復元

以下の各章で、ファイルとファイルシステムのバックアップおよび復元について説明します。

第 46 章	ファイルとファイルシステムのバックアップに関するガイドライン、および計画に役立つ情報を提供します。
第 47 章	個々のファイルとファイルシステム全体をローカルデバイスまたはリモートデバイスからバックアップする方法を説明します。
第 48 章	UFS ファイルシステムのスナップショットを作成する手順を説明します。
第 49 章	個々のファイルとファイルシステム全体を復元する手順を説明します。
第 50 章	ufsdump コマンドの機能、ufsdump コマンドとufsrestore コマンドの構文およびオプションについて説明します。
第 51 章	さまざまなバックアップメディアで dd、pax、cpio、tar コマンドを使用する手順、およびファイルを異なるヘッダー形式でコピーする手順を説明します。
第 52 章	テープデバイスを追加する方法、およびテープデバイスのタイプやバックアップデバイス名を判断して磁気テープカートリッジを操作する手順を説明します。

第 46 章

ファイルシステムのバックアップと復元 (概要)

この章では、`ufsdump` コマンドと `ufsrestore` コマンドを使用して、ファイルシステムのバックアップおよび復元を実行する際のガイドラインと計画の作成について説明します。

この章の内容は次のとおりです。

- 679 ページの「ファイルシステムのバックアップと復元の新機能」
- 680 ページの「バックアップと復元についての参照先」
- 680 ページの「ファイルシステムのバックアップと復元とは」
- 681 ページの「ファイルシステムをバックアップする理由」
- 682 ページの「バックアップを作成するファイルシステムの計画」
- 684 ページの「バックアップタイプの選択」
- 684 ページの「テープデバイスの選択」
- 685 ページの「ファイルシステムのバックアップおよび復元の概要 (作業マップ)」
- 686 ページの「バックアップスケジュールを設定する際のガイドライン」
- 688 ページの「バックアップスケジュールの例」

ファイルシステムのバックアップと復元の新機能

この節では、Solaris 9 リリースで提供されるバックアップと復元の新機能について説明します。

UFS スナップショット

Solaris 9 リリースには、ファイルシステムのマウント中にファイルシステムのバックアップを作成する新規コマンド `fssnap` が追加されました。

`fssnap` コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。「スナップショット」は、バックアップ操作のためのファイルシステムの一時的イメージです。

詳細については、第 48 章を参照してください。

バックアップと復元についての参照先

バックアップまたは復元処理	参照先
<code>ufsdump</code> コマンドを使用したファイルシステムのバックアップ	第 47 章
<code>fssnap</code> コマンドを使用した UFS スナップショットの作成	第 48 章
<code>ufsrestore</code> コマンドを使用したファイルシステムの復元	第 49 章
<code>cpio</code> 、 <code>dd</code> 、 <code>pax</code> 、および <code>cpio</code> コマンドを使用したファイルおよびディレクトリのコピー	第 51 章

ファイルシステムのバックアップと復元とは

ファイルシステムのバックアップとは、消失、損傷、または破損に備えて、ファイルシステムをテープなどのリムーバブルメディアにコピーすることを意味します。ファイルシステムの復元とは、最新のバックアップファイルをリムーバブルメディアから作業ディレクトリにコピーすることを意味します。

この章では、UFS ファイルシステムのバックアップおよび復元に使用する `ufsdump` および `ufsrestore` コマンドについて説明します。他のコマンドを使用してファイルやファイルシステムをコピーし、ファイルの共有や移動を行うこともできます。次の表に、個々のファイルやファイルシステムをメディアにコピーする全コマンドを示します。

表 46-1 ファイルとファイルシステムのバックアップおよび復元用コマンド

作業	コマンド	参照先
1 つまたは複数のファイルシステムをローカルまたはリモートのテープデバイスにバックアップする	ufsdump	第 47 章または第 50 章
ファイルシステムの読み取り専用コピーを作成する	fssnap	第 48 章
ネットワーク上のシステムのファイルシステム全体をバックアップサーバーからバックアップする	Solstice Backup™ ソフトウェア	『Solstice Backup 5.1 管理者ガイド』
NIS+ マスターサーバーをバックアップ、復元する	nisbackup および nisrestore コマンド	『Solaris のシステム管理 (ネーミングとディレクトリサービス: FNS、NIS+ 編)』
テープまたはフロッピーディスク上でファイルをコピー、表示、検索する	tar、cpio、または pax コマンド	第 51 章
マスターディスクをクローンディスクにコピーする	dd	第 51 章
ファイルシステム全体または個々のファイルを、リムーバブルメディアから作業ディレクトリに復元する	ufsrestore コマンド	第 49 章

ファイルシステムをバックアップする理由

ファイルのバックアップは、最も重要なシステム管理作業の 1 つです。次のような原因によるデータの消失に備えて、定期的にバックアップを実行する必要があります。

- システムのクラッシュ
- 不注意によるファイルの削除
- ハードウェア障害
- 天災 (火災、台風、地震など)
- システムの再インストールやアップグレード時に発生する問題

バックアップを作成するファイルシステムの計画

頻繁に更新されるファイルシステムなど、ユーザーにとって重要なファイルシステムはバックアップしておく必要があります。次の表に、スタンドアロンシステムとサーバー用にバックアップを作成するファイルシステムの一般的なガイドラインを示します。

表 46-2 ファイルシステムをスタンドアロン用にバックアップする

バックアップするファイルシステム	説明	バックアップ間隔
ルート (/) - スライス 0	このファイルシステムにはカーネルが格納されているが、/var ディレクトリが格納されていることもある。/var ディレクトリには、メールやアカウントファイルなど、頻繁に変更されるファイルが保管される。	定期的 (毎週、毎日など)
/usr - スライス 6、/opt	一般に、新しいソフトウェアをインストールして新しいコマンドを追加すると、/usr と /opt ファイルシステムが影響を受ける。/opt ディレクトリは、ルート (/) の一部であるか、独自のファイルシステムである。	随時
/export/home - スライス 7	このファイルシステムには、スタンドアロンシステム上のユーザー全員のディレクトリおよびサブディレクトリが含まれる。	ルート (/) や /usr よりも頻繁に、サイトのニーズによっては毎日
/export、/var、または他のファイルシステム	Solaris ソフトウェアのインストール中に、これらのファイルシステムを作成した可能性がある。	サイトの必要に応じて

表 46-3 ファイルシステムをサーバー用にバックアップする

バックアップするファイルシステム	説明	バックアップ間隔
ルート (/) - スライス 0	このファイルシステムには、カーネルおよび実行可能プログラムが含まれる。	<p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>ネットワーク上でユーザーとシステムの追加および削除を頻繁に実行する場合、このファイルシステム内の構成ファイルを変更する必要がある。この場合、週に 1 度から月に 1 度の間隔で、ルート (/) ファイルシステムの完全バックアップを実行する必要がある。サイトでユーザーのメールをメールサーバー上の /var/mail ディレクトリに保管しておき、後でクライアントシステムがマウントする場合は、ルート (/) ディレクトリを毎日バックアップした方がよい。別のファイルシステムの場合は、/var を毎日バックアップした方がよい。</p>
/export - スライス 3	このファイルシステムには、ディスクレスクライアント用のカーネルおよび実行可能プログラムが含まれる。	<p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>このファイルシステム内の情報はスライス 0 にあるサーバーのルートディレクトリと同様なので、ファイルシステムは頻繁に変化することはない。サイトからメールをクライアントシステムに送信していない場合、このファイルシステムのバックアップは随時実行すればよい。送信している場合は、/export のバックアップをさらに頻繁に実行する必要がある。</p>
/usr - スライス 6、 /opt		<p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>通常、これらのファイルシステムは静的であるため、週または月に 1 度バックアップするだけでかまわない。</p>
/export/home - スライス 7	このファイルシステムには、システムの全ユーザーのディレクトリが含まれる。このファイルシステム内のファイルは、変更が多い。	毎日または毎週。

バックアップタイプの選択

ufsdump コマンドを使用すると、フルバックアップまたは増分バックアップを実行できます。fssnap コマンドを使用すると、ファイルシステムの一時イメージを作成できます。次の表に、各バックアップタイプの手順の違いを示します。

表 46-4 バックアップタイプの相違点

バックアップのタイプ	結果	長所	短所
完全 (フル)	ファイルシステムやディレクトリ全体をコピーする。	すべてのデータが1カ所に格納される。	大量のバックアップテープが必要であり、書き込みに時間がかかる。ドライブはテープ上でファイルが入っている位置に順番に移動しなければならないので、個々のファイルの検索時間が長くなる。複数のテープを検索しなければならない場合もある。
スナップショット	ファイルシステムの一時イメージを作成する。	システムはマルチユーザモードで動作可能。	スナップショットの作成中は、システムのパフォーマンスが低下する可能性がある。
増分	指定されたファイルシステム内で、前回のバックアップ以降に変更されたファイルのみをコピーする。	ファイルシステム内の小さな変化を簡単に検索できる。	どの増分テープにファイルが入っているかを探すのに時間がかかることがある。最後の完全ダンプに戻らなければならない場合もある。

テープデバイスの選択

次の表に、バックアップ処理中にファイルシステムを格納するための典型的なテープデバイスを示します。容量は、ドライブのタイプとテープに書き込むデータによって異なります。テープデバイスの詳細については、第 52 章を参照してください。

表 46-5 ファイルシステムのバックアップに使用する一般的なメディア

メディア	容量
1/2 インチのリールテープ	140M バイト (6250bpi)
2.5G バイト、1/4 インチのカートリッジ (QIC) テープ	2.5G バイト
DDS3 4mm カートリッジテープ (DAT)	12 - 24G バイト
14G バイト、8mm カートリッジテープ	14G バイト
DLT 7000 1/2 インチ カートリッジテープ	35 - 70G バイト

ファイルシステムのバックアップおよび復元の概要 (作業マップ)

この作業マップを使用して、ファイルシステムのバックアップおよび復元作業をすべて確認できます。各作業は、実行するバックアップタイプの判別などの一連の追加作業を指し示します。

作業	説明	参照先
1. バックアップするファイルシステムの識別	毎日、毎週、または毎月、バックアップの必要なファイルシステムを識別します。	682 ページの「バックアップを作成するファイルシステムの計画」
2. バックアップタイプの決定	サイトに適したファイルシステムのバックアップタイプを決定します。	684 ページの「バックアップタイプの選択」
3. バックアップの作成	次のいずれかを実行します。 ファイルシステムのフルバックアップおよび増分バックアップを実行する場合は、 <code>ufsdump</code> コマンドを使用します。 アクティブかつマウント中のファイルシステムのスナップショットを作成する場合は、 <code>fssnap</code> コマンドの使用を考慮してください。	第 47 章 第 48 章

作業	説明	参照先
	個人のホームディレクトリや、小規模で重要度の低いファイルシステムのフルバックアップを作成する場合は、tar、cpio、または pax コマンドを使用してください。	第 51 章
4. ファイルシステムの復元	(省略可能) ファイルまたはファイルシステムのバックアップに使用するコマンドに基づいて、復元方法を選択します。	
	ufsdump コマンドを使用して作成したファイルシステムのバックアップを復元します。	第 49 章
	tar、cpio、または pax コマンドを使用して作成したファイルシステムを復元します。	第 51 章
5. ルート (/) または /usr ファイルシステムの復元	(省略可能) ルート (/) または /usr ファイルシステムの復元は、非クリティカルなファイルシステムの復元よりも複雑です。理由は、これらのファイルシステムの復元中にローカル CD またはネットワークから起動する必要があるためです。	728 ページの「ルート (/) と /usr を復元する方法」

バックアップスケジュールを設定する際のガイドライン

「バックアップスケジュール」とは、ufsdump コマンドを実行するように設定するスケジュールです。この節では、バックアップスケジュールの作成時に検討すべき要素、ファイルシステムのバックアップ頻度に関するガイドラインを説明します。また、バックアップスケジュールの例もいくつか紹介します。

バックアップスケジュールは、以下の点を考慮に入れて作成します。

- バックアップに使用するテープの数を最小限に抑える
- バックアップの実行に使用できる時間
- 損傷したファイルシステムの完全復元に使用できる時間
- 不注意に削除した個々のファイルの検索に使用できる時間

バックアップ頻度

バックアップに費やす時間とメディアの数を最小限度に抑える必要がない場合は、完全バックアップを毎日実行してもかまいません。しかし、多くのサイトの場合、このようなバックアップ方法は現実的ではないので、ほとんどの場合は増分バックアップが使用されます。その場合は、サイトが過去4週間分のバックアップからファイルを十分復元できるようにしてください。そのためには、少なくとも1週分ごとに1組ずつ、合計4組のテープが必要で、各組を毎月使い回すこととなります。また、少なくとも一年分の月別のバックアップを保存し、数年分の年度別バックアップを保管しておく必要があります。

ダンプレベルを使用して増分バックアップを作成する

ufsdump コマンドで指定するダンプレベル (0-9) によって、どのファイルのバックアップが作成されるかが決まります。ダンプレベル0を指定すると、完全バックアップが作成されます。増分バックアップのスケジュール設定にレベル1から9までの番号が使用されますが、特に意味が定義されているわけではありません。レベル1から9は、累積バックアップまたは個別バックアップのスケジュール設定に使用する番号の範囲にすぎません。レベル1から9までが意味するのは、大小による相互関係だけです。

次の例で、1から9までのレベルを使用する増分ダンプの手順を示します。

日単位累積バックアップのダンプレベル

累積増分バックアップを毎日実行するのが、最も一般に使用される方法で、ほとんどの場合に推奨できます。次の例で、毎日レベル9のダンプを使用し、金曜日にはレベル5のダンプを使用してプロセスを再開するスケジュールを示します。

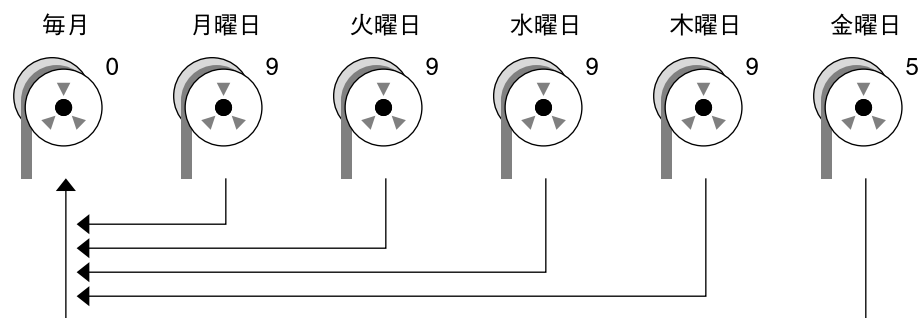


図 46-1 増分バックアップ: 日単位累積

前の例では、1 から 9 までの範囲内で他の番号を使用しても同じ結果が得られます。ポイントは、毎日同じ番号を使用し、金曜日にはそれより「小さい」番号を使用することです。たとえば、レベル 4、4、4、4、2 や 7、7、7、7、5 を指定してもかまいません。

日単位個別バックアップのダンプレベル

次の例で、1 日分の作業内容のみを別々のテープ上で保存するスケジュールを示します。この場合、月曜日から木曜日までは連続するダンプレベル番号 (3、4、5、6) を使用し、金曜日にはそれより小さい番号 (2) を使用します。

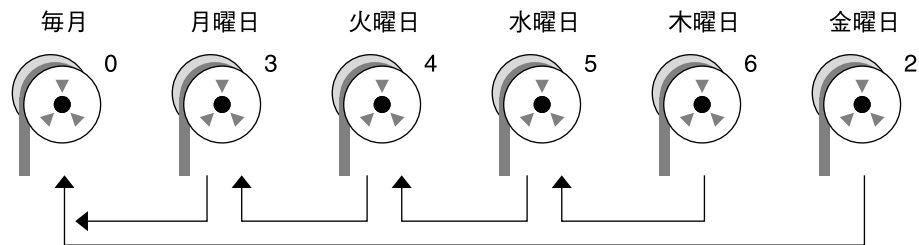


図 46-2 増分バックアップ: 日単位個別

前の例では、連番 6、7、8、9 の次に 2 を使用しても、5、6、7、8 の次に 3 を使用してもかまいません。番号自体の意味は定義されていません。番号の大小に意味があることに留意してください。

バックアップスケジュールの例

この節では、バックアップスケジュールのサンプルを示します。どのスケジュールも、完全バックアップ (ダンプレベル 0) から始めることと、`-u` オプションを使用して各バックアップを記録することを前提としています。

例 — 日単位累積、週単位累積バックアップ

次の表に、最も一般的に使用される増分バックアップスケジュールを示します。これは、ほとんどの場合に推奨できるスケジュールです。このスケジュールで実行される処理は次のとおりです。

- 前週の終わりの下位レベルのバックアップ以降に変更があったすべてのファイルが毎日保存される。

- 月～金のレベル9のバックアップの場合は、直前のレベル0またはレベル5が最も近い下位バックアップレベルになる。したがって各週のテープには、前週の終わり(第1週の場合は初期レベル0)以降に変更があったすべてのファイルが累積される。
- 毎週金曜日のレベル5の場合、レベルのうち一番近いレベルのバックアップは、月初めに実行されたレベル0である。したがって、毎週金曜日のテープには、月初めからその時点までに変更があったすべてのファイルが入っている。

表 46-6 日単位累積、週単位累積バックアップスケジュール

	開始日	月	火	水	木	金
月の1日目	0					
第1週		9	9	9	9	5
第2週		9	9	9	9	5
第3週		9	9	9	9	5
第4週		9	9	9	9	5

次の表に、日単位累積、週単位累積スケジュールを使用して、テープの内容が2週間でのどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 46-7 日単位累積、週単位累積スケジュールのテープの内容

	月	火	水	木	金
第1週	a b	a b c	a b c d	a b c d e	a b c d e f
第2週	g	g h	g h i	g h i j	a b c d e f g h i j k

日単位累積、週単位累積スケジュールのテープ要件

このスケジュールでは、6本(日単位テープを再利用したい場合)または9本(曜日ごとに4本の日単位テープを別々に使用したい場合)のテープが必要になります。その内訳は、レベル0に1本、金曜日に4本、日単位テープ用に1本または4本です。

ファイルシステム全体を復元する必要がある場合は、次の本数のテープが必要になります。その内訳は、レベル0が1本、最後の金曜日のテープ1本、前週の金曜日以降の最新の日単位テープ1本です。

例 — 日単位累積、週単位増分バックアップ

次の表に、各曜日のテープに月曜日(第1週の場合は初期レベル0)以降に変更があったすべてのファイルが累積され、毎週金曜日のテープにはその週に変更があったすべてのファイルが入っているスケジュールを示します。

表 46-8 日単位累積、週単位増分バックアップスケジュール

	開始日	月	火	水	木	金
月の1日目	0					
第1週		9	9	9	9	3
第2週		9	9	9	9	4
第3週		9	9	9	9	5
第4週		9	9	9	9	6

次の表に、日単位累積、週単位増分バックアップスケジュールの場合、テープの内容が2週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 46-9 日単位累積、週単位増分バックアップスケジュールのテープの内容

	月	火	水	木	金
第1週	a b	a b c	a b c d	a b c d e	a b c d e f
第2週	g	g h	g h i	g h i j	g h i j k

日単位累積、週単位増分バックアップスケジュールのテープ要件

このスケジュールでは、6本(日単位テープを再利用したい場合)または9本(曜日ごとに4本の日単位テープを別々に使用したい場合)のテープが必要になります。その内訳は、レベル0に1本、金曜日に4本、日単位テープ用に1本または4本です。

ファイルシステム全体を復元する必要がある場合、次の本数のテープが必要になります。その内訳は、レベル0が1本、すべての金曜日のテープ、前週の金曜日以降の最新の日単位テープ1本です。

例 — 日単位増分、週単位累積バックアップ

次の表に、各曜日のテープには前日以降に変更があったファイルのみが入り、毎金曜日のテープには月初めの初期レベル0以降に変更があったすべてのファイルが入るスケジュールを示します。

表 46-10 日単位増分、週単位累積バックアップスケジュール

	開始日	月	火	水	木	金
月の1日目	0					
第1週		3	4	5	6	2
第2週		3	4	5	6	2
第3週		3	4	5	6	2
第4週		3	4	5	6	2

次の表に、日単位増分、週単位累積スケジュールの場合、テープの内容が2週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 46-11 日単位増分、週単位累積バックアップスケジュールのテープの内容

	月	火	水	木	金
第1週	a b	c d	e f g	h i	a b c d e f g h i
第2週	j k l	m	n o	p q	a b c d e f g h i j k l m n o p q r s

日単位増分、週単位累積スケジュールのテープ要件

このスケジュールでは、少なくとも9本のテープが必要になります。これは日単位テープを毎週再利用することが前提となっていますが、これはお勧めできません。週単位テープを1カ月保存する場合は、21本のテープが必要になります。内訳は、レベル0に1本、金曜日に4本、日単位テープ4本または16本です。

ファイルシステム全体を復元する必要がある場合は、次のテープが必要になります。その内訳は、レベル0に1本、前回の金曜日のテープ、最後の金曜日以降のその週のすべての日単位テープです。

例—サーバーの月単位バックアップスケジュール

次の表に、ユーザーがプログラム開発や文書作成のようなファイル集約型の作業を実行する小型ネットワーク上の、使用頻度の高いファイルサーバーのバックアップ方針の例を示します。この例は、バックアップ期間が日曜日に始まり、1週7日間を4週間行うものと想定しています。

表 46-12 例—サーバーの月単位バックアップスケジュール

ディレクトリ	日付	レベル	テープ名
ルート (/)	第 1 日曜日	0	<i>n</i> 本のテープ
/usr	第 1 日曜日	0	<i>n</i> 本のテープ
/export	第 1 日曜日	0	<i>n</i> 本のテープ
/export/home	第 1 日曜日	0	<i>n</i> 本のテープ
	第 1 月曜日	9	A
	第 1 火曜日	9	B
	第 1 水曜日	5	C
	第 1 木曜日	9	D
	第 1 金曜日	9	E
	第 1 土曜日	5	F
ルート (/)	第 2 日曜日	0	<i>n</i> 本のテープ
/usr	第 2 日曜日	0	"
/export	第 2 日曜日	0	"
/export/home	第 2 日曜日	0	"
	第 2 月曜日	9	G
	第 2 火曜日	9	H
	第 2 水曜日	5	I
	第 2 木曜日	9	J
	第 2 金曜日	9	K
	第 2 土曜日	5	L
ルート (/)	第 3 日曜日	0	<i>n</i> 本のテープ
/usr	第 3 日曜日	0	"
/export	第 3 日曜日	0	"
/export/home	第 3 日曜日	0	"
	第 3 月曜日	9	M
	第 3 火曜日	9	N
	第 3 水曜日	5	O
	第 3 木曜日	9	P

表 46-12 例—サーバーの月単位バックアップスケジュール (続き)

ディレクトリ	日付	レベル	テープ名
	第 3 金曜日	9	Q
	第 3 土曜日	5	R
ルート (/)	第 4 日曜日	0	n 本のテープ
/usr	第 4 日曜日	0	"
/export	第 4 日曜日	0	"
/export/home	第 4 日曜日	0	"
	第 4 月曜日	9	S
	第 4 火曜日	9	T
	第 4 水曜日	5	U
	第 4 木曜日	9	V
	第 4 金曜日	9	W
	第 4 土曜日	5	X

このスケジュールでは、 $4n$ 本のテープ (ルート (/)、/usr、/export、および /export/home ファイルシステムの 4 回の完全バックアップに必要な本数) に加えて、/export/home ファイルシステムの増分バックアップ用に 24 本のテープを使用します。このスケジュールは、増分バックアップごとに 1 本ずつテープを使用し、それを 1 カ月は保存することを前提としています。

このスケジュールの機能を、次に示します。

1. 日曜日ごとに、ルート (/)、/usr、/export、および /export/home ファイルシステムの完全バックアップ (レベル0) を実行します。レベル0 のテープを少なくとも 3 カ月は保存します。
2. 月の第 1 月曜日に、テープ A を使用して /export/home ファイルシステムのレベル9 のバックアップを実行します。ufsdump コマンドは、下のレベルのバックアップ以降に変更があったすべてのファイルをコピーします。この場合は、日曜日に実行したレベル0 のバックアップ以降となります。
3. 月の第 1 火曜日に、テープ B を使用して /export/home ファイルシステムのレベル9 のバックアップを実行します。この場合も、ufsdump コマンドは、下のレベル (日曜日のレベル0 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
4. 第 1 水曜日に、テープ C を使用してレベル5 のバックアップを実行します。ufsdump コマンドは、日曜日以降に変更があったすべてのファイルをコピーします。

5. 木曜日と金曜日には、テープ D と E を使用してレベル 9 のバックアップを実行します。ufsdump コマンドは、下のレベルのバックアップ (水曜日のレベル 5 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
6. 月の第 1 土曜日に、/export/home のレベル 5 のバックアップを実行します。このバックアップでは、下のレベルのバックアップ (この場合は日曜日に実行したレベル 0 のバックアップ) 以降に変更があったすべてのファイルがコピーされます。テープを再利用する場合は、テープ A から F までを次の 4 週間の第 1 月曜日までは保存しておきます。
7. 次の 3 週間は、テープ G から L までと、日曜日のレベル 0 用に 4n 本のテープを使用して、手順 1 から 6 までを繰り返します。
8. 4 週ごとに、レベル 0 用に新しいテープ 1 組と、増分バックアップ用のテープ A から X までを再利用して、手順 1 から 7 までを繰り返します。レベル 0 のテープは、3 カ月後に再利用できるようになります。

このスケジュールでは、各ファイルを 1 カ月間で段階別に保存できます。多数のテープが必要ですが、テープのライブラリを確実に用意できます。テープの本数を減らすには、テープ A から F までを毎週再利用します。

バックアップスケジュールに関するその他の推奨事項

次の表に、バックアップスケジュールに関するその他の推奨事項を示します。

表 46-13 バックアップスケジュールに関するその他の推奨事項

必要なファイル復元操作	バックアップ間隔	コメント
各ファイルの別バージョン (ワード処理に使用するファイルシステムなど) を復元する必要がある	<ul style="list-style-type: none"> ■ 作業日ごとに日単位増分バックアップを実行する ■ 日単位増分バックアップには同じテープを再利用しない 	このスケジュールでは、その日に変更があったすべてのファイルが保存されるだけでなく、下のレベルの最後のバックアップ以降に変更があったファイルがディスク上に残る。ただし、このスケジュールの場合は、火曜日に変更があったファイルが木曜日にも変更されると、金曜日の下のレベルのバックアップでは、火曜日の夜ではなく木曜日の夜に変更されたように見えるので、毎日異なるテープを使用する必要がある。ユーザーが火曜日のバージョンを必要とする場合には、火曜日 (または水曜日) のバックアップテープを保存しておかなければ、それを復元できない。また、火曜日と水曜日には存在したファイルが木曜日に削除されても、金曜日の下のレベルのバックアップには表示されない
ファイルシステム全体を短時間で復元する必要がある	下位のレベルのバックアップを頻繁に実行する	—
同じサーバー上で多数のファイルシステムのバックアップを作成する	ファイルシステムごとにスケジュールをずらすことを検討する	この方法では、すべてのレベル 0 のバックアップを同じ日に実行しないことになる
テープの本数を最小限に抑える	<p>1 週間に実行する増分バックアップのレベルを上げる</p> <p>週末に実行するバックアップのレベルを上げる。日単位と週単位の増分バックアップを同じテープに入れる</p> <p>日単位と週単位の増分バックアップを同じテープに入れる</p>	<p>毎日の変更のみが各日単位テープに保存される</p> <p>変更は (月単位ではなく) 週単位でしか週単位テープに保存されない</p> <p>この操作には、<code>ufsdump</code> コマンドの <code>"no-rewind"</code> オプションを使用する</p>

第 47 章

ファイルとファイルシステムのバックアップ (手順)

この章では、`ufsdump` コマンドを使用したファイルシステムのバックアップ手順について説明します。

手順の詳細については、697 ページの「ファイルとファイルシステムのバックアップ (作業マップ)」を参照してください。

バックアップの実行方法の概要については、第 46 章を参照してください。

`ufsdump` コマンドの構文、オプション、引数の詳細については、第 50 章を参照してください。

ファイルとファイルシステムのバックアップ (作業マップ)

作業	説明	参照先
1. ファイルシステムのバックアップ準備を行う。	バックアップで使用するファイルシステム、バックアップタイプ、テープデバイスを確認する。	698 ページの「ファイルシステムバックアップの実行準備」
2. ファイルシステムのバックアップに必要なテープの数を決定する。	ファイルシステムの完全バックアップに必要なテープの数を決定する。	699 ページの「完全バックアップに必要なテープ数を判別する」

作業	説明	参照先
3. ファイルシステムをバックアップする。	<p>ファイルシステムの完全バックアップを実行して、全ファイルの基準コピーを取得する。</p> <p>日単位で変更されたファイルのコピーを保持することが、サイトで重要かどうかに基づいて、ファイルシステムの増分バックアップを実行する。</p>	700 ページの「ファイルシステムのバックアップをテープに作成する方法」

ファイルシステムバックアップの実行準備

ファイルシステムバックアップの準備は、計画を作成することから始まります。詳細については、第 46 章を参照してください。この段階では、次の項目を決定します。

- バックアップを作成するファイルシステム
- 実行するバックアップのタイプ (完全または増分)
- バックアップのスケジュール
- テープドライブ

この節では、ファイルシステムのバックアップを作成する前に実行する必要がある作業について説明します。次の作業が含まれます。

- バックアップを作成するファイルシステムの名前を検索する。
- 完全バックアップの作成に必要なテープの本数を決定する。

▼ ファイルシステム名を検索する方法

1. `/etc/vfstab` ファイルの内容を表示します。

```
$ more /etc/vfstab
```

2. `mount point` 列に表示されるファイルシステム名を調べます。
3. ファイルシステムのバックアップを作成する際、`mount point` 列に表示されたディレクトリ名を使用します。

例 — ファイルシステム名を検索する

この例では、バックアップ対象のファイルシステムはルート (`/`)、`/usr`、`/datadb`、および `/export/home` です。

```

$ more /etc/vfstab
#device          device          mount          FS   fsck mount  mount
#to mount        to fsck         point          type pass at boot options
#
fd                -                /dev/fd        fd   -   no    -
/proc            -                /proc          proc -   no    -
/dev/dsk/c0t0d0s1 -                -              swap -   no    -
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /              ufs  1   no    -
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /usr           ufs  1   no    -
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /datab         ufs  2   yes   -
/dev/dsk/c0t0d0s7 /dev/rdisk/c0t0d0s7 /export/home   ufs  2   yes   -
swap            -                /tmp           tmpfs -   yes   -

```

▼ 完全バックアップに必要なテープ数を判別する

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. バックアップのサイズをバイト単位で予測します。

```
# ufsdump S file-system
```

S を指定すると、バックアップの実行に必要な予想バイト数が表示されます。

3. 予測サイズをテープの容量で割り、必要なテープの数を確認します。
テープ容量のリストについては、表 46-5 を参照してください。

例 — テープの本数を決定する

次の例では、150M バイトのテープに 489472 バイトのファイルシステムが入ります。

```
# ufsdump S /export/home
489472
```

ファイルシステムのバックアップ

バックアップを実行する際の一般的なガイドラインは次のとおりです。

- シングルユーザーモードを使用するか、ファイルシステムをマウント解除する (ファイルシステムのスナップショットを作成している場合を除く)。UFS スナップショットの詳細については、第 48 章を参照。
- ディレクトリレベルの処理 (ファイルの作成、削除、名前変更など) とファイルレベルの処理が行われているときにファイルシステムのバックアップを実行すると、バックアップに組み込まれないデータがあるので注意する。

- 単一のシステムで `ufsdump` コマンドを実行し、リモートシェルまたはリモートログインを通じてネットワーク上のシステムグループのバックアップを実行して、出力をテープドライブがあるシステムに転送できる。通常、テープドライブは `ufsdump` コマンドを実行するシステム上にあるが、他の場所にあってもかまわない。
ファイルのバックアップをリモートドライブに作成するには、`ufsdump` コマンドの出力を `dd` コマンドにパイプする方法もある。`dd` コマンドの使用方法については、第 51 章を参照。
- ネットワーク上でリモートバックアップを実行する場合、テープドライブを持つシステムの `/.rhosts` ファイル中には、ドライブを使用する各クライアントのエントリが入っていないなければならない。また、バックアップを実行する各システムの `/.rhosts` ファイルには、バックアップを開始するシステムのエントリが入っていないなければならない。
- システム上でリモートテープデバイスを指定するには、リモートテープドライブを保持するシステムの OS バージョンと一致する命名規則を使用する。たとえば、SunOS 4.1.1 またはその互換バージョンが稼働中のシステム上のリモートドライブにはデバイス `/dev/rst0` を使用する。また、Solaris 9 またはその互換バージョンを実行しているシステムでは、デバイス `/dev/rmt/0` を使用する。

注 - NIS+ マスターサーバーをバックアップする場合は、`nisbackup` コマンドを使用してください。このコマンドの使用方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : FNS、NIS+ 編)』を参照してください。

▼ ファイルシステムのバックアップをテープに作成する方法

`ufsdump` コマンドを使用してファイルシステムのバックアップを作成する一般的な手順を示します。この例では、オプションと引数の使用方法を示しています。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. システムをレベル **S** (シングルユーザーモード) に移行します。

```
# shutdown -g30 -y
```

3. (省略可能) ファイルシステムの整合性をチェックします。

```
# fsck -m /dev/rdisk/device-name
```

`fsck -m` コマンドを実行すると、ファイルシステムの整合性がチェックされます。たとえば、電源障害が発生すると、ファイルが不整合になることがあります。`fsck` コマンドの詳細については、第 43 章を参照してください。

4. ファイルシステムをリモートテープドライブにバックアップする場合

- a. テープドライブが接続されているシステム (テープサーバー) の `./rhosts` ファイルに、次のエントリを追加します。

```
host root
```

`host` エントリには、`ufsdump` コマンドを実行してバックアップを行うシステムの名前を指定します。

- b. テープサーバー上で、上記の `./rhosts` ファイルに追加したホストに、ネームサービス経由でアクセスできることを確認します。
5. テープドライブのデバイス名を確認します。
デフォルトのテープドライブは、`/dev/rmt/0` です。
6. 書き込み保護をしていないテープをテープドライブに挿入します。
7. システムのバックアップを実行します。

```
# ufsdump options arguments filenames
```

次の例では、最も一般的な `ufsdump` オプションおよび引数の使用方法を示します。

- 701 ページの「例 — ルート (/) の完全バックアップ」
- 702 ページの「例 — ルート (/) の増分バックアップ」
- 703 ページの「例 — 個々のホームディレクトリの完全バックアップ」
- 704 ページの「例 — リモートシステムへの完全バックアップ (Solaris 9 データを Solaris 9 システムへ)」

その他の `ufsdump` オプションと引数については、第 50 章を参照してください。

8. プロンプトが表示されたら、テープを取り出して、次のテープを挿入します。
9. 各テープにボリューム番号、ダンプレベル、日付、システム名、ディスクスライス、ファイルシステム名を記入したラベルを貼ります。
10. **Control + D** キーを押してシステムをレベル 3 の動作に戻します。
11. バックアップが正常に実行されたことを確認します。

```
# ufsrestore tf device-name
```

例 — ルート (/) の完全バックアップ

次の例では、ルート (/) ファイルシステムの完全バックアップを実行する方法を示します。この例では、バックアップの実行前にシステムをシングルユーザーモードにしています。次の `ufsdump` オプションが含まれます。

- `0` は、レベル 0 のダンプ (完全バックアップ) であることを示す。
- `u` は、このバックアップの実行日に `/etc/dumpdates` ファイルが更新されることを示す。
- `c` は、カートリッジテープデバイスを示す。

- `f /dev/rmt/0` は、テープデバイスを示す。
- `/` はバックアップするファイルシステムを示す。

```
# shutdown -g30 -y
# ufsdump 0ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Wed Sep 05 13:27:20 2001
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t1d0s0 (earth:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 57150 blocks (27.91MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 57076 blocks (27.87MB) on 1 volume at 265 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Wed Sep 05 13:27:20 2001
# ufsrestore tf /dev/rmt/0
      2      .
      3      ./lost+found
    3776      ./usr
    7552      ./var
   11328      ./export
   15104      ./export/home
   18880      ./etc
   22656      ./etc/default
   22657      ./etc/default/sys-suspend
   22673      ./etc/default/cron
   22674      ./etc/default/devfsadm
   22675      ./etc/default/dhcppagent
   22676      ./etc/default/fs
   22677      ./etc/default/inetinit
   22678      ./etc/default/kbd
   22679      ./etc/default/mpathd
   22680      ./etc/default/nfslogd
   22681      ./etc/default/passwd
      .
      .
      .
# (Control+D を押して、システムをレベル 3 に戻す)
```

例 — ルート (/) の増分バックアップ

次の例は、ルート (/) ファイルシステムの増分バックアップを実行する方法を示します。次の `ufsdump` オプションが含まれます。

- `9` は、レベル 9 のダンプ (増分バックアップ) であることを示す。
- `u` は、このバックアップの実行日に `/etc/dumpdates` ファイルが更新されることを示す。
- `c` は、カートリッジテープデバイスを示す。
- `f /dev/rmt/0` は、テープデバイスを示す。

- / はバックアップするファイルシステムを示す。

```
# ufsdump 9ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 9 dump: Fri Jul 13 10:58:12 2001
DUMP: Date of last level 0 dump: Fri Jul 13 10:46:09 2001
DUMP: Dumping /dev/rdisk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 200 blocks (100KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 124 blocks (62KB) on 1 volume at 8 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Fri Jul 13 10:58:12 2001
# ufsrestore tf /dev/rmt/0
      2      .
      3      ./lost+found
    5696      ./usr
   11392      ./var
   17088      ./export
   22784      ./export/home
   28480      ./opt
    5697      ./etc
   11393      ./etc/default
   11394      ./etc/default/sys-suspend
   11429      ./etc/default/cron
   11430      ./etc/default/devfsadm
   11431      ./etc/default/dhcpagent
   11432      ./etc/default/fs
   11433      ./etc/default/inetinit
   11434      ./etc/default/kbd
   11435      ./etc/default/nfslogd
   11436      ./etc/default/passwd
   11437      ./etc/default/tar
      .
      .
      .
```

例 — 個々のホームディレクトリの完全バックアップ

次の例は、/export/home/kryten ディレクトリの完全バックアップを実行する方法を示します。次の ufsdump オプションが含まれます。

- 0 は、レベル 0 のダンプ (完全バックアップ) であることを示す。
- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示す。
- c は、カートリッジテープデバイスを示す。

- `f /dev/rmt/0` は、テープデバイスを示す。
- `/export/home/kryten` は、バックアップするディレクトリを示す。

```
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Fri Jul 13 11:30:45 2001
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t3d0s7 (pluto:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 232 blocks (116KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 124 blocks (62KB) on 1 volume at 8 KB/sec
DUMP: DUMP IS DONE
# ufsrestore tf /dev/rmt/0
      2      .
2688      ./kryten
5409      ./kryten/letters
5410      ./kryten/letters/letter1
5411      ./kryten/letters/letter2
5412      ./kryten/letters/letter3
2689      ./kryten/.profile
8096      ./kryten/memos
      30      ./kryten/reports
      31      ./kryten/reports/reportA
      32      ./kryten/reports/reportB
      33      ./kryten/reports/reportC
#
```

例 — リモートシステムへの完全バックアップ (Solaris 9 データを Solaris 9 システムへ)

次の例では、Solaris 9 システム (starbug) のローカルの `/export/home` ファイルシステムを、リモートの Solaris 9 システム (earth) のテープデバイスに完全バックアップする方法を示します。次の `ufsdump` オプションが含まれます。

- `0` は、レベル 0 のダンプ (完全バックアップ) であることを示す。
- `u` は、このバックアップの実行日に `/etc/dumpdates` ファイルが更新されることを示す。
- `c` は、カートリッジテープデバイスを示す。
- `f earth:/dev/rmt/0` は、リモートシステム名およびテープデバイスを示す。
- `/export/home` は、バックアップ対象のファイルシステムを示す。

```
# ufsdump 0ucf earth:/dev/rmt/0 /export/home
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Wed Sep 05 14:52:31 2001
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s7 (starbug:/export/home) to earth:/dev ...
```



```
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 266 blocks (133KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 250 blocks (125KB) on 1 volume at 247 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Wed Sep 05 14:52:31 2001
# ufsrestore tf earth:/dev/rmt/0
      2      .
      3      ./lost+found
     7168    ./rimmer
     7169    ./rimmer/.profile
    21504    ./rimmer/skdir
    21505    ./rimmer/skdir/scd557
    21506    ./rimmer/skdir/scd772
    10752    ./lister
    10753    ./lister/.profile
    10754    ./lister/filea
    10755    ./lister/fileb
    10756    ./lister/filec
    14336    ./pmorph
    14337    ./pmorph/.profile
     3584    ./pmorph/bigdir
     3585    ./pmorph/bigdir/bigfile
    17920    ./pmorph/smallldir
    17921    ./pmorph/smallldir/smallfile
#
```


第 48 章

UFS スナップショットの使用 (手順)

この章では、UFS スナップショットの作成およびバックアップ方法について説明します。

UFS スナップショットの作成に関連した手順については、707 ページの「UFS スナップショットの使用 (作業マップ)」を参照してください。

バックアップの実行方法の概要については、第 46 章を参照してください。

UFS スナップショットの使用 (作業マップ)

作業	説明	参照先
1. UFS スナップショットを作成する。	fssnap コマンドを使用して、ファイルシステムの読み取り専用コピーを作成する。	710 ページの「UFS スナップショットを作成する方法」
2. UFS スナップショットの情報を表示する。	UFS スナップショット情報 (raw スナップショットデバイスなど)を確認する。	711 ページの「UFS スナップショットの情報を表示する方法」
3. (省略可能) UFS スナップショットを削除する。	バックアップ済みまたは不要になったスナップショットを削除する。	712 ページの「UFS スナップショットを削除する方法」
4. UFS スナップショットをバックアップする。	次の方法のいずれかを選択する。	

作業	説明	参照先
	<p>ufsdump コマンドを使用して、UFS スナップショットの完全バックアップを作成する。</p> <p>ufsdump コマンドを使用して、UFS スナップショットの増分バックアップを作成する。</p> <p>tar コマンドを使用して、UFS スナップショットのバックアップを作成する。</p>	<p>713 ページの「UFS スナップショットの完全バックアップを作成する方法 (ufsdump)」</p> <p>714 ページの「UFS スナップショットの増分バックアップの作成方法 (ufsdump)」</p> <p>714 ページの「UFS スナップショットのバックアップ方法 (tar)」</p>
5. (省略可能) UFS スナップショットからデータを復元する。	ufsrestore コマンドを使用して、データを復元するのと同じ方法で、UFS スナップショットを復元する。	725 ページの「ファイルシステム全体を復元する方法」

UFS スナップショットの概要

Solaris リリースには、ファイルシステムのマウント中にファイルシステムをバックアップするための、`fssnap` コマンドが含まれています。`fssnap` コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。「スナップショット」は、バックアップ操作のためのファイルシステムの一時的イメージです。

`fssnap` コマンドを実行すると、1つの仮想デバイスと1つのバックアップストアファイルが作成されます。ユーザーは、既存の Solaris バックアップコマンドを使用して、実際のデバイスのように動作し実際のデバイスのように見える「仮想デバイス」をバックアップすることができます。「バックアップストア (backing-store)」ファイルは、スナップショット作成後に変更されたデータのコピーを含むビットマップ化ファイルです。

なぜ UFS スナップショットを使用するか

UFS スナップショットにより、バックアップ時に、ファイルシステムをマウントされた状態にしシステムをマルチユーザーモードにしておくことができます。これまで、バックアップを実行するために `ufsdump` コマンドを使用する時は、ファイルシステムをアクティブでない状態に保つためにシステムをシングルユーザーモードにすることが推奨されていました。より確実なバックアップのために、`tar` や `cpio` などの追加の Solaris のバックアップコマンドを使用して UFS スナップショットのバックアップを行うこともできます。

`fssnap` コマンドにより、企業レベルではないシステムの管理者が、大規模な記憶容量の必要なく、Sun StorEdge™ Instant Image のような企業レベルツールのパワーを得ることができます。

UFS スナップショットは、Instant Image 製品に似ています。Instant Image は、取り込まれるファイルシステム全体のサイズに等しいスペースを割り当てます。ただし、UFS スナップショットが作成するバックキングストアファイルは、必要なディスクスペースの容量しか占有しないため、バックキングストアファイルのサイズに上限を設定することも可能です。

次の表は、UFS スナップショットと Instant Image との特徴的な違いを示します。

UFS スナップショット	Instant Image
バックキングストアファイルのサイズは、スナップショットがとられた後のデータの変更に量による	バックキングストアファイルのサイズは、コピーされるファイルシステム全体のサイズに等しい
システムのリポート後は保持されない	システムのリポート後も保持される
UFS ファイルシステムで動作する	ルート (/) または /usr ファイルシステムでは使用できない
Solaris 8 1/01 以降のリリース	Sun StorEdge 製品の一部

UFS スナップショットは大規模なファイルシステムをコピーすることができますが、企業レベルのシステムには Instant Image の方が適しています。UFS スナップショットは、小さめのシステムに適しています。

UFS スナップショットのパフォーマンス上の問題

UFS スナップショットの初回作成時に、ファイルシステムのユーザーが短い一時停止に気づく場合があります。一時停止の時間は、取り込まれるファイルシステムのサイズに応じて増加します。スナップショットがアクティブな間、ファイルシステムへの書き込み時にユーザーが若干のパフォーマンス低下に気づく場合がありますが、ファイルシステムからの読み込み時には影響はありません。

UFS スナップショットの作成と削除

`fssnap` コマンドを使用して UFS スナップショットを作成する場合、バックキングストアファイルがどれだけのディスクスペースを消費するかを監視してください。バックキングストアファイルは、初めはスペースを全く使用せず、その後特によく使用されて

いるシステムにおいて急速に拡大します。バックストアファイルが拡大するのに十分なスペースを必ず確保しておくか、または `-o maxsize=n [k,m,g]` オプション (`n [k,m,g]` はバックストアファイルの最大限のサイズ) でそのサイズを制限してください。



注意 - バックストアファイルにスペースが不足する場合、スナップショットが削除されてしまうことがあり、バックアップが失敗します。スナップショットのエラーの可能性を調べるため、`/var/adm/messages` ファイルをチェックしてください。

詳細については、`fssnap_ufs(1M)` のマニュアルページを参照してください。

▼ UFS スナップショットを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ファイルシステムに、バックストアファイル用の十分なディスクスペースが存在することを確認してください。

```
# df -k
```

3. 同じ場所に同じ名前の既存のバックストアファイルが存在していないことを確認します。

```
# ls /backing-store-file
```

4. UFS スナップショットを作成します。

```
# fssnap -F ufs -o bs=/backing-store-file /file-system
```

注 - バックストアファイルは、スナップショットを作成するファイルシステムとは異なるファイルシステムに存在する必要があります。

5. スナップショットが作成されたことを確認します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

例 — UFS スナップショットを作成する

次の例は、`/usr` ファイルシステムのスナップショットを作成する方法を示します。バックストアファイルは `/scratch/usr.back.file`、仮想デバイスは `/dev/fssnap/1` です。

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr  
/dev/fssnap/1
```

次の例は、バックイングストアファイルを 500M バイトに制限する方法を示します。

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /export/home
/dev/fssnap/1
```

▼ UFS スナップショットの情報を表示する方法

`fssnap -i` オプションを使用すると、システムの現在のスナップショットを表示することができます。1つのファイルシステムを指定する場合、そのスナップショットについての詳細な情報が表示されます。特定のファイルシステムを指定しない場合は、現在の UFS スナップショットすべておよび対応する仮想デバイスの情報が表示されます。

注 - 次の例に示すように、拡張スナップショット情報を表示する場合は、UFS ファイルシステム固有の `fssnap` コマンドを使用してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 現在のスナップショットをリスト表示します。

```
# /usr/lib/fs/ufs/fssnap -i
Snapshot number           : 0
Block Device              : /dev/fssnap/0
Raw Device                : /dev/rfssnap/0
Mount point               : /export/home
Device state              : idle
Backing store path        : /var/tmp/bs.file
Backing store size        : 0 KB
Maximum backing store size : Unlimited
Snapshot create time      : Wed Aug 29 15:22:06 2001
Copy-on-write granularity : 32 KB
```

特定の 1 つのスナップショットについての詳細な情報を表示する場合は、次のようにします。

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number           : 0
Block Device              : /dev/fssnap/0
Raw Device                : /dev/rfssnap/0
Mount point               : /usr
Device state              : idle
Backing store path        : /var/tmp/bs.file
Backing store size        : 0 KB
Maximum backing store size : Unlimited
Snapshot create time      : Wed Aug 29 15:23:35 2001
Copy-on-write granularity : 32 KB
```

UFS スナップショットの削除

UFS スナップショットを作成する際、バックアップストアファイルがリンクされないように指定することができます。これは、そのバックアップストアファイルが、スナップショットが削除された後で削除されることを示します。UFS スナップショットを作成する際に `-o unlink` オプションを指定しない場合は、後で手動で削除する必要があります。

バックアップストアファイルは、バックアップストアファイルの削除に `-o unlink` オプションを使用した場合はスナップショットが削除されるまで、またはバックアップストアファイルを手動で削除するまで、ディスクスペースを使用します。

▼ UFS スナップショットを削除する方法

スナップショットは、システムをリブートするか、あるいは `fssnap -d` コマンドを使用して UFS スナップショットを含むファイルシステムのパスを指定することで、削除できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. 削除するスナップショットを特定します。

```
# /usr/lib/fs/ufs/fssnap -i
```

3. スナップショットを削除します。

```
# fssnap -d /file-system
Deleted snapshot 1.
```

4. (省略可能) スナップショットの作成時に `-o unlink` オプションを使用しなかった場合は、そのバックアップストアファイルを手動で削除する必要があります。

```
# rm /file-system/backing-store-file
```

例 — UFS スナップショットを削除する

次の例は、`unlink` オプションを使用しなかった場合に、スナップショットを削除する方法を示します。

```
# fssnap -i
    0 / 1 /usr
# fssnap -d /usr
Deleted snapshot 1.
# rm /scratch/usr.back.file
```

UFS スナップショットのバックアップ

UFS スナップショットの完全または増分バックアップを作成できます。UFS スナップショットのバックアップ作成に、標準の Solaris バックアップコマンドを使用できます。

UFS スナップショットを含む仮想デバイスは、標準の読み取り専用デバイスとして動作します。これは、仮想デバイスを、ファイルシステムのデバイスをバックアップするかのようにバックアップできることを意味します。

`ufsdump` コマンドを使用して UFS スナップショットをバックアップする場合、バックアップ時にスナップショットの名前を指定することができます。詳細については、次の節を参照してください。

▼ UFS スナップショットの完全バックアップを作成する方法 (`ufsdump`)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. バックアップする **UFS** スナップショットを指定します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

たとえば、次のようになります。

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number      : 0
Block Device         : /dev/fssnap/0
Raw Device           : /dev/rfssnap/0
Mount point          : /usr
Device state         : idle
Backing store path   : /var/tmp/back.store
Backing store size   : 576 KB
Maximum backing store size : Unlimited
Snapshot create time : Wed Dec 12 09:39:37 2001
Copy-on-write granularity : 32 KB
```

3. **UFS** スナップショットをバックアップします。

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

たとえば、次のようになります。

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4. スナップショットがバックアップされたことを確認します。

```
# ufsrestore tf /dev/rmt/0
```

▼ UFS スナップショットの増分バックアップの作成方法 (ufsdump)

UFS スナップショットの増分バックアップを実行する (最後のスナップショット以降に変更のあったファイルだけをバックアップする) 場合、`ufsdump` コマンドと新規 `N` オプションを組み合わせて使用します。このオプションは、増分ダンプをトラックするために `/etc/dumpdates` ファイルに挿入されるファイルシステムのデバイス名を指定します。

次の例では、`ufsdump` コマンド内で `fssnap` コマンドを組み込んでファイルシステムの増分バックアップを作成しています。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. UFS スナップショットの増分バックアップを作成します。

たとえば、次のようになります。

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdisk/c0t1d0s0 `fssnap -F ufs -o raw,bs=  
/export/scratch,unlink /dev/rdisk/c0t1d0s0`
```

上記の例では、ブロックデバイスではなく `raw` デバイスの名前を表示するために `-o raw` オプションが使用されています。このオプションの使用により、`fssnap` コマンドを `raw` デバイスを必要とするコマンド (`ufsdump` コマンドなど) に組み込むことが容易になります。

3. スナップショットがバックアップされたことを確認します。

```
# ufsrestore ta /dev/rmt/0
```

▼ UFS スナップショットのバックアップ方法 (tar)

`tar` コマンドを使用してスナップショットをバックアップする場合、バックアップを行う前にスナップショットをマウントします。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. スナップショット用のマウントポイントを作成します。

たとえば、次のようになります。

```
# mkdir /backups/home.bkup
```

3. スナップショットをマウントします。

```
# mount -F ufs -o ro /dev/fssnap/1 /backups/home.bkup
```

4. マウントスナップショットのディレクトリに移動します。

```
# cd /backups/home.bkup
```

5. `tar` コマンドを使用して、スナップショットをバックアップします。

```
# tar cvf /dev/rmt/0 .
```

UFS スナップショットのバックアップからのデータの復元

仮想デバイスから作成されたバックアップは、基本的には、スナップショットがとられた時点でのオリジナルのファイルシステムの状態を表しています。バックアップから復元を行う場合は、オリジナルのファイルシステムから直接そのバックアップを作成した(たとえば `ufsrestore` コマンドを使用してバックアップを実行した)かのように復元します。`ufsrestore` コマンドを使用してファイルまたはファイルシステムを復元する方法については、第 49 章を参照してください。

第 49 章

ファイルとファイルシステムの復元 (手順)

この章では、`ufsdump` コマンドを使用してバックアップを作成したファイルおよびファイルシステムを、`ufsrestore` コマンドを使用して復元する方法について説明します。

ファイルおよびファイルシステムの復元に関連した手順については、717 ページの「ファイルおよびファイルシステムのバックアップを復元する (作業マップ)」を参照してください。

ファイルやファイルシステムのアーカイブ、復元、コピー、または移動に使用可能な他のコマンドについては、第 51 章を参照してください。

ファイルおよびファイルシステムの バックアップを復元する (作業マップ)

次の作業マップに、ファイルおよびファイルシステムの復元手順を示します。

作業	説明	参照先
ファイルとファイルシステムの復元準備	復元するファイルシステムやファイル、テープデバイス、およびファイルの復元方法を決定する。	718 ページの「ファイルとファイルシステムを復元するための準備」
使用するテープの決定	バックアップテープを調べ、復元するファイルまたはファイルシステムを含む最新のバックアップの日付を確認する。	720 ページの「使用するテープを決定する方法」
ファイルの復元	次のいずれかの復元方法を選択する。	

作業	説明	参照先
	<p>ファイルの対話的な復元 - 正確なファイル名を把握していない場合、この方法を使用する。この方法では、メディアの内容を参照して、ファイルやディレクトリ単位で選択できる。</p> <p>ファイルの非対話的な復元 - 復元するファイル名を把握している場合には、この方法を使用して高速に操作できる。</p> <p>ファイルシステムの復元 - 復元時の手順の一部として新規ディスクドライブを使用できる場合は、この方法を使用する。</p>	<p>721 ページの「対話式でファイルを復元する方法」</p> <p>723 ページの「例 — 特定のファイルを復元する」</p> <p>725 ページの「ファイルシステム全体を復元する方法」</p>
ルート (/) または /usr ファイルシステムの復元	ルート (/) または /usr ファイルシステムを復元する場合には、ローカル CD またはネットワークからシステムを起動する必要がある。	728 ページの「ルート (/) と /usr を復元する方法」

ファイルとファイルシステムを復元するための準備

ufsrestore コマンドは、ufsdump コマンドを使用して作成されたバックアップから、現在の作業ディレクトリにファイルをコピーします。ufsrestore コマンドを使用すると、レベル 0 のダンプとそれ以降の増分ダンプからファイルシステム階層全体を読み込み直したり、任意のバックアップテープから個々のファイルを復元できます。ufsrestore コマンドがスーパーユーザーの権限で実行された場合には、ファイルの所有者、最新の変更時刻、モード (ファイルのアクセス権) は元のまま、ファイルが復元されます。

ファイルまたはファイルシステムの復元を開始する前に、次の点を確認してください。

- 必要なテープ (またはフロッピーディスク)
- ファイルシステム全体を復元する raw デバイス名
- 使用するテープデバイスのタイプ
- テープドライブのデバイス名 (ローカルまたはリモート)

ファイルシステム名の確認

バックアップテープに適切な名前が付いている場合は、テープラベルに入っているファイルシステム名 (`/dev/rdisk/device-name`) が使えるはずですが、詳細については、698 ページの「ファイルシステム名を検索する方法」を参照してください。

必要なテープデバイスのタイプの決定

ファイルを復元するには、バックアップメディアと互換性のあるテープデバイスを使用する必要があります。バックアップメディアの形式によって、ファイルの復元にどんなドライブを使用しなければならないかが決まります。たとえば、使用するバックアップメディアが 8mm テープの場合、ファイルの復元には 8mm テープドライブを使わなければなりません。

テープデバイス名の決定

テープデバイス名 (`/dev/rmt/n`) をバックアップテープラベル情報の一部として指定しているかも知れません。同じドライブを使ってバックアップテープを復元しようとする場合には、ラベル内にあるデバイス名を使うことができます。メディアのデバイスとデバイス名の詳細については、第 52 章を参照してください。

ファイルシステムの復元

バックアップを実行すると、ファイルやディレクトリは、それらが含まれるファイルシステムからの相対的な位置に保存されます。ファイルとディレクトリを復元するときは、`ufsrestore` コマンドが現在の作業ディレクトリにファイル階層を作成し直します。

たとえば、`/export/doc/books` ディレクトリ (`/export` はファイルシステム) からバックアップされたファイルは、`/export` からの相対的な位置に保存されます。つまり、`books` ディレクトリ内の `book1` ファイルは、テープ上で `./doc/books/book1` として保存されます。後で、`./doc/books/book1` ファイルを `/var/tmp` ディレクトリに復元する場合、そのファイルは `/var/tmp/doc/books/book1` に復元されます。

個別のファイルやディレクトリを復元するときには、`/var/tmp` などの一時的な場所に復元するのが賢明です。ファイルやディレクトリを確認したら、それを適当な位置に移動させてもかまいません。個別のファイルやディレクトリはそれぞれ元の位置に復元できます。その場合には、新しいファイルをバックアップテープからの古いバージョンで上書きしないかどうか確かめてください。

他のユーザーとの重複を防ぐために、`/var/tmp/restore` などのサブディレクトリを作成し、そのディレクトリに移動して、ファイルを復元することをお勧めします。

階層を復元する場合、ファイルを配置するファイルシステム内の一時ディレクトリにファイルを復元する必要があります。復元後に、`mv` コマンドを使用して階層全体を適切な場所に移動します。

注 – 一時的な場合でも、`/tmp` ディレクトリにファイルを復元してはなりません。一般的に `/tmp` ディレクトリは、TMPFS ファイルシステムとしてマウントされており、TMPFS は ACL などの UFS ファイルシステム属性をサポートしていません。

▼ 使用するテープを決定する方法

1. 復元するファイルを最後に変更したのはいつごろだったかをユーザーにたずねます。
2. バックアップ計画を参照し、該当するファイルまたはファイルシステムを含む最後のバックアップの日付を調べます。

ファイルの最新バージョンを検索するには、増分バックアップファイルを最高レベルから最低レベルへ、最新バージョンから最古バージョンへ逆方向に調べます。

3. (省略可能) オンラインアーカイブファイルを保持している場合、使用に適したメディアを判別します。

```
# ufsrestore ta archive-name ./path/filename ./path/filename
```

<code>t</code>	テープ上の各ファイルを表示する。
<code>a</code>	テープではなくオンラインアーカイブファイルから内容一覧を読み取る。
<code>archive-name</code>	オンラインアーカイブファイル名を指定する。
<code>./path/filename</code>	オンラインアーカイブ上で検索するファイル名を指定する。コマンドが成功した場合は、 <code>ufsrestore</code> は <code>i</code> ノード番号とファイル名を出力する。成功しなかった場合は、エラーメッセージを出力する。

詳細については、`ufsrestore (1M)` のマニュアルページを参照してください。

4. 復元するファイルを含むメディアをドライブに挿入し、適切なメディアであることを確認します。

```
# ufsrestore tf /dev/rmt/n ./path/filename ./path/filename
```


ファイル名には必ず完全パス名を使用してください。ファイルがバックアップに入っていれば、その名前と i ノード番号が表示されます。ファイルがバックアップに入っていない場合、ファイルがそのボリュームに入っていないことを示すメッセージが表示されます。

5. (省略可能) 同じテープに複数のダンプファイルが入っている場合は、使用するダンプの位置までテープを移動します。

```
# ufsrestore tfs /dev/rmt/n tape-number
```

例 — 使用するテープを決定する

次の例は、オンラインアーカイブに `/etc/passwd` ファイルが存在するかどうかを確認する方法を示します。

```
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

次の例は、バックアップテープに `/etc/passwd` が存在するかどうかを確認する方法を示します。

```
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

▼ 対話式でファイルを復元する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. (省略可能) 安全性のため、書き込み保護を設定します。
3. ボリューム 1 のテープをテープドライブに挿入します。
4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5. 対話式で復元を開始します。

```
# ufsrestore if /dev/rmt/n
```

情報を伝えるためのメッセージと `ufsrestore>` プロンプトが表示されます。

6. 復元するファイルのリストを作成します。

- a. ディレクトリの内容を表示します。

```
ufsrestore> ls directory
```

- b. ディレクトリを変更します。

```
ufsrestore> cd directory-name
```

- c. 復元するファイルとディレクトリのリストを作成します。

```
ufsrestore> add filename filename
```

- d. (省略可能) 必要に応じ、復元するファイルのリストからディレクトリまたはファイル名を削除します。

```
ufsrestore> delete filename
```

7. (省略可能) 復元処理中にファイル名を表示します。

```
ufsrestore> verbose
```

8. ファイルを復元します。

```
ufsrestore> extract
```

どのボリューム番号を使用するかを指定するプロンプトが表示されます。

9. ボリューム番号を入力して、**Return** キーを押します。ボリュームが**1**つしかない場合には、**1**を入力して**Return** キーを押します。

```
Specify next volume #: 1
```

リスト内のファイルとディレクトリが抽出され、現在の作業ディレクトリに復元されます。

10. 現在の作業ディレクトリのモードを変更したくない場合は、**set owner/mode** プロンプトが表示されたときに**n**を入力します。

```
set owner/mode for `.'? [yn] n
```

ufsrestore コマンドによる最後の整理が完了すると、ufsrestore のプロンプトが表示されます。

11. ufsrestore プログラムを終了します。

```
ufsrestore> quit
```

シェルプロンプトが表示されます。

12. 復元されたファイルを確認します。

- a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

- b. リストをチェックして、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

13. ファイルを適切なディレクトリに移動します。

例 — 対話式でファイルを復元する

次の例では、`/etc/passwd` ファイルと `/etc/shadow` ファイルがバックアップテープから復元されます。

```

# cd /var/tmp
# ufsrestore if /dev/rmt/0
ufsrestore> ls
.:
.cpr_config  etc/          lost+found/  sbin/        usr/
TT_DB/       export/       mnt/         sccs/        var/
b/           home/         net/         share/       vol/
bin          kernel/       opt/         shared/      ws/
dev/         lib           platform/    src/         xfn/
devices/     license/      proc/        tmp/
ufsrestore> cd etc
ufsrestore> add passwd shadow
ufsrestore> verbose
verbose mode on
ufsrestore> extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./etc/shadow
extract file ./etc/passwd
Add links
Set directory mode, owner, and times.
set owner/mode for `.'? [yn] n
ufsrestore> quit
#

```

▼ 例 — 特定のファイルを復元する

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. (省略可能) 安全性のため、書き込み保護を設定します。
3. ボリューム 1 のテープをテープドライブに挿入します。
4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5. ファイルを復元します。

```
# ufsrestore xvf /dev/rmt/n filename
```

x	<i>filename</i> 引数内に指定されたファイルまたはディレクトリをコピーするように ufsrestore コマンドに指定する。
v	復元処理中にファイル名を表示する。
f /dev/rmt/n	テープデバイス名を識別する。

filename

次のように空白で区切られた 1 つ以上の個別のファイルまたはディレクトリの名前を指定する。例:

```
./export/home/user1/mail  
./export/home/user2/mail
```

6. ファイルが入っているボリューム番号を入力して、**Return** キーを押します。

```
Specify next volume #: 1
```

ファイルは現在の作業ディレクトリに復元されます。

7. 現在のディレクトリのモードを変更しない場合は、**set owner/mode** プロンプトが表示されたときに **n** と入力して **Return** キーを押します。

```
set owner/mode for './?' [yn] n
```

8. 復元されたファイルを確認します。

- a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

- b. リストをチェックして、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

9. ファイルを適切なディレクトリに移動します。

例 — 特定のファイルを復元する

次の例では、`passwd` と `shadow` ファイルが `/var/tmp` ディレクトリに復元されます。

```
# cd /var/tmp  
# ufsrestore xvf /dev/rmt/0 ./etc/passwd ./etc/shadow  
Verify volume and initialize maps  
Media block size is 126  
Dump date: Wed Dec 12 10:54:45 2001  
Dumped from: the epoch  
Level 0 dump of / on earth:/dev/dsk/c0t1d0s0  
Label: none  
Extract directories from tape  
Initialize symbol table.  
Make node ./etc  
Extract requested files  
You have not read any volumes yet.  
Unless you know which volume your file(s) are on you should start  
with the last volume and work towards the first.  
Specify next volume #: 1  
extract file ./etc/passwd  
extract file ./etc/shadow  
Add links
```

```
Set directory mode, owner, and times.
set owner/mode for `.'? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

例 — リモートのテープデバイスからファイルを復元する

ufsrestore コマンドを使用する際、テープデバイス名の前に *remote-host* : を追加することにより、ファイルをリモートドライブから復元できます。

次の例では、システム *venus* 上のリモートテープドライブ */dev/rmt/0* を使用してファイルを復元します。

```
# ufsrestore xf venus:/dev/rmt/0 ./etc/hosts
```

▼ ファイルシステム全体を復元する方法

時には、全面的に復元しなければならないほどファイルシステムが破壊される場合があります。一般的な例として、ディスク障害が発生した場合は、ハードウェアを交換してからファイルシステムを全面的に復元する必要があります。ディスクの交換方法については、第 34 章または第 35 章を参照してください。/export/home などのファイルシステムを全面的に復元するには、時間がかかります。ファイルシステムを一貫性のある方法でバックアップしていれば、最後の増分バックアップ時の状態に復元することができます。

注 - ルート (/) または /usr ファイルシステムの復元には、この手順は使用できません。これらのファイルシステムを復元する方法については、728 ページの「ルート (/) と /usr を復元する方法」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じ、ファイルシステムをマウント解除します。

```
# umount /dev/rdisk/device-name
```

3. ファイルシステムを新規作成します。

```
# newfs /dev/rdisk/device-name
```

raw デバイス上に新しいファイルシステムを構築するかどうかを確認するプロンプトが表示されます。意図しないファイルシステムを間違えて損失してしまわないように、*device-name* が正しいことを確認します。

詳細については、newfs (1M) のマニュアルページを参照。

4. 新しいファイルシステムを作成しなければならないかどうかを確認します。

```
newfs: construct a new file system /dev/rdsk/cwtxdysz:(y/n)? y
```

新しいファイルシステムが作成されます。

5. 新しいファイルシステムを一時的なマウントポイントにマウントします。

```
# mount /dev/dsk/device-name /mnt
```

6. マウントポイントのディレクトリに移動します。

```
# cd /mnt
```

7. (省略可能) 安全性のため、書き込み保護を設定します。

8. レベル 0 テープの第 1 ボリュームをテープドライブに挿入します。

9. ファイルを復元します。

```
# ufsrestore rvf /dev/rmt/n
```

レベル 0 ダンプが復元されます。ダンプの実行に複数のテープが必要な場合は、番号の順にテープをロードするようにプロンプトが表示されます。

10. テープを外し、次のレベルのテープをドライブにロードします。

テープの復元は必ず 0 から始め、一番高いレベルまで続けてください。

11. ダンプレベルごとに、一番低いレベルから一番高いレベルまで、手順 8 から手順 10 までの操作を繰り返します。

12. ファイルシステムが復元されていることを確認します。

```
# ls
```

13. `restoresymtable` ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために `ufsrestore` コマンドが作成し、使用した `restoresymtable` ファイルを削除します。

14. 別のディレクトリに移動します。

```
# cd /
```

15. 新しく復元されたファイルシステムをマウント解除します。

```
# umount /mnt
```

16. 最後のテープを取り出し、書き込み保護されていない新しいテープをテープドライブに挿入します。

17. 新しく復元されたファイルシステムのレベル 0 のバックアップを作成します。

```
# ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
```

レベル 0 のバックアップが実行されます。 `ufsrestore` はファイルの位置を移動し、i ノード割り当てを変更するので、新しく作成されたファイルシステムの

バックアップは、必ずすぐに作成するようにしてください。

18. 復元されたファイルシステムをマウントします。

```
# mount /dev/dsk/device-name mount-point
```

復元されたファイルシステムがマウントされ、使用できるようになります。

19. 復元およびマウントされたファイルシステムが使用できることを確認します。

```
# ls /mount-point
```

例 — ファイルシステム全体を復元する

次の例は、/export/home ファイルシステムの復元方法を示します。

```
# umount /export/home
# newfs /dev/rdisk/c0t3d0s7
newfs: construct a new file system /dev/rdisk/c0t3d0s7: (y/n)? y
/dev/rdisk/c0t3d0s7: 410400 sectors in 270 cylinders of 19 tracks,
80 sectors
200.4MB in 17 cyl groups (16 c/g, 11.88MB/g, 5696 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 24432, 48832, 73232, 97632, 122032, 146432, 170832, 195232, 219632,
244032, 268432, 292832, 317232, 341632, 366032, 390432,
# mount /dev/dsk/c0t3d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump date: Sat Jul 14 08:49:33 2001
Dumped from: the epoch
Level 0 dump of /export/home on earth:/dev/dsk/c0t3d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./kryten
Make node ./kryten/letters
Make node ./kryten/reports
Extract new leaves.
Check pointing the restore
extract file ./kryten/.cshrc
extract file ./kryten/.login
extract file ./kryten/b
extract file ./kryten/memos
extract file ./kryten/letters/b
extract file ./kryten/letters/letter1
extract file ./kryten/letters/letter2
extract file ./kryten/letters/letter3
extract file ./kryten/reports/reportA
extract file ./kryten/reports/reportB
```

```

extract file ./kryten/reports/reportC
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# ls
# rm restoresymtable
# cd /
# umount /mnt
# ufsdump 0ucf /dev/rmt/0 /export/home
.
.
.
# mount /dev/dsk/c0t3d0s7 /export/home
# ls /export/home

```

▼ ルート (/) と /usr を復元する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 新しいシステムディスクを、ルート (/) と /usr ファイルシステムが復元されるシステムに追加します。
システムディスクの追加方法については、第 34 章または第 35 章を参照してください。
3. 新しいファイルシステムを一時的なマウントポイントにマウントします。

```
# mount /dev/dsk/device-name /mnt
```
4. /mnt ディレクトリに移動します。

```
# cd /mnt
```
5. (省略可能) 安全性のため、書き込み保護を設定します。
6. ルートファイルシステムを復元します。

```
# ufsrestore rvf /dev/rmt/n
```


レベル 0 のテープが復元されます。
7. テープを外し、次のレベルのテープをドライブにロードします。
テープの復元は必ずダンプレベル 0 から始め、最も低いレベルから最も高いレベルまで続けてください。
8. 必要に応じ、復元を続行します。

```
# ufsrestore rvf /dev/rmt/n
```


次のレベルのテープが復元されます。
9. テープを追加するたびに、手順 7 と手順 8 を繰り返します。

10. ファイルシステムが復元されていることを確認します。

```
# ls
```

11. `restoresymtable` ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために `ufsrestore` コマンドが作成し、使用した `restoresymtable` ファイルを削除します。

12. ルート (`/`) ディレクトリに変更します。

```
# cd /
```

13. 新しく作成されたファイルシステムをマウント解除します。

```
# umount /mnt
```

14. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdsk/device-name
```

復元されたファイルシステムの整合性がチェックされます。

15. ルートパーティションにブートブロックを作成します。

```
# installboot /usr/platform/`uname-i`/lib/fs/ufs/bootblk  
/dev/rdsk/device-name
```

詳細については、`installboot` (1M) のマニュアルページを参照してください。

SPARC システム上で `installboot` コマンドを使用する方法については、730 ページの「SPARC: 例 — ルート (`/`) ファイルシステムを復元する」を参照してください。x86 システム上で `installboot` コマンドを使用する方法については、730 ページの「x86: 例 — ルート (`/`) ファイルシステムを復元する」を参照してください。

16. 新しいテープをテープドライブに挿入します。

17. 新しいファイルシステムのバックアップを作成します。

```
# ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
```

レベル 0 のバックアップが実行されます。`ufsrestore` はファイルの位置を移動し、`i` ノード割り当てを変更するので、新しく作成されたファイルシステムのバックアップは、必ずすぐに作成するようにしてください。

18. 必要に応じ、`/usr` ファイルシステムについて手順 5 から 16 を繰り返します。

19. システムをリブートします。

```
# init 6
```

システムがリブートされます。

SPARC: 例 — ルート (/) ファイルシステムを復元する

次の例は、SPARC システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、システムはローカルの CD またはネットワークから起動するものとします。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

x86: 例 — ルート (/) ファイルシステムを復元する

次の例は、x86 システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、システムはローカルの CD またはネットワークから起動するものとします。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i`/lib/fs/
ufs/bootblk /dev/rdisk/c0t3d0s2
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

第 50 章

UFS バックアップおよび復元コマンド (参照情報)

この章では、`ufsdump` コマンドと `ufsrestore` コマンドの参照情報を示します。

この章の内容は次のとおりです。

- 731 ページの「`ufsdump` コマンドの機能」
- 736 ページの「`ufsdump` コマンドのオプションと引数」
- 739 ページの「`ufsdump` とセキュリティに関する注意事項」
- 739 ページの「`ufsrestore` コマンドのオプションと引数」

ufsdump コマンドの機能

`ufsdump` コマンドは、ファイルシステムのバックアップ作成時に 2 つのパスを作成します。最初のパスでは、このコマンドは `raw` デバイスファイル内でファイルシステムを走査し、メモリー内にディレクトリとファイルのテーブルを作成します。次に、そのテーブルをバックアップメディアに書き込みます。2 つ目のパスでは、`ufsdump` は `i` ノードに番号順にアクセスし、ファイルの内容を読み込んでメディアに書き込みます。

デバイス特性の判断

`ufsdump` コマンドに必要なことは、適切なブロックの大きさを認識することと、どのようにしてメディアの終わりを検出するかということです。

メディアの終わりの検出

ufsdump コマンドは、一連の固定長レコードを書き込みます。ufsdump コマンドは、レコードの一部にしか書き込まれていないという通知を受け取ると、メディアの物理的な終わりに達したものと判断します。この方法は、ほとんどのデバイスに有効です。部分的なレコードしか書き込まれなかったことをデバイスが ufsdump に通知できない場合、ufsdump が別のレコードの書き込みを試みると、メディアエラーが発生します。

注 - DAT デバイスと 8mm テープデバイスでは、メディアの終わりが検出されます。カートリッジテープデバイスと 1/2 インチテープデバイスでは、メディアの終わりは検出されません。

ufsdump は、ほとんどのデバイスのメディアの終わりを自動的に検出します。したがって、通常は -c、-d、-s、-t オプションを使用しなくても、複数のボリュームのバックアップを実行できます。

メディアの終わりを検出するオプションを使用する必要があるのは、デバイスがメディアの終わりを検出する方法を ufsdump コマンドで判断できない場合、または restore コマンドを使って SunOS 4.1 システム上でファイルを復元しようとしている場合に限られます。restore コマンドとの互換性を確保するため、サイズオプションを使用すると、従来どおり、現在のテープやフロッピーディスクの終わりに達する前に、ufsdump を次のテープやフロッピーディスクに強制的に進ませることができます。

ufsdump データのコピー

ufsdump コマンドは、raw ディスクスライスからデータのみをコピーします。ファイルシステムがまだ有効であれば、メモリーバッファ内のデータがコピーされていない可能性があります。ufsdump によるバックアップでは、空きブロックはコピーされず、ディスクスライスのイメージも作成されません。シンボリックリンクが他のスライス上のファイルを指す場合は、リンク自体がコピーされます。

/etc/dumpdates ファイルの役割

ufsdump コマンドを -u オプション付きで使用すると、/etc/dumpdates というファイルを管理し、更新できます。/etc/dumpdates ファイル内の各行は、バックアップが作成されたファイルシステム、前回のバックアップレベル、バックアップの日付、曜日、および時刻を表しています。たとえば、次のようになります。

```
/dev/rdsk/c0t0d0s7          0 Mon Dec 10 16:26:10 2001
/dev/rdsk/c0t0d0s7          9 Tue Dec 11 16:45:14 2001
/dev/rdsk/c0t0d0s7          9 Wed Dec 12 16:54:47 2001
```

増分バックアップの実行時に、`ufsdump` コマンドは `/etc/dumpdates` ファイルをチェックして、下のレベルの最後のバックアップ日付を調べます。次に、下のレベルのバックアップ以降に更新されたすべてのファイルをメディアにコピーします。バックアップが完了すると、完了したばかりのバックアップを記述する新しい情報行によって、そのレベルの最後のバックアップの情報行が置き換えられます。

`/etc/dumpdates` ファイルを使用して、バックアップが実行中であるかどうかを検査してください。機器に問題が発生している場合は、この検査が特に重要です。機器の障害が原因でバックアップを完了できないと、そのバックアップは `/etc/dumpdates` ファイルに記録されません。

ディスク全体を復元する必要がある場合は、`/etc/dumpdates` ファイル内で最後のバックアップの日付とレベルをチェックできるので、ファイルシステム全体を復元するために必要なファイルを判断できます。

注 - `/etc/dumpdates` ファイルは編集可能なテキストファイルですが、編集するかどうかはユーザーの判断によります。ファイルに変更を加えた結果、アーカイブテープと一致なくなると、必要なテープ (またはファイル) がどれであるかわからなくなることがあります。

バックアップデバイス (*dump-file*) 引数

dump-file 引数 (`-f` オプションで使用) では、次のいずれかのバックアップ先を指定します。

- ローカルのテープドライブかフロッピーディスクドライブ
- リモートのテープドライブかフロッピーディスクドライブ
- 標準出力

この引数は、バックアップ先がデフォルトのローカルテープドライブ `/dev/rmt/0` でないときに使用します。 `-f` オプションを使用する場合は、*dump-file* の値を指定しなければなりません。

注 - また、*dump-file* 引数では、ローカルディスクまたはリモートディスク上のファイルを指すこともできるので、誤用するとファイルシステムがいっぱいになる可能性があります。

ローカルのテープドライブまたはフロッピーディスクドライブ

通常、*dump-file* 引数には、テープデバイスかフロッピーディスク用の `raw` デバイスファイルを指定します。`ufsdump` コマンドは、出力デバイスへの書き込み時にバックアップファイルを1つ作成しますが、このファイルは複数のテープやフロッピーディスクにまたがってもかまいません。

デバイスの省略形を使用して、システム上のテープデバイスかフロッピーディスクデバイスを指定します。第 1 のデバイスは常に 0 です。たとえば、SCSI テープコントローラが 1 つと、中密度の形式を使用する QIC-24 テープドライブが 1 つある場合は、次のデバイス名を使用します。

```
/dev/rmt/0m
```

テープデバイス名を指定するときは、名前の末尾に文字 *n* を付けて、バックアップの完了後にテープドライブを巻き戻さないように指定することもできます。たとえば、次のようになります。

```
/dev/rmt/0mn
```

テープに複数のファイルを格納する場合は、`no-rewind` オプションを使用します。バックアップ中に領域を使い果たすと、`ufsdump` コマンドから新しいテープの挿入を促すプロンプトが表示されるまで、テープは巻き戻されません。デバイスの命名規則の詳細については、772 ページの「バックアップデバイス名」を参照してください。

リモートのテープドライブまたはフロッピーディスクドライブ

`host:device` という形式で、リモートのテープデバイスまたはフロッピーディスクを指定します。ローカルシステム上の `root` ユーザーがリモートシステムへのアクセス権を持っている場合、`ufsdump` コマンドはリモートデバイスに書き込みます。通常、`root` ユーザーとして `ufsdump` コマンドを実行するのであれば、ローカルシステム名をリモートシステムの `/.rhosts` ファイルに記述しておく必要があります。デバイスを `user@host:device` と指定した場合、`ufsdump` コマンドは指定されたユーザーでリモートシステム上のデバイスへのアクセスを試みます。この場合、指定されたユーザーの名前が、リモートシステム上の `/.rhosts` ファイル中に含まれている必要があります。

デバイスには、`ufsdump` コマンドを実行するシステムではなく、そのデバイスが存在するシステムのオペレーティングシステムに合った命名規則を使用してください。デバイスが SunOS の旧バージョン (4.1.1 など) を実行するシステム上にある場合は、SunOS 4.1 でのデバイス名 (`/dev/rst0` など) を使用します。システムが Solaris ソフトウェアを実行中の場合は、SunOS 5.9 でのデバイス名 (`/dev/rmt/0` など) を使用します。

ufsdump コマンドで標準出力を使用する

`dump-file` 引数としてダッシュ (-) を指定すると、`ufsdump` コマンドは標準出力に書き込みます。

注 - *dump-file* 引数として標準出力を指定すると、*-v* オプション (検査) は機能しません。

`ufsdump` コマンドを使用して標準出力に書き込み、`ufsrestore` コマンドを使用して標準入力から読み込むと、パイプライン内でファイルシステムをコピーできます。次のようになります。

```
# ufsdump 0f - /dev/rdsk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

バックアップを作成するファイルを指定する

コマンド行の最後の引数として、バックアップするファイル (*filenames*) を必ず指定してください。この引数は、バックアップのコピー元または内容を指定します。

ファイルシステムの場合、次のように raw デバイスファイルを指定します。

```
/dev/rdsk/c0t0d0s6
```

ファイルシステムは、そのエントリが `/etc/vfstab` ファイルに入っていれば、マウントポイントディレクトリ (`/export/home` など) を使用して指定できます。

デバイスの命名規則の詳細については、772 ページの「バックアップデバイス名」を参照してください。

個々のファイルやディレクトリごとに、1 つまたは複数の名前を空白で区切って入力します。

注 - `ufsdump` コマンドを使用して (ファイルシステム全体ではなく) 1 つまたは複数のディレクトリやファイルのバックアップを作成するときには、レベル 0 のバックアップが実行されます。増分バックアップは適用されません。

テープの性質を指定する

テープの性質を指定しなければ、`ufsdump` コマンドはデフォルト設定を使用します。テープカートリッジ (*-c*)、密度 (*-d*)、サイズ (*-s*)、トラック数 (*-t*) を指定できます。オプションの順序とその引数の順番が一致していれば、オプションはいくつでも指定できます。

`ufsdump` の制限

次に、`ufsdump` コマンドでは実行できない操作を示します。

- ファイルシステムのバックアップに必要なテープやフロッピーディスクの数を自動的に計算する。仮実行 (ドライラン) モード (S オプション) を使用すると、実際にファイルシステムをバックアップする前に必要な容量を判断できる。
- アクティブなファイルシステムをバックアップするときの問題を最小限度に抑えるために、組み込みエラーチェック機能を提供する。
- サーバーからリモートにマウントされたファイルをバックアップする。サーバー上のファイルのバックアップは、そのサーバー上で実行しなければならない。ユーザーがサーバー上で所有するファイル上で `ufsdump` コマンドを実行するアクセス権は拒否される。

ufsdump コマンドのオプションと引数

この節では、`ufsdump` コマンドのオプションと引数について詳しく説明します。`ufsdump` コマンドの構文を、次に示します。

```
/usr/sbin/ufsdump options arguments filenames
```

<i>options</i>	1文字のオプション名からなる1つの文字列。
<i>arguments</i>	オプションの引数を指定する。複数の文字列も可。オプション文字と引数は、同じ順序で並べる必要がある。
<i>filenames</i>	バックアップするファイルを指定する。これらの引数は常に最後に指定する。

ufsdump のデフォルトオプション

オプションを指定せずに `ufsdump` コマンドを実行する場合は、次の構文を使用します。

```
# ufsdump filenames
```

`ufsdump` コマンドでは、デフォルトで次のオプションと引数が使用されます。

```
ufsdump 9uf /dev/rmt/0 filenames
```

これらのオプションでは、デフォルトのテープドライブ上にその推奨密度でレベル9の増分バックアップが作成されます。

ufsdump コマンドのオプション

次の表に、`ufsdump` コマンドのオプションを示します。

表 50-1 ufsdump コマンドのオプション

オプション	説明
0-9	ダンプレベル。レベル 0 は、 <i>filenames</i> で指定したファイルシステム全体の完全バックアップ用。レベル 1～9 は、最後の下位バックアップ以降に変更があったファイルの増分バックアップ用。
a <i>archive-file</i>	アーカイブファイル。指定したディスク上のファイルにバックアップ用の内容一覧を格納 (アーカイブ) する。このファイルは、 <i>ufsrestore</i> コマンドでしか認識できず、復元すべきファイルがバックアップファイル内にあるかどうかと、もしあればどのメディアボリュームに入っているかを判断するために使用される。
b <i>factor</i>	ブロック係数。1 処理ごとにテープに書き込まれる 512 バイトのブロック数を指定する。
c	カートリッジ。カートリッジテープにバックアップを作成する。メディアの終わりの検出を適用するときは、このオプションでブロックサイズを 126 に設定する。
d <i>bpi</i>	テープ密度。このオプションは、 <i>ufsdump</i> コマンドでメディアの終わりを検出できない場合にのみ使用する。
D	フロッピーディスク。フロッピーディスクにバックアップを作成する。
f <i>dump-file</i>	ダンプファイル。デフォルトデバイスではなく <i>dump-file</i> で指定したコピー先にファイルを書き込む。ファイルを <i>user@system:device</i> として指定すると、 <i>ufsdump</i> コマンドは指定されたユーザーとしてリモートシステム上で実行しようとする。ローカルシステム上でこのコマンドを実行してリモートシステムにアクセスするため、指定されたユーザーはリモートシステム上に <i>.rhosts</i> ファイルを保持する必要がある。
l	自動ロード。このオプションは、オートロード (スタックローダ) テープドライブがある場合に使用する。テープの終わりに達すると、このオプションはドライブをオフラインにして、テープドライブの準備ができるまで 2 分間待つ。2 分以内にドライブの準備ができると、続行する。2 分経過してもドライブの準備ができていない場合、オペレータに別のテープをロードするように促すプロンプトが表示される。
n	通知。介入が必要になると、 <i>sys</i> グループのユーザー全員の端末にメッセージを送る。
o	オフライン。テープやフロッピーディスクの処理が終わると、ドライブをオフラインにして巻き戻し (テープの場合)、可能であればメディアをはずす (たとえば、フロッピーディスクを取り出ししたり、8mm の自動ロードテープをはずす)。

表 50-1 ufsdump コマンドのオプション (続き)

オプション	説明
s <i>size</i>	サイズ。テープの場合はフィート数、フロッピーディスクの場合は 1024 バイトのブロック数を指定する。このオプションは、 <code>ufsdump</code> コマンドでメディアの終わりを検出できない場合にのみ使用する。
S	バックアップのサイズを予想する。バックアップを実際に行わずに必要な容量を判断し、バックアップの予想バイト数を示す数値を 1 つ出力する。
t <i>tracks</i>	トラック数。1/4 インチカートリッジテープのトラック数を指定する。このオプションは、 <code>ufsdump</code> コマンドでメディアの終わりを検出できない場合にのみ使用する。
u	ダンプレコードをアップデートする。ファイルシステムの完全バックアップを実行する場合には、 <code>/etc/dumpdates</code> ファイルにエントリを追加する。エントリは、ファイルシステムのディスクスライスのデバイス名、ダンプレベル (0 ~ 9)、および日付を示す。u オプションを使用しないときや、個々のファイルかディレクトリのバックアップを作成するときは、レコードは書き込まれない。バックアップのレコードがすでに同じレベルに存在する場合は、それが置き換えられる。
v	検査。各テープまたはフロッピーディスクへの書き込み後に、ソースファイルシステムと対照してメディアの内容を検査する。不整合が検出されると、オペレータに新しいメディアのマウントを促すプロンプトを表示してプロセスを繰り返す。ファイルシステム内で操作が実行されると、 <code>ufsdump</code> コマンドが不整合を報告するため、このオプションはマウント解除されたファイルシステムにのみ使用する必要がある。
w	警告。 <code>/etc/dumpdates</code> ファイルに表示されるファイルシステムのうち、特定の日にバックアップされていないファイルシステムを表示する。このオプションを使用すると、他のすべてのオプションは無視される。
W	強調表示付きの警告。 <code>/etc/dumpdates</code> 内のすべてのファイルシステムを表示し、特定の日にバックアップされていないファイルシステムを強調表示する。このオプションを使用すると、他のすべてのオプションは無視される。

注 - `/etc/vfstab` ファイルには、ファイルシステムのバックアップ頻度に関する情報は含まれません。

ufsdump とセキュリティに関する注意事項

セキュリティ保護を適用するには、次の操作を実行する必要があります。

- `ufsdump` コマンドの実行には、`root` アクセス権を必要とする。
- 集中バックアップを実行する場合は、クライアント上とサーバー上の `/.rhost` ファイルから `root` アクセスのエントリを削除する。

セキュリティに関する一般的な情報は、『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

ufsrestore コマンドのオプションと引数

`ufsrestore` コマンドの構文を、次に示します。

```
ufsrestore options arguments filenames
```

<i>options</i>	1 文字のオプション名からなる 1 つの文字列。i、r、R、t、x から 1 つだけ選択する。表 50-3 に示す追加オプションは、省略可能。
<i>arguments</i>	オプションに対応する引数。オプション文字と引数は、同じ順序で並べる必要がある。
<i>filenames</i>	復元するファイルを、x または t オプションの引数として指定する。これらの引数は常に最後に指定する。

次の表に示す `ufsrestore` コマンドオプションのうち、1 つだけを指定する必要があります。

表 50-2 ufsrestore コマンドに必要なオプション

オプション	説明
i	対話式。ufsrestore コマンドを対話モードで実行する。このモードでは、限られたシェルコマンドセットを実行してメディアの内容を表示し、復元するファイルやディレクトリを個別に選択できる。対話式コマンドのリストについては、表 50-4 を参照。
r	再帰的。メディアの内容全体を現在の作業ディレクトリ (ファイルシステムの最上位レベル) に復元する。完全ダンプ (restoresymtable など) の最上部に増分ダンプを復元するための情報も含まれる。ファイルシステムを完全に復元するには、このオプションを使用して完全 (レベル 0) ダンプを復元してから、各増分ダンプを復元する。このオプションは新しい (newfs コマンドで作成したばかりの) ファイルシステム用だが、バックアップメディアにないファイルが保存される。
R	復元の再開。復元を再開するボリュームをたずねるプロンプトを表示し、チェックポイントから再開する。完全復元 (r オプション) が中断された後は、このオプションを指定して ufsrestore コマンドを実行し直す。
x [<i>filenames</i>]	抽出。filenames 引数で指定したファイルを選択的に復元する。filenames では、ファイルとディレクトリのリストを指定できる。h オプションも指定しなければ、指定したディレクトリの下の子ファイルがすべて復元される。filenames を省略するか、ルートディレクトリを表す「.」を入力すると、メディアのすべてのボリューム上 (または標準入力から) のすべてのファイルが復元される。既存のファイルは上書きされ、警告が表示される。
t [<i>filenames</i>]	内容一覧。filenames 引数で指定したファイルがメディアと対照してチェックされる。ファイルごとに、完全ファイル名と i ノード番号 (ファイルが見つかった場合) が表示されるか、ファイルが「ボリューム」上にないことを示す (複数のボリュームダンプのボリュームを意味する)。filenames 引数を入力しなければ、メディアのすべてのボリューム上のファイルが表示される (どのボリュームにファイルが入っているかは区別されない)。h オプションも指定すると、内容ではなく filenames で指定したディレクトリファイルのみがチェックされ表示される。内容一覧は、メディアの最初のボリューム、またはアーカイブファイル (a オプションを使用した場合) から読み込まれる。このオプションは x、r オプションと一緒に使用できない。

次の表に、ufsrestore の追加オプションを示します。

表 50-3 ufsrestore コマンドの追加オプション

オプション	説明
a <i>archive-file</i> [<i>filenames</i>]	ダンプの内容一覧は、メディア (最初のボリューム) ではなく、指定した <i>archive-file</i> からダンプの内容一覧が取り出される。このオプションを t、i、または x オプションと組み合わせて使用すると、メディアをマウントしなくても、ファイルがメディアに存在するかどうかを確認できる。x オプションと対話型抽出オプションと組み合わせて使用すると、ファイルを抽出する前に適切なボリュームのマウントを促すプロンプトが表示される。
b <i>factor</i>	ブロック係数。1 回の処理でテープから読み込む 512 バイトのブロック数を指定する。デフォルトでは、ufsrestore コマンドはテープへの書き込みに使用したブロックサイズの使用を試みる。
d	デバッグ。デバッグメッセージ機能をオンにする。
f <i>backup-file</i>	バックアップファイル。ファイルは、デフォルトのデバイスファイル /dev/rmt/0m ではなく <i>backup-file</i> で指定したソースから読み込まれる。f オプションを使用する場合は、 <i>backup-file</i> の値を指定する必要がある。 <i>backup-file</i> が <i>system:device</i> 形式であれば、ufsrestore はリモートデバイスから読み込む。 <i>backup-file</i> 引数を使用すると、ローカルディスクやリモートディスク上のファイルも指定できる。 <i>backup-file</i> が "-" であれば、ファイルは標準入力から読み込まれる。
h	ディレクトリの展開をオフにする。指定したディレクトリファイルのみが抽出または表示される。
m	指定したファイルが、バックアップ階層内の位置に関係なくディスク上の現在のディレクトリに復元され、i ノード番号を使用して名前が変更される。たとえば、現在の作業ディレクトリが /files であれば、i ノード番号が 42 のバックアップ ./dready/fcs/test 内のファイルは、/files/42 として復元される。このオプションは、少数のファイルを抽出する場合のみ有用である。
s <i>n</i>	最初のボリュームメディア上の <i>n</i> 番目のバックアップファイルまでスキップする。このオプションは、1 本のテープに複数のバックアップを入れるときに便利である。
v	詳細表示。各ファイルが復元されるたびに、その名前と i ノード番号が表示される。
y	メディアの読み込みエラーが発生しても処理を続行する。処理を停止して続行するかどうかを選択するプロンプトを表示せずに、不良ブロックをスキップしようとする。このオプションによって、コマンドは肯定の応答とみなすよう命令される。

次の表に、ufsrestore の対話式コマンドを示します。

表 50-4 対話式の復元コマンド

オプション	説明
ls [<i>directory-name</i>]	現在のディレクトリまたは指定したディレクトリの内容を表示する。ディレクトリは接尾辞 / 付きで表示される。現在のリスト内で復元 (抽出) されるエントリは接頭辞 * 付きで表示される。詳細オプションを使用すると、i ノード番号が表示される。
cd <i>directory-name</i>	バックアップ階層内の指定したディレクトリに変更する。
add [<i>filename</i>]	現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストに追加する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のすべてのファイルがリストに追加される。ディレクトリに復元したいすべてのファイルが 1 つのバックアップテープやフロッピーディスクに入っていないことがある。すべてのファイルの最新バージョンを抽出するには、さまざまなレベルの複数のバックアップから復元しなければならない場合がある。
delete [<i>filename</i>]	現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストから削除する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のファイルがすべて削除される。ファイルとディレクトリは、構築中の抽出リストからのみ削除される。メディアまたはファイルシステムからは削除されない。
extract	リスト内のファイルを抽出し、ディスク上の現在の作業ディレクトリからの相対パスで指定される位置に復元する。単一ボリュームのバックアップに関して、ボリューム番号を確認するプロンプトが表示されたら、1 を指定する。複数テープや複数フロッピーディスクから少数のファイルを復元する場合は、最後のテープまたはフロッピーディスクから始める。
help	対話式で使用できるコマンドのリストが表示される。
pwd	バックアップ階層内の現在の作業ディレクトリのパス名が表示される。
q	それ以上ファイルを復元しないで対話モードを終了する。
setmodes	バックアップ元となったファイルシステムのルートディレクトリのモードに合わせて、復元するファイルのモードを設定できる。set owner/mode for '.' [yn]? というプロンプトが表示される。y (yes の意味) を入力すると、バックアップ元となったファイルシステムのルートディレクトリに合わせて、現在のディレクトリのモード (アクセス権、所有者、時刻) を設定できる。このモードは、ファイルシステム全体を復元する場合に使用する。 n (no の意味) を入力すると、現在のディレクトリのモードは変更されずにそのまま残る。このモードは、バックアップの一部をファイルのバックアップ元とは異なるディレクトリに復元するときに使用する。

表 50-4 対話式の復元コマンド (続き)

オプション	説明
verbose	<p>詳細オプションのオンとオフを切り替える (対話型モードの外側では、コマンド行から <code>v</code> と入力することもできる)。詳細モードがオンの場合、対話型の <code>ls</code> コマンドでは <code>i</code> ノード番号が表示され、<code>ufsrestore</code> コマンドでは各ファイルが抽出されるたびにファイル情報が表示される。</p>
what	<p>テープやフロッピーディスク上のバックアップヘッダが表示される。</p>

第 51 章

UFS ファイルとファイルシステムのコピー (手順)

この章では、各種のバックアップコマンドを使用して、UFS ファイルとファイルシステムをディスク、テープ、フロッピーディスクにコピーする方法について説明します。

この章で説明する手順は次のとおりです。

- 748 ページの「ディスクをコピーする方法 (dd)」
- 751 ページの「ファイルシステム間でディレクトリをコピーする方法 (cpio)」
- 754 ページの「ファイルをテープにコピーする方法 (tar)」
- 755 ページの「テープ上のファイルのリストを表示する方法 (tar)」
- 755 ページの「テープからファイルを取り出す方法 (tar)」
- 756 ページの「pax コマンドを使用してファイルをテープにコピーする」
- 757 ページの「ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)」
- 759 ページの「テープ上のファイルのリストを表示する方法 (cpio)」
- 759 ページの「テープからすべてのファイルを取り出す方法 (cpio)」
- 760 ページの「テープから特定のファイルを取り出す方法 (cpio)」
- 761 ページの「ファイルをリモートテープデバイスにコピーする方法 (tar と dd)」
- 763 ページの「ファイルをリモートテープデバイスから取り出す方法」
- 764 ページの「ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)」
- 765 ページの「フロッピーディスク上のファイルのリストを表示する方法 (tar)」
- 766 ページの「ファイルをフロッピーディスクから取り出す方法 (tar)」
- 768 ページの「SunOS の旧リリース用のアーカイブを作成する方法」
- 769 ページの「bar ファイルをフロッピーディスクから取り出す方法」

ファイルシステムをコピーするためのコマンド

ファイルシステム全体をバックアップして復元する場合は、第 50 章で説明した `ufsdump` コマンドと `ufsrestore` コマンドを使用します。個々のファイル、ファイルシステムの一部、またはファイルシステム全体をコピーまたは移動する場合は、`ufsdump` と `ufsrestore` コマンドの代わりに、この章で説明する手順を使用できます。

次の表に、各種バックアップコマンドの用途を示します。

表 51-1 バックアップコマンドの用途

作業	コマンド	参照先
ファイルシステムをテープにバックアップする	<code>ufsdump</code>	700 ページの「ファイルシステムのバックアップをテープに作成する方法」
ファイルシステムのスナップショットを作成する	<code>fssnap</code>	第 48 章
ファイルシステムをテープから復元する	<code>ufsrestore</code>	725 ページの「ファイルシステム全体を復元する方法」
ファイルを他のシステムに転送する	<code>pax</code> 、 <code>tar</code> 、または <code>cpio</code>	752 ページの「ファイルとファイルシステムをテープにコピーする」
ファイルまたはファイルシステムをディスク間でコピーする	<code>dd</code>	748 ページの「ディスクをコピーする方法 (dd)」
ファイルをフロッピーディスクにコピーする	<code>tar</code>	764 ページの「ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)」

次の表に、各種のバックアップおよび復元コマンドを示します。

表 51-2 バックアップコマンドの概要

コマンド名	ファイルシステム境界の認識	複数ボリュームバックアップのサポート	物理コピー / 論理コピー
<code>volcopy</code>	あり	あり	物理
<code>tar</code>	なし	なし	論理

表 51-2 バックアップコマンドの概要 (続き)

コマンド名	ファイルシステム境界の認識	複数ボリュームバックアップのサポート	物理コピー / 論理コピー
cpio	なし	あり	論理
pax	あり	あり	論理
dd	あり	なし	物理
ufsdump/ufsrestore	あり	あり	論理

次の節では、各方法の長所と短所を説明し、コマンドの使用法および例を示します。

ファイルシステムをディスク間でコピーする

ファイルシステムをディスク間でコピーするには、次の 2 つのコマンドを使用します。

- volcopy
- dd

次の節では、dd コマンドを使用してファイルシステムをディスク間でコピーする方法について説明します。

ファイルシステムのリテラルコピーを作成する

dd コマンドでは、UFS ファイルシステムのリテラル (ブロックレベル) コピーを別のファイルシステムやテープに作成します。デフォルトでは、dd コマンドはその標準入力を標準出力にコピーします。

注 - 可変長テープドライブで dd コマンドを使用するときは、必ず適切なブロックサイズを指定してください。

標準入力、標準出力、またはその両方の代わりに、デバイス名を指定できます。次の例では、フロッピーディスクの内容が /tmp ディレクトリ内のファイルにコピーされます。

```
$ dd < /floppy/floppy0> /tmp/output.file
2400+0 records in
```

```
2400+0 records out
```

dd コマンドは、読み取りブロック数と書き込みブロック数をレポートします。+ の次の数値は、部分的にコピーされたブロックの数です。デフォルトのブロックサイズは 512 バイトです。

dd コマンドの構文は、他のほとんどのコマンドとは異なっています。オプションは *keyword=value* のペアで指定します。この場合、*keyword* は設定するオプションで、*value* はそのオプションの引数です。たとえば、標準入力と標準出力を次の構文に置き換えることができます。

```
$ dd if=input-file of=output-file
```

上記の例のリダイレクト記号の代わりに *keyword=value* の形式で指定するには、次のように入力します。

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

▼ ディスクをコピーする方法 (dd)

1. コピー元とコピー先のディスクが、同じディスクジオメトリを保持していることを確認します。
2. スーパーユーザーになるか、同等の役割を引き受けます。
3. **/reconfigure** ファイルを作成し、リブート時にシステムが追加するクローンディスクを認識するようにします。

```
# touch /reconfigure
```

4. システムをシャットダウンします。

```
# init 0
```

5. クローンディスクをシステムに接続します。
6. システムをブートします。

```
ok boot
```

7. マスターディスクをクローンディスクにコピーします。

```
# dd if=/dev/rdisk/device-name of=/dev/rdisk/device-name bs=block-size
```

```
if=/dev/rdisk/device-name
```

マスターディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。

```
of=/dev/rdisk/device-name
```

クローンディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。

`bs= blocksize`

ブロックサイズ (128K バイト、256K バイトなど) を指定できる。ブロックサイズの値を大きくすると、コピーに要する時間を短縮できる。

詳細については、`dd(1M)` のマニュアルページを参照してください。

- 新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

- クローンディスクのルート (`/`) ファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /mnt
```

- クローンディスクの `/etc/vfstab` を編集して、正しいデバイス名を参照するようにします。

たとえば、`c0t3d0` のインスタンスをすべて `c0t1d0` に変更します。

- クローンディスクのルート (`/`) ファイルシステムをマウント解除します。

```
# umount /mnt
```

- システムをシャットダウンします。

```
# init 0
```

- クローンディスクからシングルユーザーモードでブートします。

```
# boot diskn -s
```

注 - `installboot` コマンドをクローンディスクで実行する必要ありません。これは、ブートブロックがオーバーラップスライスの一部としてコピーされるためです。

- クローンディスクの構成を解除します。

```
# sys-unconfig
```

構成を解除すると、システムが停止します。

- 再びクローンディスクからブートし、ホスト名や時間帯などのシステム情報を与えます。

```
# boot diskn
```

- スーパーユーザーとしてログインして、一度システムがブートした後のシステム情報を確認します。

```
hostname console login:
```

例 — ディスクをコピーする (dd)

次の例では、マスターディスク /dev/rdisk/c0t0d0s2 をクローンディスク /dev/rdisk/c0t2d0s2 にコピーする方法を示します。

```
# touch /reconfigure
# init 0
ok boot
# dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2 bs=128k
# fsck /dev/rdisk/c0t2d0s2
# mount /dev/dsk/c0t2d0s2 /mnt
# cd /mnt/etc
# vi vfstab
(新しいディスクのエントリを変更)
# cd /
# umount /mnt
# init 0
# boot disk2 -s
# sys-unconfig
# boot disk2
```

cpio を使用してファイルシステム間でディレクトリをコピーする

cpio (コピーインとコピーアウト) コマンドを使用して、個々のファイル、ファイルグループ、またはファイルシステム全体をコピーできます。この節では、cpio コマンドを使用してファイルシステム全体をコピーする方法について説明します。

cpio コマンドは、ファイルのリストを1つの大型出力ファイルにコピーするアーカイブプログラムです。また、復元しやすいように、個々のファイルの間にヘッダーを挿入します。cpio コマンドを使用すると、ファイルシステム全体を別のスライス、別のシステム、またはテープやフロッピーディスクなどのメディアデバイスにコピーできます。

cpio コマンドは、メディアの終わりを認識し、別のボリュームを挿入するように促すプロンプトを表示するので、複数のテープやフロッピーディスクが必要なアーカイブを作成するには最も効率のよいコマンド (ufsdump 以外では) です。

cpio の使用時には、しばしば ls や find のコマンドを使用してコピーするファイルを選択し、その出力を cpio コマンドにパイプします。

▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. 目的のディレクトリに移動します。

```
# cd /filesystem1
```

3. **find** コマンドと **cpio** コマンドを組み合わせ実行し、ディレクトリツリーを *filesystem1* から *filesystem2* へコピーします。

```
# find . -print -depth | cpio -pdm filesystem2
```

.	現在の作業ディレクトリで処理を始める。
-print	ファイル名を出力する。
-depth	ディレクトリ階層を下降してバックアップ中にファイル名を出力する。
-p	ファイルのリストを作成する。
-d	必要に応じてディレクトリを作成する。
-m	ディレクトリ上で正しい変更時刻を設定する。

詳細については、**cpio(1)** のマニュアルページを参照してください。

指定したディレクトリ名からファイルがコピーされます。シンボリックリンクは保持されます。

また、**-u** オプションも指定できます。このオプションは、無条件にコピーを実行します。**-u** オプションを指定しない場合、古いファイルが、新しいファイルで置換されません。このオプションは、ディレクトリごとコピーしたいときに、コピーするファイルの一部がすでにターゲットのディレクトリ中に存在する場合に便利です。

4. コピー先ディレクトリの内容を表示して、コピーに成功したかどうかを確認します。

```
# cd /filesystem2  
# ls
```

5. ソースディレクトリが不要な場合は削除します。

```
# rm -rf /filesystem1
```

例 — ファイルシステム間でディレクトリをコピーする (cpio)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

ファイルとファイルシステムをテープにコピーする

tar、pax、および cpio コマンドを使用すると、ファイルとファイルシステムをテープにコピーできます。どのコマンドを選択するかは、コピーする目的に応じて異なります。3つのコマンドはすべて raw デバイスを使用するので、使用する前にテープ上でファイルシステムをフォーマットまたは作成する必要はありません。

表 51-3 tar、pax、cpio コマンドの長所と短所

コマンド	機能	長所	短所
tar	ファイルやディレクトリサブツリーを1本のテープにコピーする場合に使用する。	<ul style="list-style-type: none">■ ほとんどの UNIX オペレーティングシステムで利用できる。■ パブリックドメインバージョンもすぐに利用できる。	<ul style="list-style-type: none">■ ファイルシステムの境界を認識しない。■ 絶対パス名の長さが255文字を超えることができない。■ 空のディレクトリや特殊ファイル(デバイスファイルなど)をコピーしない。■ 複数のテープボリュームを作成する場合は使用できない。

表 51-3 tar、pax、cpio コマンドの長所と短所 (続き)

コマンド	機能	長所	短所
pax	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用する。または、POSIX 準拠システムとの間でファイルをコピーする場合に使用する。	<ul style="list-style-type: none"> ■ POSIX 準拠システムに対する互換性は、tar コマンドや cpio コマンドよりもよい。 ■ マルチベンダーサポート 	tar コマンドの短所を参照。ただし、pax は、複数のテープボリュームを作成できる。
cpio	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用する。または、SunOS 5.9 システムから SunOS 4.0/4.1 システムにファイルをコピーする場合に使用する。	<ul style="list-style-type: none"> ■ tar コマンドよりも効率的に、データをテープに書き込む。 ■ 復元時、テープ中の不良箇所をスキップする。 ■ 異なるシステムタイプ間の互換性のために、異なるヘッダー形式 (tar、ustar、crc、odc、bar) でファイルを書き込むオプションを提供する。 ■ 複数のテープボリュームを作成する。 	tar や pax コマンドよりも構文が難しい。

使用するテープドライブとデバイス名は、各システムのハードウェアの構成によって異なります。テープデバイス名の詳細については、771 ページの「使用するメディアの選択」を参照してください。

tar を使用してファイルをテープにコピーする

tar コマンドを使用してファイルをテープにコピーする前に、次のことを覚えておかなければなりません。

- tar コマンドに `-c` オプションを指定してファイルをテープにコピーすると、テープに入っているすべての既存のファイルまたはテープの現存の位置以降にある既存のファイルすべてが破壊(上書き)される。

- ファイルをコピーするとき、ファイル名の一部にファイル名置換ワイルドカード文字 (? と *) を使用できる。たとえば、接尾辞 .doc が付いたすべての文書をコピーするには、ファイル名引数として *.doc と入力する。
- tar アーカイブからファイルを抽出するときには、ファイル名置換ワイルドカードは使用できない。

▼ ファイルをテープにコピーする方法 (tar)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ tar cvf /dev/rmt/n filenames
```

c	アーカイブの作成を指定する。
v	各ファイルがアーカイブされるたびに、その名前を表示する。
f /dev/rmt/n	アーカイブを指定したデバイスまたはファイルに書き込むように指定する。
filenames	コピーするファイルとディレクトリを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

指定した名前のファイルがテープにコピーされ、テープ上の既存のファイルが上書きされます。

4. テープをドライブから取り出して、テープラベルにファイル名を記入します。
5. コピーしたファイルがテープ上に存在することを確認します。

```
$ tar tvf /dev/rmt/n
```

tar テープ上のファイルを表示する方法については、755 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

例 — ファイルをテープにコピーする (tar)

次の例では、3つのファイルをテープドライブ0のテープにコピーします。

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
```

```
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/n
```

▼ テープ上のファイルのリストを表示する方法 (tar)

1. テープをテープドライブに挿入します。
2. テープの内容を表示します。

```
$ tar tvf /dev/rmt/n
```

t	テープ上のファイルの内容一覧が表示される。
v	t オプションと一緒に使用すると、テープ上のファイルに関する詳細情報が表示される。
f /dev/rmt/n	テープデバイスを示す。

例 — テープ上のファイルのリストを表示する (tar)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ tar tvf /dev/rmt/0
drwx--x--x  0/1          0 Jul 14 09:24 2001 reports/
-rw-----t  0/1      30000 Jul 14 09:23 2001 reports/reportA
-rw-----t  0/1      31000 Jul 14 09:24 2001 reports/reportB
-rw-----t  0/1      32000 Jul 14 09:24 2001 reports/reportC
```

▼ テープからファイルを取り出す方法 (tar)

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルを取り出します。

```
$ tar xvf /dev/rmt/n [filenames]
```

x	指定したアーカイブファイルからのファイルの抽出を指定する。指定したドライブのテープに含まれるすべてのファイルが現在のディレクトリにコピーされる。
---	--

<code>v</code>	各ファイルを取り出すたびに、その名前を表示する。
<code>f /dev/rmt/n</code>	アーカイブを含むテープデバイスを示す。
<code>filenames</code>	取り出すファイルを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

詳細については、tar(1)のマニュアルページを参照してください。

4. ファイルがコピーされていることを確認します。

```
$ ls -l
```

例 — テープ上のファイルを取り出す (tar)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

注 - テープから抽出されるファイル名は、アーカイブに格納されているファイル名と同一でなければなりません。ファイルの名前やパス名が不明な場合は、まずテープ上のファイルのリストを表示します。テープ上のファイルをリスト表示する方法については、755 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

pax コマンドを使用してファイルをテープにコピーする

▼ ファイルをテープにコピーする方法 (pax)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/0 filenames
```

<code>-w</code>	書き込みモードを有効にする。
<code>-f /dev/rmt/0</code>	テープドライブを識別する。
<code>filenames</code>	コピーするファイルとディレクトリを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

詳細については、`pax(1)` のマニュアルページを参照してください。

4. ファイルがテープにコピーされていることを確認します。

```
$ pax -f /dev/rmt/0
```

5. テープをドライブから取り出して、テープラベルにファイル名を記入します。

例 — ファイルをテープにコピーする (pax)

次の例は、`pax` コマンドを使用して、現在のディレクトリ内のファイルをすべてコピーする方法を示します。

```
$ pax -w -f /dev/rmt/0 .  
$ pax -f /dev/rmt/0  
filea fileb filec
```

cpio コマンドを使用してファイルをテープにコピーする

▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み保護をしていないテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ ls | cpio -oc> /dev/rmt/n
```

```

ls                                cpio コマンドにファイル名のリストを渡す。
cpio -oc                          cpio コマンドがコピーアウトモード (-o) で動作
                                  し、ASCII 文字形式 (-c) でヘッダー情報を書き
                                  込むように指定する。このオプションにより他の
                                  ベンダーのシステムとの互換性を保つ。
> /dev/rmt/n                       出力ファイルを指定する。

```

ディレクトリ内のすべてのファイルは、指定したドライブ内のテープにコピーされ、テープ上の既存のファイルが上書きされます。コピーされた合計ブロック数が表示されます。

4. ファイルがテープにコピーされていることを確認します。

```
$ cpio -civt < /dev/rmt/n
```

5. テープをドライブから取り出して、テープラベルにファイル名を記入します。

例 — ディレクトリ内のすべてのファイルをテープにコピーする (cpio)

次の例では、 /export/home/kryten ディレクトリ内のすべてのファイルをテープドライブ 0 のテープにコピーする方法を示します。

```

$ cd /export/home/kryten
$ ls | cpio -oc> /dev/rmt/0
92 blocks
$ cpio -civt < /dev/rmt/0
-rw-----t  1 kryten  users      400 Jul 14 09:28 2001, b
drwx--x--x  2 kryten  users        0 Jul 14 09:26 2001, letters
-rw-----t  1 kryten  users    10000 Jul 14 09:26 2001, letter1
-rw-----t  1 kryten  users    10100 Jul 14 09:26 2001, letter2
-rw-----t  1 kryten  users    11100 Jul 14 09:27 2001, letter3
-rw-----t  1 kryten  users   12300 Jul 14 09:27 2001, letter4
drwx--x--x  2 kryten  users        0 Jul 14 09:27 2001, memos
-rw-----t  1 kryten  users     400 Jul 14 09:28 2001, memosmemoU
-rw-----t  1 kryten  users     500 Jul 14 09:28 2001, memosmemoW
-rw-----t  1 kryten  users     100 Jul 14 09:27 2001, memosmemoX
-rw-----t  1 kryten  users     200 Jul 14 09:28 2001, memosmemoY
-rw-----t  1 kryten  users     150 Jul 14 09:28 2001, memosmemoZ
drwx--x--x  2 kryten  users        0 Jul 14 09:24 2001, reports
92 blocks
$

```

▼ テープ上のファイルのリストを表示する方法 (cpio)

注 - 内容一覧を表示するには、cpio コマンドがアーカイブ全体を処理する必要があるため、かなりの時間がかかります。

1. テープをテープドライブに挿入します。
2. テープ上のファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/n
```

-c	ファイルを ASCII 文字形式で読み込むように指定する。
-i	cpio がコピーインモードで動作することを指定する。この時点ではファイルをリストするだけ。
-v	ls -l コマンドの出力と同様の形式で出力を表示する。
-t	指定したテープドライブ内のテープ上にあるファイルの内容一覧を表示する。
< /dev/rmt/n	既存の cpio アーカイブの入力ファイルを指定する。

例 — テープ上のファイルのリストを表示する (cpio)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/0
drwx--x--x  2 kryten  users          0 Jul 14 09:34 2001, answers
-rw-----t  1 kryten  users         800 Jul 14 09:36 2001, b
drwx--x--x  2 kryten  users          0 Jul 14 09:32 2001, sc.directives
-rw-----t  1 kryten  users       200000 Jul 14 09:35 2001, direct241
drwx--x--x  2 kryten  users          0 Jul 14 09:32 2001, tests
-rw-----t  1 kryten  users          800 Jul 14 09:36 2001, test13times
396 blocks
```

▼ テープからすべてのファイルを取り出す方法 (cpio)

相対パス名を使用してアーカイブを作成した場合、入力ファイルはそれを取り出すときに現在のディレクトリ内のディレクトリとして作成されます。ただし、絶対パス名を指定してアーカイブを作成した場合は、それと同じ絶対パス名を使用してシステム上でファイルが再び作成されます。



注意 – 絶対パス名を使用すると、自分のシステム上にある元のファイルを上書きすることになるので危険です。

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからすべてのファイルを取り出します。

```
$ cpio -icvd < /dev/rmt/n
```

-i	標準入力からファイルを取り出す。
-c	ファイルを ASCII 文字形式で読み込むように指定する。
-v	取り出されたファイルを <code>ls</code> コマンドの出力と同様の形式で表示する。
-d	必要に応じてディレクトリを作成する。
< /dev/rmt/n	出力ファイルを指定する。

4. ファイルがコピーされていることを確認します。

```
$ ls -l
```

例 — テープからすべてのファイルを取り出す (cpio)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
$ cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

▼ テープから特定のファイルを取り出す方法 (cpio)

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルのサブセットを取り出します。

```
$ cpio -icv "*file" < /dev/rmt/n
```


-i	標準入力からファイルを取り出す。
-c	ヘッダーを ASCII 文字形式で読み込むように指定する。
-v	取り出されたファイルを <code>ls</code> コマンドの出力と同様の形式で表示する。
"*file"	パターンに一致するすべてのファイルを現在のディレクトリにコピーするように指定する。複数のパターンを指定できるが、個々のパターンを二重引用符で囲まなければならない。
< /dev/rmt/n	入力ファイルを指定する。

詳細については、`cpio(1)` のマニュアルページを参照してください。

4. ファイルがコピーされていることを確認します。

```
$ ls -l
```

例 — 指定したファイルをテープから取り出す (`cpio`)

次の例では、末尾に接尾辞 `chapter` が付いているすべてのファイルをドライブ 0 のテープから取り出します。

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

ファイルをリモートテープデバイスにコピーする

▼ ファイルをリモートテープデバイスにコピーする方法 (`tar` と `dd`)

1. リモートテープドライブを使用するには、次の前提条件を満たしている必要があります。

- a. ローカルホスト名 (および、オプションでコピーを実行するユーザーのユーザー名) が、リモートシステムの `/etc/hosts.equiv` ファイルに記述されている必要がある。または、コピーを実行するユーザーが、リモートマシン上の自分のホームディレクトリをアクセス可能にし、かつ `$HOME/.rhosts` 内にローカルマシン名を記述しておく必要がある。
詳細については、`hosts.equiv(4)` のマニュアルページを参照。
 - b. リモートシステムのエントリがローカルシステムの `/etc/inet/hosts` ファイル内またはネームサービスの `hosts` ファイル内になければならない。
2. リモートコマンドの実行に必要なアクセス権を保持していることを確認するには、次のように入力します。

```
$ rsh remotehost echo test
```

`test` と表示された場合、リモートコマンドの実行に必要なアクセス権を保持しています。Permission denied と表示された場合は、手順 1 の内容を確認してください。

3. ファイルを置きたいディレクトリに移動します。
4. テープをテープドライブに挿入します。
5. ファイルをリモートテープドライブにコピーします。

```
$ tar cvf - filenames | rsh remote-host dd of=/dev/rmt/n obs=block-size
```

<code>tar cf</code>	テープアーカイブを作成し、アーカイブに含まれるファイルをリスト表示し、テープデバイスを指定する。
<code>- (ハイフン)</code>	可変部としてテープデバイスの代わりに指定する。
<code>filenames</code>	コピーするファイルを指定する。
<code> rsh remote-host</code>	<code>tar</code> コマンドの出力をパイプを通してリモートシェルに渡す。
<code>dd of= /dev/rmt/n</code>	出力デバイスを指定する。
<code>obs= block-size</code>	ブロック係数を指定する。

6. テープをドライブから取り出して、テープラベルにファイル名を記入します。

例 — ファイルをリモートテープドライブにコピーする (tar と dd)

```
# tar cvf - * | rsh mercury dd of=/dev/rmt/0 bs=126b
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
```

```
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

▼ ファイルをリモートテープデバイスから取り出す方法

1. テープをテープドライブに挿入します。
2. 一時ディレクトリに移動します。

```
$ cd /var/tmp
```

3. ファイルをリモートテープドライブから取り出します。

```
$ rsh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

```
rsh remote-host
```

dd コマンドを使用してテープデバイスからファイルを取り出すために起動するリモートシェル。

```
dd if=/dev/rmt/n
```

入力デバイスを指定する。

```
| tar xvBpf -
```

dd コマンドの出力を tar コマンドにパイプして、ファイルを復元する。

4. ファイルが抽出されたことを確認します。

```
$ ls -l /var/tmp
```

例 — ファイルをリモートのテープドライブから抽出する

```
$ cd /var/tmp
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```

ファイルとファイルシステムをフロッピーディスクにコピーする

ファイルやファイルシステムをフロッピーディスクにコピーする前に、フロッピーディスクをフォーマットする必要があります。フロッピーディスクをフォーマットする方法については、第 19 章を参照してください。

`tar` コマンドを使用して、UFS ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーします。

UFS ファイルを複数のフォーマット済みフロッピーディスクにコピーする必要がある場合は、`cpio` コマンドを使用します。`cpio` コマンドはメディアの終わりを認識し、次のボリュームの挿入を促すプロンプトを表示します。

注 - ボリューム管理の関係で、`cpio` コマンドを使用して UFS ファイルを複数のフォーマット済みフロッピーディスクにコピーする手順は単純ではありません。

ファイルをフロッピーディスクにコピーする際の注意事項

- `tar` に `-c` オプションを指定してフォーマット済みフロッピーディスクにファイルをコピーすると、フロッピーディスク上の既存のファイルは破壊 (上書き) される。
- `tar` イメージを格納済みのフロッピーディスクはマウントできない。
- 複数のボリュームを扱う場合は、`cpio` コマンドを使用してください。`tar` コマンドは 1 つのボリュームに対して使用できるユーティリティです。

詳細については、`tar(1)` のマニュアルページを参照してください。

▼ ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (`tar`)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み保護されていないフォーマット済みフロッピーディスクをドライブに挿入します。
3. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

4. 必要に応じて、再度フォーマットします。

```
$ rmformat -U /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
```

5. ファイルをフロッピーディスクにコピーします。

```
$ tar cvf /vol/dev/aliases/floppy0 filename ...
```

指定した名前のファイルがフロッピーディスクにコピーされ、フロッピーディスク上の既存のファイルがすべて上書きされます。

6. ファイルがコピーされていることを確認します。

```
$ tar tvf /vol/dev/aliases/floppy0
```

ファイルのリストを表示する方法については、765 ページの「フロッピーディスク上のファイルのリストを表示する方法 (tar)」を参照してください。

7. フロッピーディスクをドライブから取り出します。

8. ファイル名をフロッピーディスクラベルに記入します。

例 — ファイルを1枚のフォーマット済みフロッピーディスクにコピーする (tar)

次の例では、2つのファイルをフロッピーディスクにコピーします。

```
$ volcheck
$ cd /home/smith
$ ls evaluation*
evaluation.doc  evaluation.doc.backup
$ tar cvf /vol/dev/aliases/floppy0 evaluation*
a evaluation.doc 86 blocks
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/aliases/floppy0
```

▼ フロッピーディスク上のファイルのリストを表示する方法 (tar)

1. フロッピーディスクをドライブに挿入します。
2. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

3. フロッピーディスク上のファイルのリストを表示します。

```
$ tar tvf /vol/dev/aliases/floppy0
```

例 — フロッピーディスク上のファイルのリストを表示する (tar)

次の例では、フロッピーディスク上のファイルのリストを表示します。

```
$ volcheck
tar tvf /vol/dev/aliases/floppy0
rw-rw-rw-6693/10 44032 Jun 9 15:45 evaluation.doc
rw-rw-rw-6693/10 43008 Jun 9 15:55 evaluation.doc.backup
$
```

▼ ファイルをフロッピーディスクから取り出す方法 (tar)

1. ファイルを置きたいディレクトリに移動します。
2. フロッピーディスクをドライブに挿入します。
3. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

4. ファイルをフロッピーディスクから取り出します。

```
$ tar xvf /vol/dev/aliases/floppy0
```

フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

5. ファイルが取り出されたことを確認します。

```
$ ls -l
```

6. フロッピーディスクをドライブから取り出します。

例 — ファイルをフロッピーディスクから取り出す (tar)

次の例では、フロッピーディスクからすべてのファイルを取り出します。

```
$ volcheck
$ cd /home/smith/Evaluations
$ tar xvf /vol/dev/aliases/floppy0
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

次の例では、フロッピーディスクから個々のファイルを取り出します。

```
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0 evaluation.doc
x evaluation.doc, 44032 bytes, 86 tape blocks
$ ls -l
```

指定した名前のファイルがフロッピーディスクから取り出され、現在の作業ディレクトリに格納されます。

ファイルを複数のフロッピーディスクにアーカイブする方法

大量のファイルをフロッピーディスクにコピーする場合は、いっぱいになったフロッピーディスクを別のフォーマット済みフロッピーディスクと交換するように促すプロンプトを表示させることができます。cpio コマンドにはこの機能があります。使用する cpio コマンドはファイルをテープにコピーする場合と同じですが、テープデバイス名ではなくデバイスとして /vol/dev/aliases/floppy0 を指定します。

cpio コマンドの使用方法については、757 ページの「ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)」を参照してください。

ファイルを別のヘッダー形式でコピーする

SunOS 5.9 の cpio コマンドを使用して作成したアーカイブには、SunOS の旧リリースとの互換性がない場合があります。cpio コマンドを使用すると、他の複数の形式で読み込めるアーカイブを作成できます。これらの形式は、-H オプションと次のいずれかの引数で指定します。

- crc または CRC - チェックサム付きの ASCII ヘッダー
- ustar または USTAR - IEEE/P1003 データ交換
- tar または TAR - tar のヘッダーと形式
- odc - 小型デバイス番号付きの ASCII ヘッダー
- bar - bar のヘッダーと形式

ヘッダーオプションを使用する場合の構文を次に示します。

```
cpio -o -H header-option < file-list> output-archive
```

SunOS の旧リリース用のアーカイブを作成する方法

cpio コマンドを使用してアーカイブを作成します。

```
$ cpio -oH odc < file-list > /dev/rmt/n
```

引数 -H は、入力に対して出力の場合と同じ意味を持ちます。-H オプションを使用してアーカイブを作成した場合は、読み込むときにも同じオプションを使用しないと、cpio コマンドが失敗します。次の例には、cpio のエラーメッセージが含まれていません。

例 — SunOS の旧リリース用にアーカイブを作成する

```
$ find . -print | cpio -oH tar> /tmp/test
113 blocks
$ cpio -iH bar < /tmp/test
cpio: Invalid header "bar" specified
USAGE:
    cpio -i[bcdfkmrstuvBSV6] [-C size] [-E file] [-H hdr]

[-I file [-M msg]] [-R id] [patterns]
    cpio -o[acvABLV] [-C size] [-H hdr] [-O file [-M msg]]
    cpio -p[adlmuvLV] [-R id] directory
```

各種オプションを使用してアーカイブを作成するときには、必ずメディアのラベルにアーカイブ上のファイル名やファイルシステム名と一緒にコマンド構文を記入してください。

アーカイブの作成時にどの cpio オプションを使用したかがわからない場合は、各種オプションをいろいろ組み合わせてみて、どの方法でアーカイブを読み込むことができるかを調べてください。

オプションのリストについては、cpio(1) のマニュアルページを参照してください。

bar コマンドで作成したファイルを取り出す

SunOS 4.0 または 4.1 の bar コマンドを使用してアーカイブしたファイルをフロッピーディスクから取り出すには、cpio の -H bar オプションを使用します。

注 – ファイルを取り出すには、-H bar オプションと共に -i オプションを使用する必要があります。bar ヘッダーオプションを使用してファイルを作成することはできません。

▼ **bar** ファイルをフロッピーディスクから取り出す方法

1. ファイルを置きたいディレクトリに移動します。
2. フロッピーディスクをドライブに挿入します。
3. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

4. フロッピーディスクから **bar** ファイルを取り出します。
フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

```
$ cpio -ivH bar < /vol/dev/aliases/floppy0
```


第 52 章

テープドライブの管理 (手順)

この章では、テープドライブを管理する方法について説明します。

この章で説明する手順は次のとおりです。

- 774 ページの「テープドライブの状態を表示する方法」
- 775 ページの「磁気テープカートリッジのたるみを直す方法」
- 776 ページの「磁気テープカートリッジを巻き戻す方法」

使用するメディアの選択

通常は、次のテープメディアを使用して Solaris システムのバックアップを作成します。

- 1/2 インチのリールテープ
- 1/4 インチのストリームカートリッジテープ
- 8mm のカートリッジテープ
- 4mm のカートリッジテープ (DAT)

フロッピーディスクを使用してバックアップを実行することもできますが、時間がかかり煩雑です。

どのメディアを選択するかは、メディアをサポートする機器とファイルの格納に使用するメディア (通常はテープ) の可用性によって決まります。バックアップはローカルシステムから実行しなければなりません、ファイルはリモートデバイスに書き込まれます。

次の表に、ファイルシステムの典型的なバックアップメディアと、その記憶容量を示します。容量は、ドライブのタイプとテープに書き込むデータによって異なります。

表 52-1 メディアの記憶容量

メディア	容量
1/2 インチのリールテープ	140M バイト (6250bpi)
2.5G バイト、1/4 インチのカートリッジ (QIC) テープ	2.5G バイト
DDS3 4mm カートリッジテープ (DAT)	12 - 24G バイト
14G バイト、8mm カートリッジテープ	14G バイト
DLT™ 7000 1/2 インチ カートリッジテープ	35 - 70G バイト

バックアップデバイス名

バックアップに使用するテープまたはフロッピーディスクドライブに論理デバイス名を与えて指定します。この名前は、「raw」デバイスファイルの格納されたサブディレクトリを指し、ドライブの論理ユニット番号が含まれます。テープドライブの命名規則に従い、物理デバイス名ではなく論理デバイス名を使用します。次の表に、この命名方式を示します。

表 52-2 バックアップデバイスの基本的なデバイス名

デバイスの種類	名前
テープ	<code>/dev/rmt/n</code>
フロッピーディスク	<code>/vol/dev/rdiskette0/unlabeled</code>

通常は、次の図に示す方法で、テープドライブデバイスを指定します。

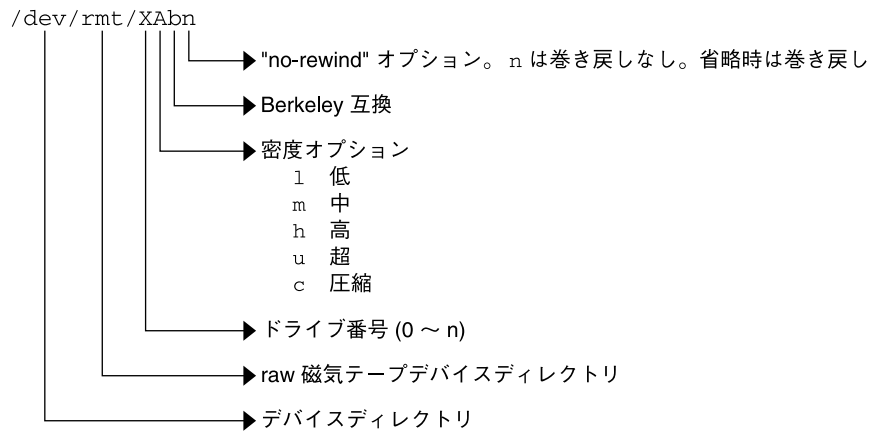


図 52-1 テープドライブデバイス名

密度を指定しないと、テープドライブは通常その「推奨」密度で書き込みますが、これは一般にそのテープドライブでサポートされる最大密度です。ほとんどの SCSI ドライブはテープ上の密度やフォーマットを自動的に検出し、それに従って読み取りを実行します。ドライブでサポートされる密度を調べるには、`/dev/rmt` サブディレクトリを確認してください。このサブディレクトリには、各テープで異なる出力密度をサポートするためのテープデバイスファイルのセットが含まれます。

SCSI コントローラは、最大 7 台の SCSI テープドライブを持つことができます。

テープドライブの巻き戻しオプションを指定する

通常は、テープドライブを 0 から n までの論理デバイス番号で指定します。次の表に、「巻き戻し」または「巻き戻しなし」のオプションを付けてテープデバイス名を指定する方法を示します。

表 52-3 テープドライブの「巻き戻し」または「巻き戻しなし」オプション

ドライブおよび巻き戻し	使用するオプション
第 1 のドライブ、巻き戻し	<code>/dev/rmt/0</code>
第 1 のドライブ、巻き戻しなし	<code>/dev/rmt/0n</code>
第 2 のドライブ、巻き戻し	<code>/dev/rmt/1m</code>
第 2 のドライブ、巻き戻しなし	<code>/dev/rmt/1n</code>

テープドライブに別の密度を指定する

デフォルトでは、テープドライブはその「推奨」密度で書き込みますが、これは一般にそのテープドライブでサポートされる最大密度です。テープデバイスを指定しなければ、コマンドはデバイスでサポートされるデフォルト密度でドライブ番号 0 に書き込みます。

テープを特定の密度しかサポートされないテープドライブが付いたシステムに転送するには、目的の密度で書き込むデバイス名を指定します。次の表に、テープドライブに別の密度を指定する方法を示します。

表 52-4 テープドライブに別の密度を指定する

ドライブ、密度、巻き戻し	使用するオプション
第 1 のドライブ、低密度、巻き戻し	<code>/dev/rmt/0l</code>
第 1 のドライブ、低密度、巻き戻しなし	<code>/dev/rmt/0ln</code>
第 2 のドライブ、中密度、巻き戻し	<code>/dev/rmt/1m</code>
第 2 のドライブ、中密度、巻き戻しなし	<code>/dev/rmt/1mn</code>

ユニットおよび密度オプションについては、772 ページの「バックアップデバイス名」を参照してください。

テープドライブの状態を表示する

mt コマンドの `status` オプションを使用すると、テープドライブに関する状態情報を表示できます。mt コマンドは、`/kernel/drv/st.conf` ファイルに記述されているすべてのテープドライブの情報を表示します。

▼ テープドライブの状態を表示する方法

1. 情報を表示するドライブにテープをロードします。
2. `mt` コマンドを使用してテープドライブ情報を表示します。

```
# mt -f /dev/rmt/n status
```
3. テープドライブ番号を 1、2、3 というように置き換えて手順 1 と 2 を繰り返し、使用可能なすべてのテープドライブに関する情報を表示します。

例 — テープドライブの状態を表示する

次の例は、QIC-150 テープドライブ (/dev/rmt/0) と Exabyte テープドライブ (/dev/rmt/1) の状態を示します。

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0   retries= 0
file no= 0   block no= 0
```

次の方法を使用すると、システムを手早くポーリングしてすべてのテープドライブを検査できます。

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

磁気テープカートリッジの取り扱い

テープの読み込み中にエラーが発生した場合は、テープのたるみを直し、テープドライブを掃除してからやり直してください。

磁気テープカートリッジのたるみを直す方法

mt コマンドを使用して磁気テープカートリッジのたるみを直します。

たとえば、次のようになります。

```
$ mt -f /dev/rmt/1 retension
$
```

注 - QIC 以外のテープドライブのたるみは直さないでください。

磁気テープカートリッジを巻き戻す方法

磁気テープカートリッジを巻き戻すには、`mt` コマンドを使用します。

たとえば、次のようになります。

```
$ mt -f /dev/rmt/1 rewind
$
```

ドライブの管理とメディア処理のガイドライン

バックアップテープは読み込めなければ役に立ちません。このため、定期的に掃除およびチェックを行い、テープドライブが正常に動作するようにしてください。テープドライブのクリーニング手順については、ハードウェアのマニュアルを参照してください。次のいずれかの方法で、テープハードウェアをチェックできます。

- テープにファイルをコピーし、ファイルを読み込んで、コピーをオリジナルと比較する。
- `ufsdump` コマンドの `-v` オプションを使用して、メディアの内容をソースファイルシステムと比較して検査する。`-v` オプションを有効にするには、ファイルシステムをマウント解除するか、完全にアイドル状態にする必要がある。

ハードウェアは、システムからレポートされないような障害を起こす可能性があるので注意してください。

バックアップ後は、必ずテープにラベルを付けてください。第 46 章で説明したバックアップ方法を採用する場合は、ラベルを「テープ A」、「テープ B」という具合に指定する必要があります。このラベルは変更しないでください。バックアップを実行するたびに別のテープラベルを作成して、バックアップ日付、マシン名、バックアップを作成したファイルシステム、バックアップレベル、テープ番号 (複数のボリュームにまたがる場合は n 本のうちの 1 本目)、サイト特有の情報を入力します。

テープは、磁気を発生させる機器から離れた埃のない安全な場所に保管してください。たとえば、アーカイブしたテープを遠隔地の防火キャビネットに保管します。

各ジョブ (バックアップ) がどのメディア (テープボリューム) に格納されているかということと、各バックアップファイルがどこに保管されているかを記録したログを作成し、管理する必要があります。

索引

数字・記号

4.3 Tahoe ファイルシステム, 554
9660 CD フォーマット, 242

A

Admintool

パッケージの削除 (手順), 339
パッケージの追加 (手順), 337, 339
パッケージの追加と削除
概要, 337

ASN.1 (Abstract Syntax Notation 1), 286

AutoClient, 定義, 126

autofs, 568

B

banner コマンド (PROM), 189

bar コマンド, 作成したファイルを取り出す (手順), 768

base64, 286

basedir キーワード (管理ファイル), 293, 294

bin グループ, 75

Bourne シェル

ユーザー初期設定ファイルも参照

環境変数, 97, 101

基本機能, 96

シェル (ローカル) 変数, 97, 99

Break キー, 199, 201

BSD Fat Fast ファイルシステム, 554

C

cachefspack コマンド

概要, 613

使用方法, 613

CacheFS ファイルシステム

cachefspack エラーの障害障害追跡, 618

cachefspack コマンドでパッキング (概要)
, 613

cachefspack コマンドでパッキング (手順)
, 613

CacheFS 統計情報の表示, 626

CacheFS の統計情報の収集 (概要), 622

CacheFS ログインの設定 (手順), 623

CacheFS ログインの停止, 624

CacheFS ログファイルの場所を調べる, 624

fsck コマンドのチェック (手順), 611

fsck コマンドのチェック (例), 611

概要, 599

作業セット(キャッシュ)のサイズの表
示, 625

削除 (手順), 609

作成 (手順), 601

情報の表示 (手順), 608

パッキングリストの作成 (手順), 615

パックされたファイルの表示 (手順), 614

パックされたファイルの表示 (例), 614

パラメータ, 599

ファイルのパッキングの解除 (手順), 617

ファイルをキャッシュにパックする (手順)
, 616

マウント (手順), 603

CD

ISO 9660 フォーマット, 242

CD (続き)
UFS CD
SPARC と x86 フォーマット, 242
名前, 240
CD-ROM デバイス
マウント済み CD からのソフトウェアの追加
例, 329
CDPATH 環境変数, 98
cdrw コマンド
CD-RW メディアの消去 (手順), 278
CD 上のオーディオトラックの抽出 (手順)
, 276
CD のコピー (手順), 277
CD メディアの確認 (手順), 270
CD メディアのチェック (手順), 271
アクセスの制限 (手順), 270
オーディオ CD の作成 (手順), 275
説明, 267
データ CD とオーディオ CD の書き込み (概
要), 269
マルチセッションのデータ CD の作成 (手順)
, 273
cfgadm
PCI ホットプラグ (概要), 387
SCSI ホットプラグ (概要), 387
cfsadmin コマンド, 601, 609
clri コマンド, 558
cpio コマンド, 757
概要, 750
指定したファイルをテープから取り出す (手
順), 760
テープからすべてのファイルを取り出す (手
順), 760
テープ上のファイルのリスト表示 (手順)
, 759
ファイルシステム間でディレクトリをコピー
(手順), 751
別のヘッダー形式でファイルをコピー (手順)
, 767
cp コマンド, リムーバブルメディア上の情報の
コピー (手順), 245
.cshrc ファイル
カスタマイズ, 80, 96, 102
説明, 94
C シェル
環境変数, 97, 101
基本機能, 96
シェル (ローカル) 変数, 97, 99

C シェル (続き)
ユーザー初期設定ファイル, 94, 102, 108
ユーザー初期設定ファイルを参照
サイト初期設定ファイルの参照用, 95
作成, 96

D

daemon グループ, 75
dd コマンド
概要, 747
クローンディスクの作成 (手順), 748
ファイルをリモートテープドライブから取り
出す (tar)(手順), 763
ファイルをリモートテープにコピー (tar)(手
順), 761
DER (Distinguished Encoding Rules), 286
/dev/dsk ディレクトリ, 447
/dev/rdisk ディレクトリ, 447
devfsadm コマンド, 446
dfstab ファイル
共有されたローカルのリムーバブルメディア
の設定 (手順), 249
ユーザーのホームディレクトリ, 112
df コマンド, 448, 558
dmesg コマンド, 383
SPARC の例, 383
x86 の例, 384
DOS, ファイルシステム, 554
driver not attached メッセージ, 381
DVD-ROM, 556

E

eject コマンド, リムーバブルメディア (手順)
, 247
env コマンド, 96
/etc/dfs/dfstab ファイル
共有されたリムーバブルメディアの設定 (手
順), 249
ユーザーのホームディレクトリ, 112
/etc/dumpdates ファイル, 732
/etc/init.d ディレクトリ, 172
/etc/inittab ファイル, 163, 165
エントリの説明, 163
デフォルトの例, 164

/etc/passwd ファイル, 88
修復
 SPARC, 201
修復 (例)
 x86, 215
説明, 88
フィールド, 88
ユーザー ID 番号の割り当て, 75
ユーザーアカウントの削除, 87
/etc/passwd ファイルの修復
 SPARC, 201
 x86, 215
/etc/rmmount.conf ファイル, リムーバブル
 メディアドライブの共有 (手順), 249
/etc/shadow ファイル, 説明, 88
/etc/skel ディレクトリ, 94
/etc/vfstab ファイル, 113
/etc ファイル
 ユーザーアカウント情報, 74, 88
/export/home ディレクトリ, 561
/export/home ファイルシステム, 79

F

FDFS ファイルシステム, 558
ff コマンド, 558
FIFOFS ファイルシステム, 558
FIFO i ノード, 645
format.dat ファイル
 エントリの作成 (概要), 490
 エントリの作成 (手順), 490
 キーワード, 533, 536
 構文規則, 533
format ユーティリティ
 analyze メニュー, 530
 defect メニュー, 531
 fdisk メニュー, 529
 partition メニュー, 528, 529
 SCSI ディスクドライブの自動構成 (概要)
 , 491
 SCSI ディスクドライブの自動構成 (手順)
 , 493
 Solaris fdisk パーティションの作成 (手順)
 , 515
 概要, 467
 機能と利点, 467
 コマンド名の入力方法, 538

format ユーティリティ (続き)
 システム上のディスクの確認 (手順), 476
 システム上のディスクの確認 (例), 478
 使用上のガイドライン, 469
 使用する場合, 468
 情報の保存についての推奨事項, 525
 ディスクスライス情報の表示 (手順), 481
 ディスクスライス情報の表示 (例), 482
 ディスクスライスとディスクラベルの作成
 (手順)
 SPARC, 503
 ディスクスライスとディスクラベルの作成 (
 手順)
 x86, 521
 ディスクのフォーマット (手順), 480
 ディスクのフォーマット (例), 480
 ディスクラベルの作成
 例, 485
 ディスクラベルの作成 (手順), 483
 入力, 537, 539
 破損したディスクラベルの復元 (手順), 487
 フォーマットされているかを調べる (手順)
 , 479
 ブロック番号の指定方法, 537
 ヘルプ機能の使用方法, 539
 メインメニュー, 526
free hog スライス, 提供側スライスを参照
FRU (現場交換可能ユニット), 126
fsck コマンド, 448, 558
 FSACTIVE 状態フラグ, 640
 FSBAD 状態フラグ, 640
 FSCLEAN 状態フラグ, 640
 FSSTABLE 状態フラグ, 640
 構文とオプション, 655
 修正する条件, 642
 修復, 651
 状態フラグ, 640
 対話式で使用, 649
 チェック
 i ノードリストサイズ, 643
 空き i ノード, 644
 空きブロック, 644
 スーパーブロック, 643
fsdb コマンド, 559
fssnap コマンド
 UFS スナップショット情報の削除 (手順)
 , 712
 UFS スナップショットの作成 (手順), 710

fssnap コマンド (続き)
UFS スナップショットの情報の表示 (手順)
 , 711
fstypes ファイル, 569
fstyp コマンド, 559
fuser コマンド
リムーバブルメディアが使用中であるかどうかの確認 (手順), 247
リムーバブルメディアにアクセス中のプロセスの強制終了 (手順), 247

G

GECOS フィールド (passwdファイル), 89
GID, 75
大きな値, 76
定義, 80
割り当て, 81
grep コマンド, 569
groups コマンド, 81
group ファイル
説明, 88
フィールド, 91
ユーザーアカウントの削除, 87

H

halt コマンド, 178
High Sierra ファイルシステム, 554
history 環境変数, 98
HOME 環境変数, 98
/home (自動マウント), 569
/home ファイルシステム, ユーザーのホームディレクトリ, 79
HSFS, High Sierra ファイルシステムを参照

I

ID 番号
グループ, 75, 80, 81
ユーザー, 75, 76, 86
init コマンド
スタンドアロンシステムのシャットダウン, 183
説明, 178

init 状態, 実行レベルを参照
installboot コマンド, 508, 523
ISO 9660 ファイルシステム, 554
ISO 標準, 9660 CD フォーマット, 242
i ノード, 667
FIFO, 645
キャラクタ型特殊ファイル, 644
形式とタイプのチェック, 644
ごとのバイト数, 672
サイズ, 646
シンボリックリンク, 645
通常, 644
ディレクトリ, 644
不正な番号, 647
ブロック型特殊ファイル, 644
リンク数, 645
i ノードの形式, 644
i ノードの状態, 645
i ノードのタイプ, 644
i ノードのリンク数, 645
i ノードリストサイズ, 643

J

Java キーストア, 289

K

/kernel/drv ディレクトリ, 380
keytool コマンド, 289
概要, 322
Korn シェル
環境変数, 97, 101
基本機能, 96
シェル (ローカル) 変数, 97, 99
ユーザー初期設定 ファイル, 96
ユーザー初期設定ファイル, 94, 95, 102, 108
ユーザー初期設定ファイルを参照

L

L1+A キー, 199, 201
labelit コマンド, 559
LANG 環境変数, 98, 100, 101
LC 環境変数, 100, 101

LK パスワード, 87, 91
local.cshrc ファイル, 94
local.login ファイル, 94
local.profile ファイル, 94
locale 環境変数, 98
.login ファイル
 カスタマイズ, 80, 96, 102
 説明, 94
LOGNAME 環境変数, 98
lost+found ディレクトリ, 640
LPDEST 環境変数, 98

M

MAIL 環境変数, 97, 98
MANPATH 環境変数, 98
media was found メッセージ, 260
mkfile コマンド, 635, 637
mkfs コマンド, 559, 572
mkisofs コマンド, データ CD のファイルシス
 テムの作成 (手順), 272
MNTFS ファイルシステム, 561
mnttab ファイル, 566
mountall コマンド, 559
mount コマンド, 448
mt コマンド, 775

N

NAMEFS ファイルシステム, 558
ncheck コマンド, 559
newfs コマンド, 448, 572, 673
newgrp コマンド, 81
NFS
 サーバーの説明, 568
 説明, 568
 のvfstab エントリ, 586
nfsd デーモン
 起動, 249
 実行しているかどうかの確認, 248
NIS
 ユーザーアカウント, 74, 88
NIS+
 グループ, 81
 ユーザーアカウント, 74, 88
noaccess ユーザー/グループ, 75, 92

noask_pkgadd 管理ファイル, 294, 330
nobody ユーザー/グループ, 75, 92
no media was found メッセージ, 260
NP パスワード, 91

O

/opt ディレクトリ, 561
OS サーバー, 説明, 131

P

passwd ファイル, 88
 修復
 SPARC, 201
 修復 (例)
 x86, 215
 テープからの復元 (例), 724
 フィールド, 88, 89
 ユーザー ID 番号の割り当て, 75
 ユーザーアカウントの削除, 87
patchadd コマンド
 署名付きパッチ, 282
 署名付きパッチの追加 (手順), 354
 署名のないパッチの追加, 370
PatchPro, キーストア, 289
patchrm コマンド, 372
PATH 環境変数
 設定, 99, 100
 説明, 98, 99
path シェル変数, 97
PC BIOS (ブート), 222
PCFS ファイルシステム, 554
PCI デバイス
 PCI アダプタカードを取り付ける (手順)
 , 403
 PCI アダプタカードを取り外す (手順), 403
 PCI 構成の問題の障害追跡, 405
 PCI デバイス情報の表示 (手順), 402
PCMCIA メモリーカード
 他のシステム上の PCMCIA メモリーカード
 へのアクセス (例), 253
 リモートでのマウント (例), 253
PEM (Privacy Enhanced Message), 286
PKCS7 (Public Key Cryptography Standard #7)
 , 286

pkgadd コマンド

- a オプション (管理ファイル), 293, 294, 328, 330
- d オプション (デバイス名), 328, 329, 330, 331, 332
- s オプション (スプールディレクトリ), 331, 332

概要, 290, 295

使用にあたっての前提条件, 292

署名付きパッケージ, 282

署名付きパッケージの追加, 326

スプールディレクトリと, 331

スプールディレクトリと (例), 333

代替ベースディレクトリと, 294

パッケージの追加 (手順), 328

- HTTP URL の使用, 330
- ユーザーの対話操作を省略, 293, 294

pkgadm addcert コマンド, pkgadm コマンドを参照

pkgadm listcert コマンド, pkgadm コマンドを参照

pkgadm removecert コマンド, pkgadm コマンドを参照

pkgadm コマンド

- pkgadm addcert コマンド
 - 概要, 322
 - 信頼される証明書のインポート, 322
- pkgadm listcert コマンド
 - 概要, 322
 - 出力, 286
 - 証明書の情報の表示, 324
- pkgadm removecert コマンド
 - 概要, 322
 - 証明書の削除, 325

概要, 295

pkgchk コマンド

- 概要, 295
- 使用 (例), 335

pkginfo コマンド

- インストール済みのすべてのパッケージの表示 (例), 333
- 概要, 292, 295
- 使用方法, 333

pkgparam コマンド, 概要, 295

pkgrm コマンド

- rm コマンド, 292
- rm コマンドと, 335
- 概要, 290, 295, 335

pkgrm コマンド (続き)

- 使用にあたっての前提条件, 292
- 注意, 292, 335
- パッケージの削除 (手順), 335

pkgtrans コマンド, 概要, 295

- /pkg ディレクトリ, 333

PKI (Public Key Infrastructure) サイト, 289

PROCFS ファイルシステム, 概要, 557

- /proc ディレクトリ, 557, 561

prodreg コマンド, 282

- インストールした製品の一覧表示 (手順), 304
- 概要, 295, 304
- ソフトウェア製品間の依存関係の確認 (手順), 309
- ソフトウェア属性の一覧 (手順), 307
- ソフトウェアのアンインストール (手順), 313
- 損傷を受けたソフトウェアのアンインストール (手順), 317
- 損傷を受けたソフトウェアの再インストール (手順), 320
- 損傷を受けたソフトウェアの特定 (手順), 310

Product Registry

- インストールした製品情報の表示 (手順), 300
- インストールした製品の一覧表示 (手順), 304
- ソフトウェア製品間の依存関係の確認 (手順), 309
- ソフトウェア属性の一覧 (手順), 307
- ソフトウェアのアンインストール (手順), 302, 313
- ソフトウェアのインストール (手順), 301
- ソフトウェアの削除, 299
- ソフトウェアの追加, 299
- 損傷を受けたソフトウェアのアンインストール (手順), 317
- 損傷を受けたソフトウェアの再インストール (手順), 320
- 損傷を受けたソフトウェアの特定 (手順), 310
- 目的, 299

.profile ファイル

- カスタマイズ, 80, 96, 102
- 説明, 94

PROM

- ROM Rev 番号の確認, 189
- ブート設定の変更, 191
- モニター, 221
- リビジョンレベルの確認, 189
- prompt シェル変数, 98
- PROM からのブートの設定, 191
- PROM のリビジョンレベル, 189
- prtconf コマンド, 382
- prtvtop コマンド, 448
 - 使用例, 486
- PS1 環境変数, 98

R

- raw ディスクデバイスインタフェース, 447, 448
- reboot コマンド, 178
- removef コマンド, 292
- reset コマンド, 193
- rmmount.conf ファイル, リムーバブルメディアドライブの共有 (手順), 249
- Rock Ridge 拡張 (HSFS ファイルシステム), 554

S

- /sbin/rc0 スクリプト, 167
- /sbin/rc1 スクリプト, 167
- /sbin/rc2 スクリプト, 168
- /sbin/rc3 スクリプト, 170
- /sbin/rc5 スクリプト, 170
- /sbin/rc6 スクリプト, 170
- /sbin/rcS スクリプト, 170
- SCSI ディスクドライブ, 491
- SCSI テープドライブ, 773
- SCSI デバイス
 - cfgadm コマンド (手順), 392
 - cfgadm コマンドを使用した構成解除 (手順), 393
 - cfgadm コマンドを使用した構成 (手順), 394
 - cfgadm コマンドを使用した接続解除 (手順), 395
 - cfgadm コマンドを使用した接続 (手順), 396
 - cfgadm コマンドを使用して取り外す (手順), 398
- SCSI 構成の問題の障害追跡, 399

SCSI デバイス (続き)

- SCSI コントローラ上の同一デバイスを交換する (手順), 397
- SCSI デバイスを SCSI バスに取り付ける (手順), 396
- 失敗した SCSI 構成解除操作を解決する (手順), 401
- setenv コマンド, 97
- set コマンド, 97
- shadow ファイル
 - 説明, 88
 - フィールド, 90, 91
- shareall コマンド, 568
- share コマンド, 568
 - リムーバブルメディアを他のシステムで使用可能にする (手順), 249
- SHELL 環境変数, 99
- shutdown コマンド
 - サーバーのシャットダウン, 156
 - サーバーのシャットダウン (手順), 180
 - 説明, 178
 - ユーザーへの通知, 179
- /skel ディレクトリ, 94
- smpatch コマンド, 282
 - 署名付きパッチを追加するための準備, 355
 - ダウンロードとインストール (手順), 357
 - 注意点, 356
 - パッケージ要件, 345
 - パッケージ要件の確認 (手順), 357
- Solaris Product Registry
 - インストールした製品情報の表示 (手順), 300
 - ソフトウェア製品間の依存関係の確認 (手順), 309
 - ソフトウェア属性の一覧 (手順), 307
 - ソフトウェアのアンインストール (手順), 302, 313
 - ソフトウェアのインストール (手順), 301
 - ソフトウェアの削除, 299
 - ソフトウェアの追加, 299
 - 損傷を受けたソフトウェアのアンインストール (手順), 317
 - 損傷を受けたソフトウェアの再インストール (手順), 320
 - 損傷を受けたソフトウェアの特定 (手順), 310
 - 目的, 299

Solaris Web Start, ソフトウェアの追加 (手順)
 , 297
Solaris Device Configuration Assistant (デバイス
構成用補助), 概要, 207
Solaris fdisk パーティション, ガイドライ
ン, 515
Solaris 管理コンソール
RBACと組み合わせて使用, 52
起動 (手順), 56
使用する理由, 46
説明, 43
ツールの説明, 44
Solaris ユーザー登録, ユーザー登録を参照
SPARC システム, UFS フォーマット, 242
SPECFS ファイルシステム, 558
staff グループ, 81
Stop+A キー, 199, 201
stty コマンド, 100
SunOS のデフォルトファイルシステム, 560
SunSolve, 信頼される証明書, 289
SUNW接頭辞, 292
Sun ソフトウェアパッケージ
インストール, 330
追加 (例), 329
swapadd コマンド, 633
SWAPFS ファイルシステム, 558
swap コマンド, 635
sync コマンド, 201, 203
sync コマンドによるファイルシステムの同期
化, 201, 203
sysdef コマンド, 382

T

tar コマンド
1 枚のフロッピーディスクにコピー (手順)
 , 764
概要, 753
テープからファイルを取り出す (手順), 755
テープ上のファイルのリスト表示 (手順)
 , 755
ファイルをテープへコピー (手順), 754
ファイルをフロッピーディスクから取り出す
(手順), 766
ファイルをリモートテープから取り出す (dd)
(手順), 763

tar コマンド (続き)
ファイルをリモートテープにコピー (dd)(手
順), 762
フロッピーディスク上のファイルのリスト表
示 (手順), 765
TERMINFO 環境変数, 99
TERM 環境変数, 99
TMPFS ファイルシステム
概要, 556
作成 (手順), 575
/tmp ディレクトリ, 556, 561
tty (疑似), 75
tty タイプの疑似ユーザーログイン, 75
TZ 環境変数, 99

U

UDF ファイルシステム, 555
UFS CD, SPARC と x86 フォーマット, 242
ufsdump コマンド, 700
オプションと引数, 736
完全バックアップの例, 701, 703
機能, 731
制限, 735
増分バックアップの例, 702
データのコピー方法, 732
ファイルシステムをテープへバック
アップ, 700
メディアの終わりの検出, 732
リモートシステムへの完全バックアップ
例, 704
ufsrestore コマンド, 739
使用するテープの決定 (手順), 720
使用の準備 (概要), 718
対話式でない復元 (手順), 723
対話式の復元 (手順), 721
ファイルシステム全体の復元 (手順), 725
リモートテープドライブからの復元 (例)
 , 725
ルート (/) および /usr ファイルシステムの
復元 (手順), 728
UFS スナップショット
完全バックアップの作成, 714
完全バックアップの作成 (手順), 713
削除 (手順), 712
作成 (手順), 710
説明, 709

- UFS スナップショット (続き)
 - 増分バックアップの作成 (手順), 714
 - バックアップ (tar)(手順), 714
 - 表示 (手順), 711
- UFS ファイルシステム, 554, 562
 - /etc/vfstab によるマウント, 587
 - mount コマンドによるマウント, 588
 - 拡張基礎タイプ, 562
 - 作成 (手順), 573
 - 状態フラグ, 562
 - 大規模ファイルシステム, 562
 - 大規模ファイルを持たないマウント (手順), 589
 - マウント, 586
 - マウント解除 (手順), 594
- UFS ログイン、概要, 562
- UID, 86
 - 大きな値, 76
 - 定義, 75
 - 割り当て, 76
- umask コマンド, 101
- umountall コマンド, 559
- umount コマンド, 559
- UNIX グループ, 80
- UNIX ファイルシステム, 554
- USB カメラ、追加 (手順), 430
- USB デバイス
 - Solaris USB Architecture (USB A), 419
 - USB カメラの追加 (手順), 430
 - USB 大容量ストレージデバイスの追加
 - vold を使用, 425
 - vold を使用しない, 425
 - USB 大容量ストレージデバイスの取り外し
 - vold を使用しない, 426
 - USB デバイスサブツリーの接続 (手順), 442
 - USB デバイスの構成解除 (手順), 441
 - USB デバイスの構成 (手順), 441
 - USB デバイスの接続 (手順), 442
 - USB デバイスのリセット (手順), 443
 - vold を使用した USB 大容量ストレージデバイスのマウント解除 (手順), 428
 - vold を使用した USB 大容量ストレージデバイスのマウント (手順), 428
 - vold を使用しない USB 大容量ストレージデバイスのマウント解除 (手順), 429
 - vold を使用しない USB 大容量ストレージデバイスのマウント (手順), 429
- USB デバイス (続き)
 - オーディオ
 - 概要, 432
 - 主デバイスの変更 (手順), 435
 - 主要デバイスの識別 (手順), 433
 - 使用されていないデバイスのリンクの削除 (手順), 436
 - 追加, 433
 - デバイスの所有権, 437
 - オーディオデバイスの問題の障害追跡, 437
 - 概要, 416
 - キーボードとマウスデバイス, 420
 - 合成デバイス, 418
 - サポート, 416
 - サポートするデバイスの最大数, 416
 - ストレージデバイス, 424
 - 大容量ストレージデバイスを取り外す
 - vold を使用, 426
 - デバイスクラス, 419
 - デバイス情報の表示 (手順), 439
 - デバイスノード, 419
 - 電源管理, 422
 - ドライバ, 419
 - 名前, 418
 - バスの説明, 417
 - 複合デバイス, 418
 - 物理デバイスの階層, 417
 - ホストコントローラとルートハブ, 421
 - ホットプラグ (概要), 425
 - 略語, 417
 - /usr ファイルシステム, 560
 - uucp グループ, 75

V

- /var/sadm/install/admin ディレクトリ, 293
- /var/sadm/patch, 370
- /var/spool/pkg ディレクトリ, 331, 333
- /var ディレクトリ, 561
- vfstab ファイル, 569, 633
 - LOFS のエントリ, 577
 - すべてのファイルをマウント, 586
 - スワップの追加, 633
 - デフォルト, 567
 - にエントリを追加する (手順), 585
 - ファイルシステム名の検索, 698

volcopy コマンド, 559
volmgt start コマンド, 244

W

who コマンド, 162, 179

X

X.509, 286
x86 システム, UFS フォーマット, 242

あ

アーカイブ
SunOS の旧リリース用 (cpio)(手順), 768
ファイルを複数のフロッピーディスクにアーカイブする (cpio)(手順), 767
空き i ノード, 644
空きブロック, 644, 668
空き容量 (最小), 671
アクセス
ディスクデバイス, 447, 450
テープデバイス, 450
リムーバブルメディア (手順), 244
リムーバブルメディアのパス名, 236
アクセス権, 101
アプライアンス, 定義, 127
暗号化, 88

い

一次グループ, 81
一時ファイルシステム (TMPFS), 概要, 556

お

オプション, ufsdump コマンド, 736
音楽用 CD/DVD の再生, 245

か

カートリッジテープ, たるみを直す, 775
解決方法, SCSI 構成解除操作の失敗 (手順), 401
書き込み, データ CD とオーディオ CD (概要), 269
鍵、ユーザー、ユーザー鍵を参照
拡張基礎タイプ (UFS ファイルシステム), 562
確認
CD メディア (手順), 270
nfsd デーモンが実行している, 248
pkginfo コマンドによるソフトウェアパッケージのインストール, 331
PROM リビジョンレベル, 189
システム上のディスク (手順), 477
システムにログインしているユーザー, 179
システムの実行レベル (手順), 162
ソフトウェアのインストール (例), 335
ソフトウェアパッケージのインストール
pkginfo コマンド, 331
ファイルシステム名, 719
マウント済みのファイルシステム, 585
仮想ファイルシステムテーブル, 567
仮想メモリー記憶域、定義, 630
環境変数
LOGNAME, 98
LPDEST, 98
PATH, 98, 100
SHELL, 99
TZ, 99
説明, 96, 101
間接ブロック, 646
完全バックアップ
作成 (ufsdump), 700
定義, 684
必要なテープの本数, 699
リモートシステムへ
例, 704
例, 701, 703

き

キーストア, 285
記憶域 (仮想メモリー)、定義, 630
記憶容量 (メディア), 684, 771
疑似 tty, 75
疑似ファイルシステム, 概要, 555
疑似ユーザーログイン, 75

起動

- nfsd デーモン, 249
- ディスクレスクライアント (手順), 143
- ボリューム管理 (手順), 244

キャラクタ型特殊 i ノード, 644

強制終了

- ファイルシステムを使用中のすべてのプロセス (手順), 593
- リムーバブルメディアにアクセス中のプロセス (手順), 247

共有

- ファイル, 568
- ユーザーのホームディレクトリ, 113
- ユーザーのホームディレクトリ (手順), 112
- リムーバブルメディア (手順), 248

記録

- 増分バックアップ, 733
- ダンプ, 732

く

クリーンなシャットダウン, 178

グループ

- ID 番号, 75, 80, 81
 - UNIX, 80
 - アクセス権の設定, 101
 - 一次, 81
 - 一次グループの変更, 81
 - 管理のガイドライン, 80, 81
 - 情報の格納, 88, 91
 - 説明, 73, 80
 - デフォルト, 81
 - 名前
 - 説明, 80
 - 名前の説明, 80
 - 二次, 81
 - ネームサービス, 81
 - ユーザーが所属するグループの表示, 81
- グループ ID 番号, 75, 80, 81

け

- 決定, テープデバイスのタイプ, 719
- 検査, インストール済みのパッケージ (例), 335
- 検出, メディアの終わり
 - ufsdump コマンド, 732

こ

交換, SCSI コントローラ上の同一デバイス (手順), 397

構成

- SCSI コントローラ、cfgadm コマンドを使用 (手順), 393
- SCSI デバイス、cfgadm コマンドを使用 (手順), 394
- USB デバイス (手順), 441

構成解除

- SCSI コントローラ、cfgadm コマンドを使用 (手順), 393
- USB デバイス (手順), 441

構造、シリンダグループの, 666

構文

- fsck コマンド, 655
- newfs, 673

コピー

- CD (手順), 277
- 個別のファイル (cpio)(概要), 750
- ディレクトリ内のすべてのファイルをテープにコピー (cpio)(手順), 757
- ファイルシステム間でディレクトリをコピー (cpio)(概要), 750
- ファイルシステム間でディレクトリをコピー (cpio)(手順), 751
- ファイルシステム全体 (dd), 747
- ファイルのグループ (cpio)(概要), 750
- ファイルをテープへコピー (pax)(手順), 756
- ファイルをテープへコピー (tar)(手順), 754
- ファイルをフロッピーディスクにコピー (tar)(手順), 764
- ファイルをリモートテープにコピー (tar と dd)(手順), 761
- フロッピーディスクにファイルをコピー (概要), 764
- 別のヘッダー形式でファイルをコピー (cpio)(手順), 767
- リムーバブルメディア上の情報 (手順), 245

さ

サーバー

- OS サーバー, 131
 - 説明, 125
- サーバーとクライアントのサポート, 説明, 124
- サービスの起動と停止, 171

- 再構成ブート, 492
 - x86 の例, 514
 - SPARC の例, 502
- 最小空き容量, 671
- 最小値, ユーザーログイン名の長さ, 74
- 最小文字数, ユーザーパスワードの長さ, 78
- サイズ
 - i ノード, 646
 - ファイルシステムのチェック, 643
 - フラグメント, 670
- 再設定, SPARC システム, 193
- 最大数
 - USB デバイスサポート, 416
 - ユーザーが所属できる二次グループ, 81
- 最大値
 - ユーザー ID 番号, 75
 - ユーザーログイン名の長さ, 74
- 最大文字数, ユーザーパスワードの長さ, 78
- 最適化のタイプ, 671
- サイト初期設定ファイル, 95
- 削除
 - CacheFS ファイルシステム (手順), 609
 - Solaris Product Registry によるソフトウェアの削除, 299
 - UFS スナップショット情報
 - 例, 712
 - UFS スナップショット情報 (手順), 712
 - 管理ファイルによるパッケージの削除, 294
 - 使用されていない USB デバイスのリンク (手順), 436
 - スワップファイル, 638
 - ソフトウェアパッケージ
 - ガイドライン, 292
 - ソフトウェアパッケージ (手順), 335
 - ディスクレスクライアント OS サービス (手順), 144
 - ディスクレスクライアント OS サービス (例), 144
 - ユーザーのホームディレクトリ, 87
 - ユーザーのメールボックス, 87
- 作成
 - Solaris fdisk パーティション (手順), 515
 - UFS スナップショット
 - 例, 710
 - UFS スナップショット情報の完全バックアップ (手順), 713
 - UFS スナップショット (手順), 710

- 作成 (続き)
 - UFS スナップショットの増分バックアップ (手順), 714
 - UFS ファイルシステム (手順), 573
 - 一時ファイルシステム (TMPFS) (手順), 575
 - オーディオ CD (手順), 275
 - 互換性のあるアーカイブ (cpio)(手順), 768
 - スワップファイル, 635
 - ディスクスライスとディスクラベル (手順)
 - SPARC, 503
 - x86, 521
 - データ CD のファイルシステム (手順), 272
 - パッキングリスト (手順), 615
 - ファイルシステム (概要), 572
 - マルチセッションのデータ CD (手順), 273
 - ループバックファイルシステム (概要), 576
- サポートされていないデバイス, 380

- し
- シェル
 - 環境, 96, 99
 - 環境変数, 96, 97, 101
 - 基本機能, 96
 - ユーザー初期設定ファイル, 93, 95, 96, 102, 108
 - ローカル変数, 97, 99
- シェル変数, 97, 99
- シェル変数のエクスポート, 97
- 時間 (最適化のタイプ), 671
- 時間帯の環境変数, 99
- 磁気テープカートリッジ
 - たるみを直す, 775
 - 巻き戻し, 776
- 識別, デバイス, 381
- システムアカウント, 75
- システムシャットダウンコマンド, 178
- システム初期設定ファイル, 80
- システムタイプ
 - AutoClient, 126
 - アプライアンス, 127
 - 概要, 124
 - サーバー, 125
 - スタンドアロンシステム, 125
 - 選択のガイドライン, 127
 - ディスクレスクライアント, 126

- システムディスク
 - 接続 (手順)
 - SPARC, 501
 - x86, 513
 - 説明, 466
 - ブートブロックのインストール (手順)
 - SPARC, 508
 - x86, 523
- 実行制御スクリプト, 166
 - サービスの起動と停止, 171
 - 追加 (手順), 172
 - 無効 (手順), 173
- 実行レベル
 - 0 (電源切断レベル), 162
 - 1 (シングルユーザーレベル), 162
 - 2 (マルチユーザーレベル), 162
 - 3 (NFS を使用できるマルチユーザー)
 - ブート, 194
 - 3 (NFS を使用できるマルチユーザー状態)
 - ブート, 208
 - 3 (NFS を使用できるマルチユーザーレベル)
 - , 162
 - システムが移行すると実行される処理, 165
 - 実行されるプロセス, 165
 - 6 (リブートレベル), 162
 - s または S (シングルユーザー状態)
 - ブート, 195
 - s または S (シングルユーザーレベル), 162
 - ブート, 209
 - 確認 (手順), 162
 - 定義, 161
 - デフォルトの実行レベル, 161
- 自動構成プロセス, 379
- 自動マウント
 - と /home, 569
 - ユーザーのホームディレクトリ, 80
- シャットダウン
 - shutdown と init コマンドによるクリーンなシャットダウン, 178
 - サーバー (手順), 179
 - システム、ガイドライン, 156
 - スタンドアロンシステム (手順), 183
- シャットダウン時間のユーザーへの通知, 179
- 収集, CacheFS の統計情報 (概要), 622
- 重複ブロック, 646
- 終了
 - ファイルシステムのすべてのプロセス (手順), 593
 - ボリューム管理 (手順), 243
- 手動によるマウント, リモートのメディア (手順), 252
- 準備
 - バックアップ, 698
 - ファイルの復元 (概要), 718
- 障害追跡
 - cachefspack エラー, 618
 - PCI 構成の問題, 405
 - SCSI 構成の問題, 399
 - USB オーディオデバイスの問題, 437
 - ディスクレスクライアントの問題, 148
- 消去, CD-RW メディア (手順), 278
- 状態フラグ
 - fsck, 640
 - UFS ファイルシステム, 562
- 証明書
 - 削除, 325
 - 表示, 324
- 証明書、信頼される
 - インポート, 322
 - 概要, 286
 - 取得, 289
 - 定義, 285
- 証明書発行局, 289
- 初期設定ファイル, システム, 80
- 署名付きパッチ
 - パッチも参照
 - 最適な追加方法, 346
 - ダウンロード (手順), 353
 - 追加 patchadd (手順), 354
- 調べる
 - 完全バックアップに必要なテープの本数, 699
 - テープデバイス名, 719
 - テープドライブのタイプ, 774
- シリンダグループ, 666
- シングルユーザーレベル, 実行レベル s または S を参照
- シンボリックリンク, 645
- す
 - スーパーブロック, 643, 653, 666

スーパーユーザー (ルート)、なる, 50
スーパーユーザー (ルート) になる, 50
スーパーユーザー (ルート) のパスワード、忘れる
SPARC, 201
x86, 216
スケルトンディレクトリ (/etc/skel), 80, 87
スタンドアロンシステム, 説明, 125
スプールディレクトリ
ソフトウェアパッケージのインストール (手順), 331
ソフトウェアパッケージのインストール (例), 333, 335
すべてのデバイスの電源を落とす、手順, 185
スライス (定義), 462
スワップパーティション、定義, 630
スワップファイル
vfstab へ追加, 633
削除, 638
作成, 635
表示, 634

せ

制限, リムーバブルメディアへのアクセス (手順), 270
整合性が失われたファイルシステムの修正, 652
セキュリティ, ユーザー ID 番号の再利用, 76
接続
SCSI コントローラ (手順), 396
USB デバイス (手順), 442
接続解除
SCSI コントローラ (手順), 395
USB デバイスのサブツリー (手順), 442
設定, CacheFS ログイン, 623

そ

増分バックアップ, 684, 733
例, 702
その他 (アクセス権の設定), 101
ソフトウェア管理
ツール, 290
パッケージ, 284
パッケージの命名規則, 292

ソフトウェアパッケージ
インストール, 333
スプールディレクトリからのインストール (例), 332

た

大規模ファイルシステムオプション, 582
対話式
ファイルシステムのチェック, 650
復元 (手順), 721
ダンプレベル
定義, 687
日単位個別バックアップ, 688
日単位累積バックアップ, 687

ち

チェック
CacheFS ファイルシステム (手順), 611
CacheFS ファイルシステム (例), 611
CD メディア (手順), 271
i ノードの形式とタイプ, 644
i ノードリストの整合性, 643
空き i ノード, 644
空きブロック, 644
ファイルシステムのサイズ, 643
ファイルシステムの修復と, 649
ファイルシステムを対話式でチェック, 650
抽出, CD 上のオーディオトラック (手順), 276
調査
ファイルシステムタイプ, 569
ファイルシステムのタイプ, 569
直接入出力, 564

つ

追加
/etc/vfstab ファイルへのエントリ (手順), 585
Solaris Product Registry によるソフトウェアの追加, 299
USB オーディオデバイス, 433
USB カメラ (手順), 430

追加 (続き)

- USB 大容量ストレージデバイスに `vold` を使用, 425
 - USB 大容量ストレージデバイスに `vold` を使用しない, 425
 - 管理ファイルによるパッケージの追加, 293
 - サーバーとクライアントのサポート
 - 説明, 124
 - 実行制御スクリプト (手順), 172
 - 周辺デバイス (手順), 385
 - スプールディレクトリからのパッケージの追加 (例), 333
 - スプールディレクトリへのパッケージの追加 (例), 335
 - スワップを `vfstab` へ追加, 633
 - ディスク (概要)
 - SPARC, 501
 - x86, 512
 - ディスクレスクライアント OS サービス (手順), 140
 - ディスクレスクライアントの OS サービスの追加準備 (手順), 138
 - デバイスドライバ (手順), 386
 - パッケージ (前提条件), 292
 - パッケージ、例, 330
 - 複数バージョンのパッケージ, 293
 - ベースディレクトリによるパッケージの追加, 294
 - マウント済みの CD からパッケージを (例), 329
 - ユーザー初期設定ファイル, 87
 - リモートパッケージサーバーからパッケージを (例), 330
- 通常の `i` ノード, 644

て

停止

- CacheFS ロギング, 624
- 復元のためにシステムを
 - SPARC, 198
- 復元のためにシステムを (手順)
 - x86, 213
- リムーバブルメディアにアクセス中のプロセスの強制終了 (手順), 247

ディスク

- SCSI ドライブの自動構成, 491

ディスク (続き)

- 新しいディスク上でのファイルシステムの作成 (手順)
 - x86, 523
 - 欠陥セクターの修復, 493, 495
 - システム上のディスクの確認 (手順), 477
 - システムディスクの接続
 - x86, 513
 - システムディスクの接続 (手順)
 - SPARC, 501
 - 新規ディスクでのファイルシステムの作成 (手順)
 - SPARC, 508
 - スライス情報の表示 (手順), 481
 - 追加 (概要)
 - SPARC, 501
 - x86, 512
 - ディスクスライスとディスクラベルの作成 (手順)
 - SPARC, 503
 - x86, 521
 - ディスクスライスとディスクラベルの作成 (例)
 - SPARC, 504
 - ディスクラベルの検査 (手順), 486
 - 二次ディスクの接続 (手順)
 - SPARC, 502
 - x86, 513
 - 二次ディスクの接続 (例)
 - SPARC, 505
 - 破損したディスクラベルの復元 (概要), 487
 - 破損したディスクラベルの復元 (手順), 487
 - フォーマット, 470
 - フォーマットされているかを調べる (手順), 479
 - フォーマットする場合 (概要), 478
 - フォーマット (手順), 480
 - ラベルの作成 (手順), 483
- ディスクコントローラ, 448
- ディスクスライス
 - システム構成の要件, 467
 - 使用するスライスの決定, 467
 - 情報の表示 (概要), 481
 - 定義, 462
- ディスクスライスの指定, 448, 450
- ディスクのコピー, `dd` コマンド (手順), 748
- ディスクのフォーマットの概要, 470
- ディスクベースのファイルシステム, 553

- ディスクラベル
 - 検査 prtvtoc コマンド (手順), 486
 - 作成 (概要), 483
 - 説明, 471
 - ディスクレスクライアント
 - OS サービスの削除 (手順), 144
 - OS サービスの削除 (例), 144
 - OS サービスの追加準備 (手順), 138
 - OS サービスの追加 (手順), 140
 - 起動 (手順), 143
 - 定義, 126
 - ディスクレスクライアント管理コマンド
 - smossservice
 - OS サービスの追加, 131
 - ディレクトリ
 - i ノード, 644
 - PATH 環境変数, 98, 99, 100
 - /proc, 557
 - /tmp, 556
 - アクセス制御, 73, 101
 - スケルトン, 80, 87
 - ファイルシステム間でコピー (cpio)(概要), 750
 - ベースディレクトリ (basedir), 293, 294
 - ホーム, 79
 - 未割り当てブロック, 647
 - ユーザーアカウント用の所有権の変更, 86
 - データディレクトリブロック, 647
 - データブロック, 648, 668
 - テープ, 776
 - tar コマンドによるファイルのリスト表示 (手順), 755
 - 記憶容量, 684, 771
 - サイズ, 684, 771
 - 指定したファイルを取り出す (cpio)(手順), 760
 - 性質, 735
 - ディレクトリ内のすべてのファイルをコピー (cpio)(手順), 757
 - ファイルを取り出す (cpio)(手順), 759
 - ファイルを取り出す (tar)(手順), 755
 - 容量, 735
 - テープ (磁気カートリッジ), たるみを直す, 775
 - テープデバイス, 復元するタイプの決定, 719
 - テープデバイス (命名), 451
 - テープドライブ
 - 管理, 776
 - 最大の SCSI テープドライブ数, 773
 - テープドライブ (続き)
 - タイプを調べる, 774
 - 巻き戻し, 773
 - リモートからの復元 (例), 725
 - テープドライブの管理, 776
 - テープドライブのタイプ, 774
 - デバイス
 - アクセス, 445
 - 電源を落とす場合, 184
 - デバイスドライバ
 - 追加, 386
 - 定義, 378
 - デバイスのインスタンス名, 446
 - デバイスの電源を落とす場合, 184
 - デバイス名
 - テープデバイス名を調べる, 719
 - バックアップ, 772
 - ファイルシステム名の確認, 719
 - デフォルト
 - SunOS ファイルシステム, 560
 - /tmp 用ファイルシステム (TMPFS), 556
 - マウントオプション, 588
- と
- 動的再構成, 387
 - 取り出し
 - bar コマンドで作成したファイル (手順), 768
 - 指定したファイルをテープから取り出す (cpio)(手順), 760
 - テープからファイルを取り出す (cpio)(手順), 759
 - テープからファイルを取り出す (tar)(手順), 755
 - ファイルをフロッピーディスクから取り出す (tar)(手順), 766
 - ファイルをリモートテープから取り出す (tar と dd)(手順), 763
 - リムーバブルメディア (手順), 247
 - 取り付ける
 - PCI アダプタカード (手順), 403
 - SCSI デバイスを SCSI バスに (手順), 396
 - 取り外し, USB 大容量ストレージデバイスに vold を使用しない, 426
 - 取り外す
 - PCI アダプタカード (手順), 403

- 取り外す (続き)
 - SCSI デバイス (手順), 398
 - USB 大容量ストレージデバイス、vold を使用, 426
- な
- 名前
 - SUNW 接頭辞, 292
 - グループ
 - 説明, 80
 - ソフトウェアパッケージの命名規則, 292
 - ユーザーログイン
 - 説明, 73, 74
 - 変更, 86
- に
- 二次グループ, 81
- 二次ディスク
 - システムへの接続 (手順)
 - SPARC, 502
 - x86, 514
 - 説明, 466
- 入出力、直接, 564
- ね
- ネームサービス
 - グループ, 81
 - ユーザーアカウント, 74, 88
- ネットワークベースのファイルシステム, 554
- は
- パーティション (スワップ)、定義, 630
- バイト (i ノードごとの数), 672
- 場所を調べる, CacheFS ログファイル, 624
- バス指向コントローラ, 449
- バス指向ディスクコントローラ, 449, 450
- パスワード (ユーザー)
 - *LK* パスワード, 87, 91
 - NP パスワード, 91
 - 暗号化, 88
- パスワード (ユーザー) (続き)
 - 設定, 78, 87
 - 説明, 73, 78, 79
 - 選択, 78
 - 注意点, 78
 - 変更
 - 頻度, 78, 91
 - ユーザーツール, 87
 - ユーザーによる変更, 78
 - 有効期限, 79, 87, 88, 91
 - ユーザーアカウントの無効化/ロック, 87, 91
 - ユーザーツール, 87
- バックアップ
 - UFS スナップショット (tar)(手順), 714
 - UFS スナップショット (完全)(手順), 714
 - UFS スナップショット情報 (増分)(手順), 714
 - 準備, 698
 - 増分バックアップの記録, 733
 - タイプ, 684
 - テープへの完全 (レベル 0) バックアップ, 700
 - デバイス名, 772
 - ファイルシステムの選択, 682
 - ファイルシステムをテープへ, 700
 - フルバックアップおよび増分バックアップの定義, 684
 - 理由, 681
- バックアップスケジュール, 686
 - ガイドライン, 686
 - サーバー用, 691
 - 推奨事項, 694
 - ダンプレベルの使用, 687
 - 日単位増分、週単位累積バックアップ, 690
 - 日単位累積、週単位増分バックアップ, 689
 - 日単位累積、週単位累積バックアップ, 688
 - 例, 688, 694
- バックアップと復元
 - ファイルシステム
 - コマンド, 680
 - 定義, 680
- パッケージ
 - 概要, 284
 - 署名付き
 - パッケージ、署名付きを参照
 - 追加
 - pkgadd コマンドも参照
 - 定義, 284
- パッケージキーストア、セットアップ, 289

パッケージ、署名付き

- 概要, 285
- 証明書の削除, 325
- 証明書の情報の表示, 324
- 信頼される証明書のインポート, 322
- 追加, 326

パッチ

- Sun Service カスタマが利用できるもの, 343
- WWW (World Wide Web) からアクセス, 343
- 一般的な利用可能性, 343
- インストール済みのものを調べる, 369
- インストール用 README, 344
- 管理, 349
- 削除, 372
- 署名付き, 282
 - 追加, 285
 - 定義, 342
- ツールとコマンド (概要), 344
- 定義, 341
- ディスク容量の要件, 350
- 入手先, 343
- 番号付けの方式, 343

パッチ、署名付き、パッチを参照

パラメータ (ファイルシステム), 669

ひ

日単位個別バックアップ, 688

日単位累積バックアップ, 687

表示

- CacheFS 統計情報, 626
- PCI デバイス情報 (手順), 402
- SCSI デバイス構成情報 (手順), 392
- UFS スナップショットの情報 (手順), 711
- USB デバイス情報 (手順), 439
- インストール済みのソフトウェアの情報, 333
- 環境変数, 96
- 作業セット(キャッシュ)のサイズ, 625
- システム構成情報, 380, 382
- スワップ空間, 634
- ディスクスライス情報 (概要), 481
- デバイス情報, 383
- バックされたファイル (手順), 614
- バックされたファイル (例), 614
- パッケージ情報 (例), 333
- パッケージに関する詳細情報 (例), 334

表示 (続き)

- ユーザーマスク, 101
- リムーバブルメディアを使用しているプロセス (手順), 247

ふ

ファイル

- /etc/default/fs, 569
 - /etc/dfs/fstypes, 569
 - /proc ディレクトリの中, 557
 - tar コマンドによるテープ上のファイルのリスト表示 (手順), 755
 - アクセス制御, 73, 101
 - 新しくインストールしたパッケージの属性の確認, 335
 - 共有, 568
 - 対話式でない復元 (手順), 723
 - 対話式の復元 (手順), 721
 - テープから取り出す (cpio)(手順), 759, 760
 - テープから取り出す (tar)(手順), 755
 - テープ上のファイルのリスト表示 (cpio)(手順), 759
 - テープヘコピー (cpio)(手順), 757
 - テープヘコピー (pax)(手順), 756
 - テープヘコピー (tar)(手順), 754
 - 複数のフロッピーディスクにアーカイブする (cpio)(手順), 767
 - フロッピーディスクから取り出す (tar)(手順), 766
 - フロッピーディスク上のファイルのリスト表示 (tar)(手順), 765
 - フロッピーディスクにコピー (tar)(手順), 764
 - メディアへのコピー用コマンド (概要), 746
 - ユーザーアカウント用の所有権の変更, 86
- ### ファイルシステム
- /, 560
 - 4.3 Tahoe, 554
 - BSD Fat Fast, 554
 - DOS, 554
 - /export/home, 561
 - FDFS, 558
 - FIFOFS, 558
 - High Sierra, 554
 - ISO 9660, 554
 - MNTFS, 561

ファイルシステム (続き)

- NAMEFS, 558
- NFS のマウント (手順), 590
- /opt, 561
- PCFS, 554
- /proc, 561
- PROCFS、概要, 557
- SPECFS, 558
- SunOS のデフォルト, 560
- SWAPFS, 558
- TMPFS, 556
- UFS, 554
- UNIX, 554
- /usr, 560
- /var, 561
- カスタムパラメータ, 669
- 管理コマンドの説明, 558
- 疑似、概要, 555
- キャッシュ (概要), 599
- 共有, 568
- サイズのチェック, 643
- 作成 (概要)
 - ループバック (LOFS), 576
- 作成 (手順)
 - TMPFS, 575
 - UFS, 573
- 修正, 652
- 修復, 651
- 使用中のすべてのプロセスを終了 (手順)
 - , 593
- シリンダグループの構造, 666
- 整合性がなくなる原因, 642
- 全体のコピー (dd), 747
- 全体の復元, 725
- 全体の復元 (手順), 725
- 大規模, 582
- タイプ, 553
- タイプの調査, 569
- 対話式でチェック, 650
- チェックと修復, 649
- ディスクベースの, 553
- ネットワークベース, 554
- 破損, 640
- バックアップ, 682
- バックアップする理由, 681
- プロセス、概要, 557
- マウント解除 (手順), 594
- マウントテーブル, 566

ファイルシステム (続き)

- マニュアルページ, 560
- 利用可能にする (概要), 579
- ファイルシステムテーブル, 仮想, 567
- ファイルシステムの修復, 651
- ファイルシステムの整合性がなくなる, 642
- ファイルシステムのタイプ, 553
- ファイルシステムの破損, 640
- ファイルシステムの復元
 - root および /usr (x86) (例), 730
 - root および /usr (手順), 728
 - 準備 (概要), 718
 - 使用するテープの決定 (手順), 720
 - 全体 (例), 725, 727
 - テープドライブのタイプ, 719
 - ルートと /usr (SPARC) (例), 730
- ファイルシステム破損の原因, 640
- ファイルシステム名, 719
- ファイルシステム用のカスタムパラメータ, 669
- ファイルとディレクトリへのアクセス制御, 73, 101
- ファイルの復元
 - 対話式 (手順), 721
 - 対話式でない (手順), 723
 - 対話式でない復元の例, 724
 - 対話式復元の例, 722
 - リモートドライブから (例), 725
- ブート
 - PC BIOS, 222
 - Solaris Device Configuration Assistant (デバイス構成用補助) (手順)
 - x86, 207
 - カーネルデバッグを使って (手順)
 - SPARC, 201
 - x86, 218
 - クラッシュダンプを強制してリブート (手順)
 - SPARC, 202
 - x86, 219
 - システム、ガイドライン, 156
- 実行レベル 3
 - SPARC, 194
- 実行レベル 3 (手順)
 - x86, 208
- 実行レベル S で
 - SPARC, 195
- 実行レベル S (手順)
 - x86, 209

- ブート (続き)
 - 対話式 (手順)
 - SPARC, 196
 - x86, 210
 - ネットワークから
 - SPARC, 197
 - x86, 212
 - 復元のために (手順)
 - SPARC, 199
 - x86, 213
- ブートタイプ、説明, 155
- ブートプロセス
 - x86, 228
 - 説明 (SPARC), 222
- ブートブロック, 666
- ブートブロックのインストール (手順)
 - SPARC, 508
 - x86, 523
- 複数バージョンのソフトウェアパッケージ, 293, 294
- 不正な「.」と「..」 エントリ, 647
- 不正な i ノード番号, 647
- 不正なスーパーブロック, 653
- 不正なスーパーブロックの復元, 653
- 不正なブロック番号, 646
- 物理デバイス名、定義, 446
- プライマリ管理者の役割
 - 作成 (概要), 54
 - 作成 (手順), 55
 - 引き受ける (手順), 55
- フラグメントサイズ, 670
- プロセスファイルシステム (PROCFS), 557
- ブロック
 - 空き, 668
 - 間接, 646
 - 重複, 646
 - 通常ファイル, 648
 - ディレクトリデータ, 647
 - データ, 668
 - 特殊 i ノード, 644
 - ブート, 666
 - 不正, 646
 - 論理サイズ, 669
- ブロックディスクデバイスインタフェース
 - 使用する場合, 448
 - 定義, 447

- フロッピーディスク
 - 他のシステム上のフロッピーディスクへのアクセス (例), 253
 - ファイルのリスト表示 (tar)(手順), 765
 - ファイルをコピー (tar)(手順), 764
 - ファイルを取り出す (tar)(手順), 766
 - ファイルを複数のフロッピーディスクにアーカイブする (cpio)(手順), 767
 - ボリューム管理による読み込み (手順), 259
 - 読み込み
 - ボリューム管理の使用, 260
 - リモートでのマウント (例), 253

- へ
- ベースディレクトリ (basedir), 293, 294
- ヘッダー形式、別のヘッダー形式でファイルをコピー (cpio)(手順), 767
- 別名、ユーザーログイン名との比較, 75
- 変更
 - 主 USB オーディオデバイス (手順), 435
 - デフォルトのブートデバイス
 - SPARC, 191
 - ユーザー ID 番号, 86
 - ユーザーアカウント用のディレクトリ所有権の変更, 86
 - ユーザーアカウント用のファイル所有権の変更, 86
 - ユーザーパスワード
 - 頻度, 78, 91
 - ユーザーツール, 87
 - ユーザーによる変更, 78
 - ユーザーログイン名, 86
- 変数
 - 環境, 96, 101
 - シェル (ローカル), 97
 - シェル (ローカル) 変数, 99

- ほ
- ホットプラグ
 - cfgadm コマンドを使用して SCSI コントローラを切り離す (手順), 395
 - PCI アダプタカードを取り付ける (手順), 403
 - PCI アダプタカードを取り外す (手順), 403

ホットプラグ (続き)
PCI デバイス (概要), 402
SCSI コントローラ上の同一デバイスを交換する (手順), 397
SCSI コントローラの構成 (手順), 393
SCSI コントローラの再構成 (手順), 393
SCSI コントローラの接続 (手順), 396
SCSI デバイスの構成 (手順), 394
SCSI デバイスを SCSI バスに取り付ける (手順), 396
SCSI デバイスを取り外す (手順), 398
概要, 387
ボリューム管理
再起動 (手順), 244
終了 (手順), 243
手動によるマウントと自動マウントの比較, 235
フロッピーディスク
読み込み, 260
フロッピーディスクの読み込み (手順), 259
利点, 234
リムーバブルメディア
アクセス, 236

ま

マウント

/etc/vfstab によるファイルシステム, 587
NFS ファイルシステム, 586
NFS ファイルシステム (手順), 590
UFS ファイルシステム, 586
UFS ファイルシステム (手順)
大規模ファイルを持たない, 589
USB 大容量ストレージデバイス、vold を使用しない (手順), 429
USB 大容量ストレージデバイス、vold を使用 (手順), 429
vfstab ファイル内のすべてのファイル, 586
他のシステム上の PCMCIA メモリーカード (例), 253
他のシステム上のフロッピーディスク (例), 253
デフォルトオプションの使用, 588
ファイルシステムの自動マウント, 568

マウント (続き)

ユーザーのホームディレクトリ
自動マウント, 80
リモート, 113
ユーザーのホームディレクトリ (手順), 113
リムーバブルメディア
自動マウントとの比較, 235
リモートのメディア (手順), 252
リモートのリムーバブルメディアの手動によるマウント (例), 252
ループバックファイルシステム (LOFS), 586
マウント解除
USB 大容量ストレージデバイス、vold を使用しない (手順), 429
USB 大容量ストレージデバイス、vold を使用 (手順), 429
ファイルシステム (手順), 594
マウントテーブル, 566
マウントポイントの定義, 565
マニュアルページ、ファイルシステム用の, 560
マルチユーザーレベル, 実行レベル 3 を参照

み

未割り当て i ノード, 645
未割り当てディレクトリブロック, 647

む

無効

実行制御スクリプト (手順), 173
ユーザーアカウント
パスワード, 87, 91
ユーザーツール, 87

め

メール別名、ユーザーログイン名との比較, 75
メディアの終わりの検出
cpio コマンド, 750
ufsdump コマンド, 732
メモリー記憶域 (仮想)、定義, 630

も

モニター (PROM), 221

ゆ

ユーザー ID 番号, 75, 76, 86

ユーザーアカウント, 73

ID 番号, 75, 76, 86

ガイドライン, 74, 80

情報の格納, 74, 88

設定

記録シート, 106

説明, 73

ネームサービス, 74, 88

無効化/ロック

パスワード, 87, 91

ユーザーツール, 87

ログイン名, 73, 74, 86

ユーザー鍵, 285

ユーザー初期設定ファイル

Bourne シェル, 93

カスタマイズ, 93, 102

概要, 80, 93, 94

カスタマイズしたファイルを追加, 87

環境変数, 97, 101

サイト初期設定ファイル, 95

シェル変数, 97, 99

ユーザーマスクの設定, 101

ローカルシステムへの参照を避ける, 95

カスタマイズ (手順), 107

シェル, 93, 95, 96, 102

説明, 73, 80, 93

デフォルト, 94

例, 102

ユーザー初期設定ファイルのカスタマイズ (手順), 107

ユーザーツール

アカウントの無効化, 87

パスワードの管理, 87

ユーザー登録

solregis コマンド, 115

説明, 116

無効, 118

問題, 117

ユーザーの初期設定ファイル, 説明, 80

ユーザーのホームディレクトリ

カスタマイズした初期設定ファイル, 80, 87

ユーザーのホームディレクトリ (続き)

共有, 113

共有 (手順), 112

削除, 87

所有権の変更, 86

説明, 73, 79

マウント

自動マウント, 80

リモート, 113

マウント (手順), 113

ローカル以外からの \$HOME への参照, 95

ローカル以外の位置からの参照 (\$HOME), 79

ユーザーパスワードの有効期限, 79, 87, 88

ユーザーマスク, 101

ユーザーログイン (疑似), 75

ユーザーログイン名

説明, 73, 74

変更, 86

よ

読み込み

フロッピーディスク

ボリューム管理の使用, 260

ボリューム管理によるフロッピーディスクの読み込み (手順), 259

り

リスト表示

テープ上のファイル (cpio)(手順), 759

テープ上のファイル (tar)(手順), 755

フロッピーディスク上のファイル (tar)(手順), 765

リセット, USB デバイス (手順), 443

リムーバブルメディア

アクセス中のプロセスの強制終了 (手順), 247

アクセス (手順), 244

アクセス (例), 244

新しいドライブの準備 (手順), 243

音楽用 CD/DVD, 245

情報のコピー (手順), 245

他のシステム上のメディアへのアクセス (手順), 252

リムーバブルメディア (続き)
他のシステム上のメディアへのアクセス (例), 252
他のシステムで使用可能にする (手順), 248
取り出し (手順), 247
名前, 240
マウント
 手動によるマウントと自動マウントの比較, 235
メディアが使用中であるかどうかの確認 (手順), 247
リムーバブルメディアのマウント (例), 252
リモートのメディアのマウント (手順), 252
リモートドライブからの復元, 725
リモートパッケージサーバー
スプールディレクトリへのパッケージの追加 (例), 332
ソフトウェアのインストール, 330
ソフトウェアのインストール (例), 330
リモートマウント, 113
領域最適化タイプ, 671

る

ルート (/) ファイルシステム, 560
ルート (スーパーユーザー)、なる, 50
ルートのパスワードの復元 (手順)
 SPARC, 201
 x86, 216
ルートのパスワード、忘れる
 SPARC, 201
 x86, 216
ルートのパスワードを忘れる
 SPARC, 201
 x86, 216
ループバックファイルシステム (LOFS)
 作成 (概要), 576
 マウント, 586

れ

レベル 0 のバックアップ, 687

ろ

ログイン名 (ユーザー)
 説明, 74
 変更, 86
ログ (ダンプの記録), 732
論理デバイス名
 定義, 446
 ディスク, 447
 テープ, 450
リムーバブルメディア, 451
論理ブロックサイズ, 669

わ

ワールド (アクセス権), 101
割り当て済み i ノード, 645

