



Solaris のシステム管理 (第 1 卷)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part Number 806-2717-11
2001 年 5 月

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 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。

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービイマクス株式会社からライセンス供与されたタイプフェイスマスクをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスクをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、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, SunOS, Solstice, Solstice AdminSuite, Solstice DiskSuite, Solaris Solve, Java, JavaStation, DeskSet, OpenWindows, NFS 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・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、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK8」は株式会社ジャストシステムの著作物であり、「ATOK8」にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

「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) は郵政省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ない、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, Volume 1*

Part No: 805-7228-10

Revision A



目次

はじめに 27

1. ユーザーアカウントとグループの管理 33

2. ユーザーアカウントとグループの管理 (概要) 35

ユーザーとグループの管理における新機能 35

ユーザーアカウントとグループとは 36

ユーザーアカウント管理のガイドライン 37

 ネームサービス 37

 ユーザー (ログイン) 名 37

 ユーザー ID 番号 38

 パスワード 42

 パスワード有効期限 43

 ホームディレクトリ 43

 ユーザーの作業環境 44

グループを管理するガイドライン 45

ユーザーアカウントとグループを管理するツール 46

Admintool の機能 49

 ユーザーアカウントの変更 49

 ユーザーアカウントの削除 50

 カスタマイズしたユーザー初期設定ファイルの追加 50

	パスワードの管理	50
	ユーザーアカウントを無効にする	50
	ユーザーアカウントとグループ情報の格納場所	51
	パスワードファイルのフィールド	51
	shadow ファイルのフィールド	54
	group ファイルのフィールド	55
	ユーザーの作業環境のカスタマイズ	58
	サイト初期設定ファイルの使用方法	59
	ローカルシステムへの参照を避ける	60
	シェア機能	60
	シェア環境	61
	PATH 変数	65
	ロケール変数	66
	デフォルトのファイルアクセス権 (umask)	67
	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	68
	例 — サイト初期設定ファイル	70
3.	ユーザーアカウントとグループの設定と管理 (手順)	71
	スーパーユーザー (root) になる	72
	▼ スーパーユーザー (root) になる方法	72
	ユーザーアカウントの設定	73
	ユーザー情報データシート	74
	▼ ユーザー初期設定ファイルをカスタマイズする方法	75
	▼ Admintool を起動する方法	76
	▼ グループを追加する方法	77
	▼ 新しいユーザーアカウントを追加する方法	79
	▼ ユーザーのホームディレクトリを共有する方法	80
	▼ ユーザーのホームディレクトリをマウントする方法	82
	ユーザーアカウントの管理	84

- ▼ グループを変更する方法 85
- ▼ グループを削除する方法 86
- ▼ ユーザーアカウントを変更する方法 86
- ▼ ユーザーアカウントを無効にする方法 88
- ▼ ユーザーのパスワードを変更する方法 90
- ▼ ユーザーアカウントのパスワード有効期限を変更する方法 91
- ▼ ユーザーアカウントを削除する方法 93
- Solaris ユーザー登録 94
 - Solaris SolveSM へのアクセス 94
 - Solaris ユーザー登録の問題の解決 95
 - ▼ Solaris ユーザー登録をやり直す方法 96
 - ▼ ユーザー登録を無効にする方法 97
- 4. サーバーとクライアントサポートの管理 99
- 5. サーバーとクライアントサポートの管理 (概要) 101
 - サーバーとクライアントとは 101
 - サポートとは 102
 - システムタイプの概要 102
 - サーバー 103
 - スタンドアロンシステム 104
 - JavaStation クライアント 104
- 6. システムのシャットダウンとブート 105
- 7. システムのシャットダウンとブート (概要) 107
 - システムのシャットダウンとブートに関する新機能 107
 - DHCP によるネットワーク経由のシステムのブート 108
 - IA: Solaris ブートフロッピーディスクを使用しない CD-ROM からのブート 108
 - シャットダウンとブートについての参照先 108
 - シャットダウンとブートの用語 109

	システムのシャットダウンに関するガイドライン	110
	システムのブートに関するガイドライン	110
	再構成用ブートの実行	111
	システムをシャットダウンする場合	111
	システムをブートする場合	112
8.	実行レベルとブートファイル (手順)	115
	実行レベル	115
	▼ システムの実行レベルを確認する方法	117
	/etc/inittab ファイル	117
	例 — デフォルトの inittab ファイル	118
	システムが実行レベル 3 になると実行される処理	119
	実行制御スクリプト	121
	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	122
	▼ 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	122
	実行制御スクリプトを追加する	123
	▼ 実行制御スクリプトを追加する方法	124
	実行制御スクリプトを無効にする	124
	▼ 実行制御スクリプトを無効にする方法	125
	実行制御スクリプトのまとめ	125
9.	システムのシャットダウン (手順)	129
	システムをシャットダウンする場合	129
	システムをシャットダウンするには	130
	デバイスの電源を落とす場合	131
	システムのダウンについてユーザーに通知する	131
	▼ システムにログインしているユーザーを知る方法	132
	▼ サーバーをシャットダウンする方法	132
	▼ スタンドアロンシステムをシャットダウンする方法	136
	▼ すべてのデバイスの電源を落とす方法	138

- 10. **SPARC: システムのブート (手順) 141**
 - SPARC: ブート PROM の使用方法 142
 - ▼ SPARC: ok プロンプトに切り換える方法 142
 - ▼ SPARC: システムの PROM リリースを確認する方法 142
 - ▼ SPARC: デフォルトのブートデバイスを変更する方法 143
 - ▼ SPARC: システムをリセットする方法 145
 - SPARC: システムのブート 145
 - ▼ SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法 147
 - ▼ SPARC: システムを実行レベル S (シングルユーザー状態) にする方法 148
 - ▼ SPARC: システムを対話式でブートする方法 149
 - ▼ SPARC: システムをネットワーク経由でブートする方法 151
 - ▼ SPARC: システムを復元するためにブートする方法 153
 - ▼ SPARC: 復元を目的としてシステムを停止する方法 155
 - 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 156
 - ▼ 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 156
 - ▼ SPARC: 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 157
- 11. **IA: システムのブート (手順) 159**
 - IA: システムのブート 159
 - IA: Solaris Device Configuration Assistant (デバイス構成用補助) のブート 161
 - ▼ IA: Solaris Device Configuration Assistant をブートする方法 162
 - ▼ IA: システムを実行レベル 3 (マルチユーザー状態) にする方法 162
 - ▼ IA: システムを実行レベル S (シングルユーザー状態) にする方法 163
 - ▼ IA: システムを対話式でブートする方法 164
 - ▼ IA: システムをネットワーク経由でブートする方法 166
 - ▼ IA: システムを復元するためにブートする方法 167
 - ▼ IA: 復元を目的としてシステムを停止する方法 171
 - ▼ IA: 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 171

- IA: クラッシュダンプを強制してシステムをリブートする 172
- ▼ IA: 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 172
- 12. ブートプロセス (参照情報) 175
 - SPARC: ブート PROM 175
 - SPARC: ブートプロセス 176
 - IA: PC BIOS 177
 - IA: ブートサブシステム 177
 - IA: Solaris のブート 178
 - IA: デバイス識別段階で表示される画面 179
 - IA: ブート段階で表示されるメニュー 181
 - IA: ブートプロセス 182
- 13. 取り外し可能な媒体の管理 185
- 14. CD とフロッピーディスクの使用方法 (概要) 187
 - 取り外し可能な媒体の管理についての参照先 187
 - 取り外し可能な媒体の機能と利点 188
 - 自動マウントと手作業によるマウントの比較 188
 - フロッピーディスクと CD で行える操作 189
- 15. コマンド行での CD の使用方法 (手順) 191
 - CD の使用方法 191
 - CD の名前について 192
 - ▼ CD をロードする方法 193
 - ▼ CD の内容を調べる方法 193
 - ▼ CD から情報をコピーする方法 194
 - ▼ CD が使用中かどうかを調べる方法 195
 - ▼ CD を取り出す方法 196
 - ▼ 他のシステム上の CD にアクセスする方法 196
 - ▼ ローカル CD を他のシステムで使用可能にする方法 198
 - ▼ 音楽 CD を演奏するようにシステムを設定する方法 201

- ▼ 新しい CD-ROM ドライブ用にシステムを準備する方法 202
- ボリューム管理の設定 203
- ▼ ボリューム管理を終了させる方法 203
- ▼ ボリューム管理を再起動する方法 203
- 16. コマンド行でのフロッピーディスクのフォーマットと使用方法 (手順) 205
 - フロッピーディスクのフォーマット 205
 - フロッピーディスクの名前について 206
 - ハードウェアの考慮点 207
 - ▼ UFS フロッピーディスクをフォーマットする方法 210
 - ▼ UFS ファイルシステムをフロッピーディスク上に作成する方法 213
 - ▼ DOS フロッピーディスクをフォーマットする方法 215
 - フロッピーディスクの使用方法 218
 - ▼ フロッピーディスクをロードする方法 218
 - ▼ フロッピーディスクの内容を調べる方法 221
 - ▼ フロッピーディスクから情報をコピーまたは移動する方法 221
 - ▼ フロッピーディスクへ情報をコピーまたは移動する方法 223
 - ▼ フロッピーディスクが使用中かどうかを調べる方法 224
 - ▼ フロッピーディスクを取り出す方法 225
 - ▼ 他のシステム上のフロッピーディスクにアクセスする方法 226
 - ▼ ローカルフロッピーディスクを他のシステムで使用可能にする方法 228
- 17. コマンド行での **PCMCIA** メモリーカードの使用方法 (手順) 233
 - PCMCIA メモリーカードのフォーマット 234
 - PCMCIA メモリーカードの名前について 234
 - ハードウェアの考慮点 235
 - ▼ UFS PCMCIA メモリーカードをフォーマットする方法 236
 - ▼ UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法 238
 - ▼ DOS PCMCIA メモリーカードをフォーマットする方法 241
 - PCMCIA メモリーカードの使用方法 243

- ▼ PCMCIA メモリーカードをロードする方法 244
 - ▼ PCMCIA メモリーカードの内容を調べる方法 246
 - ▼ PCMCIA メモリーカードから情報をコピーまたは移動する方法 247
 - ▼ PCMCIA メモリーカードへ情報をコピーまたは移動する方法 248
 - ▼ PCMCIA メモリーカードが使用中かどうかを調べる方法 249
 - ▼ PCMCIA メモリーカードを取り出す方法 250
 - ▼ 他のシステム上の PCMCIA メモリーカードにアクセスする方法 251
 - ▼ 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方法 253
- 18. ボリューム管理の動作 (参照情報) 257**
- すべての取り外し可能な媒体のマウント 257
 - フロッピーディスクへのアクセス 258
 - CD へのアクセス 259
 - アクセスを容易にするためのマウントポイント 260
 - 2 種類のシンボリックリンク 262
 - ファイルシステムアクセス用シンボリックリンク 262
 - raw デバイスアクセス用シンボリックリンク 262
 - UFS フォーマットによる制限 263
 - 混合 (ハイブリッド) フォーマットについて 264
- 19. ソフトウェアの管理 267**
- 20. ソフトウェアの管理 (概要) 269**
- ソフトウェア管理についての参照先 269
 - ソフトウェア管理における新機能 270
 - ソフトウェアパッケージの概要 270
 - ソフトウェア管理ツール 271
 - パッケージの追加または削除時の動作 272
 - パッケージを追加または削除する前に知っておくこと 273
 - クライアントソフトウェア管理に関するガイドライン 273
 - パッケージの削除に関するガイドライン 274

	パッケージ追加時のユーザーの対話操作を省略する	275
	管理ファイルの使用	275
	応答ファイルの使用	276
21.	ソフトウェアの管理 (手順)	277
	ソフトウェアパッケージを処理するコマンド	277
	パッケージの追加と削除における既知の問題	278
	パッケージの追加	278
	▼ スタンドアロンシステムにパッケージを追加する方法	278
	スプールディレクトリの使用	281
	▼ スプールディレクトリにパッケージを追加する方法	281
	パッケージのインストールの検査	284
	▼ インストールされたすべてのパッケージに関する情報を表示する方法	284
	▼ インストールされたパッケージが完全であるかどうかをチェックする方法	284
	▼ パッケージに関する詳細を表示する方法	286
	サーバーとスタンドアロンシステムからのパッケージの削除	287
	▼ パッケージを削除する方法	287
	▼ スプールされたパッケージを削除する方法	287
	Admintool を使用してソフトウェアパッケージを追加または削除する	288
	▼ Admintool を起動する方法	288
	▼ Admintool でパッケージを削除する方法	290
22.	パッチの管理 (概要)	293
	パッチについて	293
	パッチ管理用ツール	294
	パッチの配布	295
	Sun パッチにアクセスするために必要な条件	295
	World-Wide Web を介したパッチアクセス	296
	ftp を介したパッチアクセス	296
	パッチの番号付け	297

- パッチの追加に伴う動作 297
- パッチの削除に伴う動作 298
- 23. デバイスの管理 299
- 24. デバイスの管理 (概要) 301
 - デバイス管理における新機能 301
 - SCSI と PCI のホットプラグ機能 301
 - 改良されたデバイス構成 (devfsadm) 302
 - デバイス管理作業についての参照先 303
 - デバイスドライバについて 303
 - デバイスの自動構成 304
 - 機能と利点 305
 - 標準サポートされていないデバイスを使用する場合 305
 - デバイス構成情報の表示 306
 - driver not attached メッセージ 306
 - システムデバイスの識別 306
 - ▼ システム構成情報を表示する方法 307
 - ▼ デバイス情報を表示する方法 309
- 25. デバイスの構成 311
 - システムへ周辺デバイスを追加する 312
 - ▼ 周辺デバイスを追加する方法 312
 - ▼ デバイスドライバを追加する方法 314
 - 動的再構成とホットプラグ機能 315
 - 接続点 316
 - IA: PCI アダプタカードの取り外し 318
 - cfgadm コマンドによる SCSI ホットプラグ 319
 - ▼ すべてのデバイスの構成情報を表示する方法 319
 - ▼ SCSI コントローラの構成を解除する方法 320
 - ▼ SCSI コントローラを構成する方法 321

- ▼ SCSI デバイスを構成する方法 321
- ▼ SCSI コントローラを切り離す方法 322
- ▼ SCSI コントローラを接続する方法 324
- ▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法 324
- ▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法 326
- ▼ SPARC: SCSI デバイスを取り外す方法 327
 - SPARC: SCSI 構成の障害対処 328
- IA: cfgadm コマンドによる PCI ホットプラグ 329
- ▼ IA: PCI スロット構成情報を表示する方法 329
- ▼ IA: PCI アダプタカードを取り外す方法 330
- ▼ IA: PCI アダプタカードを取り付ける方法 331
 - IA: PCI 構成の障害対処 333
- 26. デバイスへのアクセス (概要) 335**
 - デバイスへのアクセス 335
 - デバイス情報が作成される方法 336
 - デバイス名の命名規則 336
 - 論理ディスクデバイス名 337
 - ディスクサブディレクトリの指定 337
 - スライスの指定 338
 - SPARC: 直接コントローラでアクセスされるディスク 339
 - IA: 直接コントローラでアクセスされるディスク 339
 - SPARC: バス指向コントローラでアクセスされるディスク 340
 - IA: SCSI コントローラでアクセスされるディスク 340
 - 論理テープデバイス名 341
 - 論理 CD-ROM デバイス名 342
- 27. ディスクの管理 343**
- 28. ディスクの管理 (概要) 345**
 - ディスク管理における新機能 345

大容量ディスクのサポート	346
ディスク管理作業についての参照先	346
概要	346
ディスク関連の用語	346
ディスクスライスについて	347
SPARC: ディスクスライス	348
IA: ディスクスライス	349
raw データスライスの使用	351
複数のディスク上のスライス配置	351
使用するスライスの決定	352
format ユーティリティ	353
定義	353
機能と利点	353
format ユーティリティを使用する場合	354
format ユーティリティ使用上のガイドライン	355
ディスクのフォーマット	356
ディスクラベルについて	357
パーティションテーブル	357
ディスクをスライスに分割する	361
free hog スライスの使用方法	361
29. ディスクの管理 (手順)	363
ディスクの管理	364
システム上のディスクの確認	365
▼ システム上のディスクを確認する方法	365
ディスクのフォーマット	367
▼ ディスクがフォーマット済みかどうかを調べる方法	368
▼ ディスクをフォーマットする方法	368
ディスクスライスの表示	370

- ▼ ディスクスライス情報を表示する方法 370
- ディスクラベルの作成と検査 373
- ▼ ディスクラベルを作成する方法 373
- ▼ ディスクラベルを検査する方法 375
- 破損したディスクラベルの復元 376
- ▼ 破損したディスクラベルを復元する方法 376
- サードパーティのディスクの追加 379
 - format.dat のエントリの作成 380
- ▼ format.dat のエントリを作成する方法 381
- SCSI ディスクドライブの自動構成 381
- ▼ SCSI ドライブを自動構成する方法 382
- 欠陥セクターの修復 384
- ▼ 表面解析を使用して欠陥セクターを調べる方法 385
- ▼ 欠陥セクターを修復する方法 386
- ディスク管理のヒント 387
 - format セッションのデバッグ 388
 - pdrvtop 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 388
- 30. **SPARC: ディスクの追加 (手順) 391**
 - SPARC: システムディスクと二次ディスクについて 391
 - SPARC: システムディスクまたは二次ディスクの追加 392
 - ▼ SPARC: システムディスクを接続してブートする方法 393
 - ▼ SPARC: 二次ディスクを接続してブートする方法 394
 - ▼ SPARC: ディスクスライスとディスクラベルを作成する方法 396
 - ▼ SPARC: ファイルシステムを作成する方法 400
 - ▼ 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 401
- 31. **IA: ディスクの追加 (手順) 403**
 - IA: システムディスクと二次ディスクについて 403

- IA: システムディスクまたは二次ディスクの追加 404
 - IA: fdisk パーティションの作成上のガイドライン 405
- ▼ IA: システムディスクを接続してブートする方法 406
- ▼ IA: 二次ディスクを接続してブートする方法 407
- ▼ IA: Solaris fdisk パーティションを作成する方法 408
- ▼ IA: ディスクスライスとディスクラベルを作成する方法 416
- ▼ IA: ファイルシステムを作成する方法 418
- ▼ IA: 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 419
- 32. format ユーティリティ (参照情報) 421
 - format ユーティリティを使用するための要件または制限 421
 - format を使用する場合に情報を保存するための推奨事項 422
 - format のメニューとコマンドの説明 422
 - partition メニュー 424
 - IA: fdisk メニュー 425
 - analyze メニュー 426
 - defect メニュー 428
 - format で使用されるファイル — format.dat 429
 - format.dat ファイルの構造 430
 - format.dat ファイルの構文 430
 - format.dat ファイル中のキーワード 431
 - パーティションまたはスライステーブル (format.dat) 433
 - format データファイルの位置を指定する 434
 - format コマンドへの入力規則 435
 - format コマンドへ数値を入力する 435
 - format コマンドへブロック番号を指定する 435
 - format のコマンド名を指定する 436
 - format コマンドへディスク名を指定する 437
 - format のヘルプを使用する 437

	format に関連するマニュアルページ	437
33.	ファイルシステムの管理	439
34.	ファイルシステムの管理 (概要)	441
	ファイルシステムにおける新機能	441
	/var/run ファイルシステム	441
	マウントテーブル (/etc/mnttab) の変更	442
	UDF (Universal Disk Format) ファイルシステムの使用	442
	UDF の特徴と利点	443
	ハードウェアとソフトウェアの要件	443
	▼ DVD-ROM デバイスを接続する方法	444
	▼ DVD-ROM デバイス上のファイルにアクセスする方法	444
	▼ UDF ファイルシステムのパラメータを表示する方法	445
	▼ UDF ファイルシステムを作成する方法	445
	▼ UDF ファイルシステムのタイプを識別する方法	446
	▼ UDF ファイルシステムを検査する方法	446
	▼ UDF ファイルシステムをマウントする方法	446
	▼ UDF ファイルシステムのマウントを解除する方法	447
	▼ UDF ファイルシステムのデバイスにラベルとボリューム名を作成する方法	447
	ファイルシステムの概要	447
	ファイルシステムのタイプ	448
	ディスクベースのファイルシステム	449
	ネットワークベースのファイルシステム	450
	仮想ファイルシステム	450
	ファイルシステム管理コマンド	453
	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	454
	汎用コマンドと専用コマンドのマニュアルページ	454
	デフォルトの Solaris ファイルシステム	455
	スワップ空間	456

- UFS ファイルシステム 457
 - UFS ファイルシステムの構成 457
 - UFS ログイン 458
 - UFS ファイルシステムの計画 459
- ファイルシステムのマウントとマウント解除 459
 - マウントされたファイルシステムテーブル 461
 - 仮想ファイルシステム 461
 - NFS 環境 463
 - 自動マウント (AutoFS) 463
 - キャッシュファイルシステム (CacheFS) 464
 - マウント方法の決定 465
 - ファイルシステムのタイプを調べる 465
 - ▼ ファイルシステムのタイプを調べる方法 466
- 35. ファイルシステムの作成 (手順) 469**
 - UFS ファイルシステムの作成 469
 - ファイルシステムパラメータ 470
 - ▼ UFS ファイルシステムを作成する方法 471
 - 一時ファイルシステム (TMPFS) の作成 473
 - ▼ TMPFS ファイルシステムを作成する方法 473
 - ループバックファイルシステム (LOFS) の作成 475
 - ▼ LOFS ファイルシステムを作成する方法 475
- 36. ファイルシステムのマウントとマウント解除 (手順) 477**
 - ファイルシステムのマウント 478
 - 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 479
 - 汎用マウントオプション 479
 - ▼ 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 482
 - ファイルシステムのマウント (/etc/vfstab) 482
 - /etc/vfstab ファイルのフィールドの説明 482

- ▼ /etc/vfstab ファイルにエントリを追加する方法 484
- ▼ 1つのファイルシステムをマウントする方法 (/etc/vfstab) 486
- ▼ /etc/vfstab ファイル内に列挙されているすべてのファイルシステムをマウントする方法 486
- ファイルシステムをマウントする (mount コマンド) 488
- ▼ UFS ファイルシステムをマウントする方法 488
- ▼ 大規模ファイルを持たない UFS ファイルシステムを再マウントする方法 489
- ▼ NFS ファイルシステムをマウントする方法 491
- ▼ IA: System V (S5FS) ファイルシステムをマウントする方法 492
- ▼ IA: ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法 493
- ファイルシステムのマウント解除 494
 - 前提条件 494
 - マウント解除したファイルシステムを検査する 495
- ▼ 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 495
- ▼ 1つのファイルシステムをマウント解除する方法 496
- ▼ すべてのファイルシステムをマウント解除する方法 (/etc/vfstab) 497
- 37. キャッシュファイルシステム (手順) 499**
 - CacheFS の機能 500
 - キャッシュファイルシステムの設定 501
 - キャッシュを作成する 502
 - ▼ キャッシュを作成する方法 502
 - キャッシュにマウントするファイルシステムの指定 503
 - ▼ mount を使用してキャッシュにマウントするファイルシステムを指定する方法 504
 - ▼ /etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法 506
 - ▼ AutoFS 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方法 507
 - キャッシュされたファイルシステムの管理 508
 - キャッシュの管理 509

- ▼ キャッシュ内のファイルシステムを変更する方法 510
- ▼ キャッシュされたファイルシステムに関する情報を表示する方法 511
- ▼ 必要に応じて整合性チェックを指定する方法 512
- ▼ キャッシュされたファイルシステムを削除する方法 512
- ▼ キャッシュされたファイルシステムの完全性をチェックする方法 514
- cachefspack によるキャッシュファイルシステムの管理 515
- ▼ キャッシュにファイルをバックする方法 516
- パッキングリスト 516
- ▼ パッキングリストを作成する方法 517
- ▼ パッキングリストに指定してファイルをキャッシュにバックする方法 517
- ▼ 正規表現として扱うファイルをパッキングリストに指定する方法 518
- ▼ 共有ディレクトリのファイルをバックする方法 519
- ファイルのパッキング解除 520
- ▼ 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 520
- バックされたファイルの情報の表示 521
- ▼ バックされたファイルの情報を表示する方法 521
- cachefspack コマンドのヘルプの表示 523
- cachefspack エラー 524
- CacheFS の統計情報 529
- CacheFS 統計情報を設定して表示するための前提条件 530
- CacheFS 統計情報の設定 531
- CacheFS ロギング 531
- ▼ ロギングプロセスを設定する方法 531
 - ログファイルの場所を調べる方法 532
 - ロギングプロセスを停止する方法 533
- キャッシュサイズの表示 533
- ▼ 作業セット (キャッシュ) のサイズを表示する方法 533

	統計情報の表示	535
	▼ キャッシュ統計情報を表示する方法	535
	キャッシュの構造と動作	536
	キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック	537
	必要に応じて行う整合性チェック	537
38.	追加スワップ空間の構成 (手順)	539
	スワップ空間について	539
	スワップ空間と仮想メモリー	540
	スワップ空間と TMPFS ファイルシステム	540
	スワップ空間の追加が必要かどうかを調べる方法	541
	スワップ関連のエラーメッセージ	541
	TMPFS 関連のエラーメッセージ	541
	スワップ空間の割り当て方法	542
	/etc/vfstab ファイル	543
	スワップ空間の計画	543
	スワップ資源の監視	544
	スワップ空間の追加	545
	スワップファイルの作成	546
	▼ スワップファイルを作成して使用可能にする方法	546
	スワップファイルを削除する	548
	▼ 余分なスワップ空間を削除する方法	548
39.	ファイルシステムの整合性チェック	551
	ファイルシステムの完全性	551
	ファイルシステムの状態はどのように記録されるか	552
	fsck でチェックして修復される内容	554
	非整合状態が発生する原因	554
	整合性がチェックされる UFS 構成要素	555

- fsck 要約メッセージ 561
- ブート時のファイルシステムチェック機能の変更 562
 - /etc/vfstab ファイル 563
- ▼ ブート中に実行されるファイルシステムチェック機能を変更する方法 564
- UFS ファイルシステムを対話式でチェックして修復する 565
 - ▼ ファイルシステムのチェックが必要かどうかを調べる方法 565
 - ▼ ファイルシステムを対話式でチェックする方法 566
 - UFS ファイルシステムの修復 567
 - ▼ ファイルシステムを修復する方法 568
- 不正なスーパーブロックの復元 568
 - ▼ 不正なスーパーブロックを復元する方法 568
 - fsck で修復できない UFS ファイルシステムを修正する方法 570
- fsck コマンドの構文とオプション 571
 - 汎用 fsck コマンドの構文、オプション、引数 571
- 40. UFS ファイルシステム (参照情報) 575**
 - 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 575
 - プラットフォームに依存するディレクトリ 584
 - UFS ファイルシステムのシリンダグループの構造 585
 - ブートブロック 585
 - スーパーブロック 585
 - i ノード 586
 - データブロック 588
 - 空きブロック 588
 - カスタムファイルシステムパラメータの決定 589
 - 論理ブロックサイズ 589
 - フラグメントサイズ 590
 - 最小空き容量 591
 - 回転の遅れ (ギャップ) 591

	最適化のタイプ	592
	ファイルの数	593
	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	594
	newfs コマンドの構文、オプション、引数	594
	汎用 mkfs コマンド	596
	UFS 直接入出力	597
	▼ 直接入出力を UFS ファイルシステムで有効にする方法	597
41.	データのバックアップと復元	599
42.	ファイルシステムのバックアップと復元 (概要)	601
	バックアップと復元についての参照先	601
	ファイルシステムのバックアップと復元とは	602
	ファイルシステムをバックアップしなければならない理由	603
	テープデバイスの選択	603
	バックアップを作成するファイルシステムの計画	604
	バックアップコマンドと復元コマンドの概要	607
	バックアップタイプの選択	608
	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	609
	バックアップスケジュールに影響する要素	609
	バックアップ頻度	609
	ダンプレベルを使用して増分バックアップを作成する	609
	バックアップスケジュールの例	611
	例 — 日単位累積、週単位累積バックアップ	611
	例 — 日単位累積、週単位増分バックアップ	613
	例 — 日単位増分、週単位累積バックアップ	614
	例 — サーバーのバックアップスケジュール	615
	バックアップスケジュールに関するその他の推奨事項	618
43.	ファイルとファイルシステムのバックアップ (手順)	621
	バックアップを実行するための準備	621

- ▼ ファイルシステム名を検索する方法 622
- ▼ 完全バックアップをとるために必要なテープの本数を決定する方法 622
- バックアップの実行 623
- ▼ テープにバックアップをとる方法 624
- 44. ファイルとファイルシステムの復元 (手順) 633
 - ファイルとファイルシステムを復元するための準備 633
 - ディスクデバイス名の決定 634
 - 必要なテープドライブのタイプの決定 634
 - テープドライブ名の決定 634
 - ファイルシステム全体の復元 635
 - 個々のファイルとディレクトリの復元 635
 - ファイルシステムの復元 636
 - ▼ 使用するテープを決定する方法 636
 - ▼ 対話式でファイルを復元する方法 637
 - ▼ 対話式でない方法で特定のファイルを復元する方法 640
 - ▼ リモートドライブを使ってファイルを復元する方法 642
 - ▼ ファイルシステム全体を復元する方法 643
 - ▼ ルート (/) と /usr を復元する方法 646
- 45. ufsdump コマンドと ufsrestore コマンド (参照情報) 651
 - ufsdump の機能 651
 - デバイス特性の判断 652
 - 媒体の終りの検出 652
 - ufsdump データのコピー 652
 - /etc/dumpdates ファイルの役割 652
 - バックアップデバイス (*dump-file*) 引数 653
 - バックアップを作成するファイルを指定する 655
 - 媒体の終りの検出 656
 - テープの性質を指定する 656

	ufsdump の制限	657
	ufsdump コマンドのオプションと引数	657
	ufsdump のデフォルトオプション	658
	ufsdump コマンドのオプション	658
	ufsdump とセキュリティに関する注意事項	660
	ufsrestore コマンドのオプションと引数	661
	ufsrestore コマンド構文	661
	ufsrestore のオプションと引数	661
	対話式の復元コマンド	664
46.	UFS ファイルとファイルシステムのコピー (手順)	667
	ファイルシステムをコピーするためのコマンド	668
	ファイルシステムをディスクにコピーする	670
	ファイルシステムのリテラルコピーを作成する	670
	▼ ディスクをクローン化する方法 (dd)	671
	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	674
	▼ 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)	674
	ファイルとファイルシステムをテープにコピーする	676
	tar を使用してファイルをテープにコピーする	678
	▼ ファイルをテープにコピーする方法 (tar)	678
	▼ テープ上のファイルのリストを表示する方法 (tar)	679
	▼ テープからファイルを取り出す方法 (tar)	680
	pax を使用してファイルをテープにコピーする	682
	▼ ファイルをテープにコピーする方法 (pax)	682
	▼ 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)	683
	▼ テープ上のファイルのリストを表示する方法 (cpio)	684
	▼ テープからすべてのファイルを取り出す方法 (cpio)	685
	▼ テープから特定のファイルを取り出す方法 (cpio)	686
	▼ ファイルをリモートテープドライブにコピーする方法 (tar と dd)	688

- ▼ ファイルをリモートテープドライブから取り出す方法 689
- 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 690
 - 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 691
- ▼ 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) 691
- ▼ 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) 692
- ▼ ファイルをフロッピーディスクから取り出す方法 (tar) 693
- ▼ 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 694
- ファイルを別のヘッダー形式でコピーする 695
- ▼ SunOS の旧リリース用のアーカイブを作成する方法 695
 - bar コマンドで作成したファイルを取り出す 696
- ▼ bar ファイルをフロッピーディスクから取り出す方法 696
- 47. テープドライブの管理 (手順) 699
 - 使用する媒体の選択 699
 - バックアップデバイス名 700
 - テープドライブのデフォルト密度を指定する 701
 - テープドライブに別の密度を指定する 702
 - テープドライブの状態を表示する 702
 - ▼ テープドライブの状態を表示する方法 702
 - 磁気テープカートリッジの取り扱い 704
 - ▼ 磁気テープカートリッジのたるみを直す方法 704
 - ▼ 磁気テープカートリッジを巻き戻す方法 704
 - ドライブの管理と媒体処理のガイドライン 705
 - 索引 707

はじめに

本書『Solaris のシステム管理 (第 1 巻)』は、Solaris システム管理に関する情報を提供する、3 巻構成のマニュアルの第 1 巻です。本書は、SPARC™ および IA 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。

このマニュアルでは、システム管理者である読者が SunOS™ 5.8 オペレーティングシステムをすでにインストールしていて、ネットワークソフトウェアの設定を終了していることを想定しています。SunOS 5.8 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 5.8 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。

システム管理に関する Solaris 8 の新機能については、各章の新機能に関する節を参照してください。

注 - Solaris オペレーティング環境は、SPARC と IA 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 ビットアドレス空間でも 32 ビットアドレス空間でも動作します。このマニュアルの情報は、章、節、項、注、箇条書き、表、例、コード例などで特に記述する場合を除き、両方のプラットフォームと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 8 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 のシステム管理』 全 3 巻の内容

Solaris システム管理について説明している 3 冊のマニュアルには、主に次に示す内容が記載されています。

『Solaris のシステム管理 (第 1 巻)』 (本書)

- 「ユーザーアカウントとグループの管理」
- 「サーバーとクライアントサポートの管理」
- 「システムのシャットダウンとブート」
- 「取り外し可能な媒体の管理」
- 「ソフトウェアの管理」
- 「デバイスの管理」
- 「ディスクの管理」
- 「ファイルシステムの管理」
- 「データのバックアップと復元」

『Solaris のシステム管理 (第 2 巻)』

- 「印刷サービスの管理」
- 「リモートシステムの利用」
- 「端末とモデムの管理」
- 「システムセキュリティの管理」
- 「システム資源の管理」

- 「システム性能の管理」
- 「Solaris ソフトウェアで発生する問題の解決」

『Solaris のシステム管理 (第 3 巻)』

- 「ネットワークサービス」
- 「IP アドレスの管理」
- 「モデム関連ネットワークサービス」
- 「リモートファイルシステムの管理」
- 「メールサービスの設定と管理」
- 「ネットワークサービスの監視」

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

<http://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、検索をおこなうこともできます。

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す。

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。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
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。	system% su password:
<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、rm <i>filename</i> 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。	sun% grep `^#define \ XV_VERSION_STRING`

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

■ C シェルプロンプト

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

■ Bourne シェルおよび Korn シェルのプロンプト

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

- スーパーユーザーのプロンプト

```
system# 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、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。
- 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。
- このマニュアル中の例は、SunOS 5.8 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、パスの一番最後に追加してください。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。

ユーザーアカウントとグループの管理

ここでは、ユーザーとグループの管理について説明します。次の章で構成されています。

第 2 章

ネットワーク環境でユーザーアカウントとグループを設定する方法について概要を説明します。

第 3 章

ユーザーマネージャとグループマネージャを使用して、ユーザーアカウントとグループを設定する手順について説明します。

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計画、ネットワーク環境でユーザーアカウントとグループを設定する方法について概要を説明します。また、ユーザーアカウントとグループ情報を格納するファイル、およびユーザーの作業環境のカスタマイズについて説明します。

この章の内容は次のとおりです。

- 36ページの「ユーザーアカウントとグループとは」
- 37ページの「ユーザーアカウント管理のガイドライン」
- 45ページの「グループを管理するガイドライン」
- 46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」
- 51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」
- 58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とグループを管理する手順については、第 3 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 Solaris リリースでは、ユーザーやグループの役割 (role) によってアクセス制御を設定する RBAC (Role-based Access Control) によって、スーパーユーザーの特権の一部をまとめてユーザーアカウントに割り当てることができます。この機能

によって、特定の問題だけを解決しなければならないユーザーに、スーパーユーザーのすべての特権を与える必要がなくなります。

詳細は、『Solaris のシステム管理 (第 2 巻)』の「役割によるアクセス制御」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、サイトにおいて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。通常のユーザーアカウントには、ユーザーがシステムにログインして、システムを (スーパーユーザーのパスワードを知らなくても) 使用するのに必要な情報が含まれます。ユーザーアカウント情報は、主に次の 4 つ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る)。
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使用して) 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最高機密のファイルを入れるディレクトリを作成できます。極秘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、最高機密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、システム管理の役割に対応した、特別な種類のユーザーアカウントもあります。こ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。詳細は、『Solaris のシステム管理 (第 2 巻)』の「役割によるアクセス制御」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、NIS または NIS+ などのネームサービスを使用できます。ネームサービスにより、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、1ヶ所で一貫した管理を行えます。ユーザーアカウントにネーム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 /etc 内のファイル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。

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。ユーザー名は、次の条件を満たしていなければなりません。

- 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文字から8文字の英数字を使用する (最初の文字は英字でなければならない、少なくとも1文字は小文字でなければならない)。
- 下線や空白文字は入れない。

ユーザー名の標準的な作り方を決めておくと便利です。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、固有の名前になるまで、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、および1、2、3などの数字を使用できます。

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 ID (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。また、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に必要です。表 2-1 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 2-1 予約済みの UID 番号

ユーザー ID 番号	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	nobody	アクセス権のないユーザー
60002	noaccess	Solaris 2.0 およびその互換バージョン と SVR4 リリースとの互換性のため に提供

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、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、固有の従業員番号に、管理者が 1000 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も適用するとは限りません。必要な場合には、固有の UID が使い果たされたとき、1 つの NIS+ ドメイン内で重複する UID を使用することができます。

大きな数値のユーザー ID とグループ ID の使用

以前の Solaris ソフトウェアリリースでは、32 ビットのデータ型を使用してユーザー ID (UID) とグループ (GID) を格納していましたが、UID と GID に使用できる最大値は 60000 に制限されていました。Solaris 2.5.1 リリースおよびその互換バージョンからは、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。

60000 を超える 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。

Solaris の以前のリリースとの相互運用性については、表 2-2 を参照してください。

表 2-2 60000 を超える UID と GID の相互運用性に関する問題

分類	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バージョン	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。これによって、大きな UID と GID を使用している環境において SunOS 4.0 およびその互換バージョンのシステムを使用すると、セキュリティの問題が発生する可能性がある。SunOS 4.0 およびその互換バージョンのシステムにはパッチが必要である。
ネームサービスの相互運用性	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 を超える UID を持つユーザーは、Solaris 2.5 およびその互換バージョン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 を超える UID を持つユーザーは、Solaris 2.5 およびその互換バージョンと 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。
UID と GID の表示	OpenWindows のファイルマネージャ	OpenWindows™ のファイルマネージャを拡張ファイルリスト表示オプションで使用する場合、大きな UID と GID は正しく表示されない。

表 2-3 大きな UID と GID の制限の要約

UID と 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バージョンと 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"> ■ NFS バージョン 2 ソフトウェアが動作している SunOS 4.0 およびその互換バージョンのシステムでは、この分類の UID は 16 ビットに切り捨てられる。その結果、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 cpio コマンド (デフォルトのアーカイブフォーマットを使用する) 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。 ■ IA 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。 ■ IA 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 cpio コマンド (-H odc フォーマットを使用する) 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文字から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。ユーザーアカウントを作成するときにユーザーのパスワードを設定します。ユーザーは、システムに初めてログインするときにそのパスワードを変更できます。

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カ月に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。

次のようなパスワードを選択してください。

- 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 swotrbr)
- 文字を数字や記号に代えた単語 (たとえば、snoopy を sn00py にする)

次のものは、パスワードに不適當です。

- 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
- 12月の "Santa" のように季節に関連した言葉
- 辞書にある単語

パスワード有効期限

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。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けを設定することができます。

ホームディレクトリ

ホームディレクトリは、ユーザーが専有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と作業によって異なります。一般的には、各ユーザーのホームディレクトリに少なくとも 15M バイトのディスク空間を割り当ててください。

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容易に行うことができるようにするために、各 /export/home ディレクトリに対して別々のファイルシステム (たとえば、/export/home1、/export/home2 など) を使用してください。

ホームディレクトリが配置される位置に関係なく、ユーザーは通常 /home/username という名前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の自動マウントについての詳細は、『Solaris のシステム管理 (第 3 巻)』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他に、ユーザーには仕事をするために必要なツールと資源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) 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詳細については、58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薦めするのは、「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の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) 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、集中管理されません。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。詳細は、『Solaris のシステム管理 (第 2 巻)』の「役割によるアクセス制御」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構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なければなりません。
- 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、最高 16 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、自分がメンバーとなっている他のグループに (`newgrp` コマンドを使用して) 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、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。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通して管理することができます。グループ管理を単純化するために、グループメンバーを集中管理できる NIS+ のようなネームサービスを使用してください。

ユーザーアカウントとグループを管理するツール

表 2-4 に、ユーザーとグループを管理する推奨ツールを示します。

表 2-4 ユーザーとグループを管理するための推奨ツール

管理するユーザーとグループの場所	推奨ツール	必要なもの	参照先
ネットワークに接続されたネームサービス (NIS、NIS+) 環境におけるリモートまたはローカル、またはその両方のシステム	AdminSuite2.3™ のユーザーマネージャとグループマネージャ (グラフィカルユーザーインターフェース)	CDE などの X window 環境が動作しているグラフィックモニター	『Solstice AdminSuite 2.3 管理者ガイド』
ローカルシステム	Admintool (グラフィカルユーザーインターフェース)	CDE などの X window が動作しているグラフィックモニター	第 3 章

Solaris の `useradd` コマンドと `groupadd` コマンドを使用しても、ローカルシステムのユーザーとグループを設定できます。ただし、これらのコマンドではネームサービスのマップやテーブルを変更できません。表 2-5 で、Solstice AdminSuite2.3 または Admintool を使用しない場合に、ユーザーアカウントとグループを管理する Solaris のコマンドについて説明します。

表 2-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法

作業	使用するネームサービス	使用するコマンド
ユーザーアカウントの追加	NIS+	<code>nistbladm</code> <code>nisclient</code>
	NIS	<code>useradd</code> <code>make</code>
	なし	<code>useradd</code>

表 2-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法 続く

作業	使用するネームサービス	使用するコマンド
ユーザーアカウントの変更	NIS+	nistbladm
	NIS	usermod make
	なし	usermod
ユーザーアカウントの削除	NIS+	nistbladm nisclient
	NIS	userdel make
	なし	userdel
ユーザーアカウントのデフォルトの設定	NIS+	なし
	NIS	useradd -D make
	なし	useradd -D
ユーザーアカウントを無効にする	NIS+	nistbladm
	NIS	passwd -r nis -l make
	なし	passwd -f files -l
ユーザーのパスワードの変更	NIS+	passwd -r nisplus
	NIS	passwd -r nis
	なし	passwd -r files

表 2-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法 続く

作業	使用するネームサービス	使用するコマンド
ユーザーアカウントのソート	NIS+	niscat sort
	NIS	ypcat sort
	なし	awk sort
ユーザーアカウントの検索	NIS+	nismatch
	NIS	ypmatch
	なし	grep
グループの追加	NIS+	nistbladm
	NIS	groupadd make
	なし	groupadd
グループ内のユーザーの変更	NIS+	nistbladm
	NIS	groupmod make
	なし	groupmod
グループの削除	NIS+	nistbladm
	NIS	groupdel make
	なし	groupdel

Admintool の機能

Admintool はローカルシステムでユーザーアカウントを設定できるグラフィカルユーザーインターフェースです。

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、Admintool 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。Admintool 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、Admintool 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。

Admintool を使用してユーザー名を変えた場合でも、ホームディレクトリの所有権は変更されます。(ユーザーのホームディレクトリが存在する場合)。

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シップがあります。Admintool の「変更 (Modify)」オプションで、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。あるいは、「グループ」ウィンドウ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。

- コメント
- ログインシェル
- パスワード
- ホームディレクトリ

ユーザーアカウントの削除

Admintool でユーザーアカウントを削除すると、passwd、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にあるファイルを削除できます。

カスタマイズしたユーザー初期設定ファイルの追加

ユーザーマネージャでユーザー初期設定ファイルをカスタマイズして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。

/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、それらを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

Admintool を使って、次のようにパスワードを管理できます。(1) ユーザーアカウントに通常のパスワードを指定する(2) 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(3) ユーザーアカウントを無効にするかロックする(4) 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

一時的にまたは永久に、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、Admintool 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。また、「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を無効にする期間を設定することができます。

また、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 内のファイルのどちらかに格納できます。NIS+ ネームサービスでは、情報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、ファイル、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 を使用する時、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。

パスワード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
```

具体的な例は次のようになります。

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

表 2-6 に、passwd ファイルの各フィールドについて説明します。

表 2-6 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 から 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。ユーザー名には下線や空白文字は使用できない。
<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 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)。各 GID は 0 から 60002 までの範囲の整数でなければならない (60001 と 60002 はそれぞれ nobody と noaccess 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ので、GCOS 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 2-13 のシェル機能の説明を参照のこと。

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タスク (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:
```

(続く)

```

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
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 2-7 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	カレンダーの更新など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
	71	ラインプリンタのデーモン
uucp	5	uucp デーモン
nuucp	6	uucp デーモン
listen	37	ネットワーク監視デーモン
nobody	60001	未承認の root ユーザーから要求を受信したときに NFS サーバーが割り当てる匿名ユーザー用のアカウント。nobody ユーザーアカウントは、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ソフトウェアプロセスに割り当てられる。

表 2-7 デフォルトの passwd ファイルのエントリ 続く

ユーザー名	ユーザー ID	説明
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しないで、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アカウント。
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
```

具体的な例は次のようになります。

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

表 2-8 に shadow ファイルの各フィールドについて説明します。

表 2-8 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。

表 2-8 shadow ファイルのフィールド 続く

フィールド名	説明
<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
```

具体的な例は次のようになります。

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

表 2-9 に group ファイルの各フィールドについて説明します。

表 2-9 group ファイルの各フィールド

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 など指定する。グループ名に許される最大文字数は 9 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。グループパスワードフィールドは初期バージョンの UNIX のなごりで、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 2-9 group ファイルの各フィールド 続く

フィールド名	説明
<i>gid</i>	グループ 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>	カ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シップを表す。各ユーザーは最高 16 個まで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タスク (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。これらの多くのエントリは、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,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:root
nobody::60001:
noaccess::60002:
nogroup::65534:

```


表 2-10 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	
bin	2	システムバイナリの実行に関連する管理グループ
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 機能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
	8	ラインプリンタのグループ
nuucp	9	uucp 機能に関連するグループ
staff	10	一般的な管理グループ
daemon	12	デーモングループ
sysadmin	14	Admintool と AdminSuite に関連する管理グループ
nobody	60001	未承認の root ユーザーから要求を受信したときに NFS サーバーが割り当てる匿名グループ
noaccess	60002	
nogroup	65534	

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でき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が、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。表 2-11 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 2-11 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	<code>\$HOME/.profile</code>	ログイン時のユーザー環境の定義
C	<code>\$HOME/.cshrc</code>	ログインシェルのあとに起動されるすべての C シェルに対するユーザー環境の定義
	<code>\$HOME/.login</code>	ログイン時のユーザー環境の定義
Korn	<code>\$HOME/.profile</code>	ログイン時のユーザー環境の定義
	<code>\$HOME/\$ENV</code>	Korn シェルの ENV 環境変数によって指定されるファイルにおける、ログイン時のユーザー環境の定義

Solaris 環境には、表 2-12 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 2-12 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	<code>/etc/skel/local.login</code>
	<code>/etc/skel/local.cshrc</code>
Bourne または Korn	<code>/etc/skel/local.profile</code>

表 2-12 デフォルトのユーザー初期設定ファイル 続く

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したり、あるいは異なるタイプのユーザーに作業環境を提供したりすることができます。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、75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。

Admintool 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 (あるいは、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初めに次のような行を入れてください。

```
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初め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。次に例を示します。

- 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/machine-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/machine-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

表 2-13 に、各シェルの基本的機能を示します。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 2-13 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	○	×	×
Bourne シェルと互換性がある構文	-	×	○
ジョブ制御	○	○	○
履歴リスト	×	○	○
コマンド行編集	×	○	○
別名 (エイリアス)	×	○	○
ログインディレクトリの 1 文字省略形	×	○	○
ファイルの上書き保護 (noclobber)	×	○	○
CTRL-D 無視 (ignoreeof)	×	○	○
拡張 cd	×	○	○
.profile とは別の初期設定ファイル	×	○	○
ログアウトファイル	○	○	×

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によって定義されます。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 両シェルでは、一般的に、大文字を使って `setenv` コマンドでシェル変数と環境変数を設定します。また、`export` コマンドで環境変数の設定を終了しなければなりません。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。表 2-14 は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 2-14 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	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>

表 2-15 に、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を説明します。各シェルで使用される変数についての詳細は、`sh(1)`、`ksh(1)`、`csh(1)` の各マニュアルページを参照してください。

表 2-15 シェル変数と環境変数の説明

変数	説明
ARCH	ユーザーのシステムアーキテクチャ (たとえば sun4、i386) を設定する。この変数は ARCH = 'uname -p' (Bourne または Korn シェル) または setenv ARCH 'uname -p' (C シェル) で設定する。この変数に影響されるシェルの動作はない。この変数は単にシェルスクリプト内での分岐に利用する。
CALENDAR	Calendar 実行ファイルにパスを設定する。
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。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、見つかったら、ディレクトリの変更が行われる。CDPATH で対象の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 に 2 つのディレクトリがある場合に、/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。
DESKSET	DeskSet™ 実行ファイルへのパスを設定する。
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。したがって、この変数を参照すればユーザー名を入手できる (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。
MANSECTS	アクセスできるマニュアルページの階層を設定する。
OPENWINHOME	OpenWindows サブシステムへのパスを設定する。

表 2-15 シェル変数と環境変数の説明 続く

変数	説明
PATH (C シェルでは path)	<p>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。</p> <p>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。</p> <p>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、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。</p>
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。
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	<p>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 変数を使う。</p> <p>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は、端末は dumb として定義される。</p>

表 2-15 シェル変数と環境変数の説明 続く

変数	説明
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 で再度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。ディレクトリのどれかで見つければ、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を重複しないように確認してください。
- 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初めに指定されます)。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。特に必要な場合はこれとは別に、次の LC 変数、LC_COLLATE、LC_CTYPE、LC_MESSAGES、LC_NUMERIC、LC_MONETARY、LC_TIME によってその他の設定を行えます。

表 2-16 は、LANG と LC 環境変数の値の一部を示します。

表 2-16 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
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 3 桁の 8 進値で設定します。最初の桁でそのユーザーのアクセス権を設定し、第 2 桁でグループのアクセス権を設定し、第 3 桁で「その他」(ワー

ルドとも呼ばれます)のアクセス権を設定します。最初の桁がゼロの場合、こ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--)に変更したいとすれば、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、表 2-17 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 2-17 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まず使用する、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。

コード例 — .profile ファイルの例

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

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。

コード例 — .cshrc ファイルの例

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

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 (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. 指定された環境変数をエクスポートする。

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。

ユーザーアカウントとグループの設定と管理 (手順)

この章では、ユーザーアカウントとグループを設定し、管理する手順について説明します。

この章の内容は次のとおりです。

- 75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」
- 76ページの「Admintool を起動する方法」
- 77ページの「グループを追加する方法」
- 79ページの「新しいユーザーアカウントを追加する方法」
- 80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」
- 82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」
- 85ページの「グループを変更する方法」
- 86ページの「グループを削除する方法」
- 86ページの「ユーザーアカウントを変更する方法」
- 88ページの「ユーザーアカウントを無効にする方法」
- 90ページの「ユーザーのパスワードを変更する方法」
- 91ページの「ユーザーアカウントのパスワード有効期限を変更する方法」
- 93ページの「ユーザーアカウントを削除する方法」
- 96ページの「Solaris ユーザー登録をやり直す方法」
- 97ページの「ユーザー登録を無効にする方法」

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2章を参照してください。

スーパーユーザー (root) になる

ユーザーを追加するなど、ほとんどの管理作業では、まず root (UID=0) でログインする必要があります。root アカウントは、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できるので、スーパーユーザーアカウントとも言います。

むやみにシステムを変更することを避けるために、スーパーユーザーアカウントは、管理作業を実行するためだけに使用してください。

root としてシステムにログインすることも、su (1M) コマンドを使用してスーパーユーザーアカウントに変更することも可能です。

役割によるアクセス制御を使用している場合、管理作業を実行する役割 (root またはその他の役割) になる必要があります。su コマンドを使用することによってその役割になることができます。特定の役割で直接ログインすることはできません。詳細は、『Solaris のシステム管理 (第2巻)』の「役割によるアクセス制御」を参照してください。

▼ スーパーユーザー (root) になる方法

次のいずれかの方法で、スーパーユーザーになります。どちらの方法でも、root のパスワードを知っておく必要があります。

- su コマンドを使用して、スーパーユーザーに変更する方法

```
% su
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。

- 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 #
```

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でスーパーユーザーとしてログインするには、`/etc/default/login` ファイルを変更しなければなりません。このファイルの変更方法については、『Solaris のシステム管理 (第 2 巻)』の「システムのセキュリティ」を参照してください。

ユーザーアカウントの設定

表 3-1 作業マップ: ユーザーアカウントの設定

作業	説明	手順の説明
1. ユーザー初期設定ファイルのカスタマイズ	(省略可能) ユーザー初期設定ファイル (<code>.cshrc</code> 、 <code>.profile</code> 、 <code>.login</code>) を設定する。これによって新規ユーザーに一貫した環境を提供できる。	75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」
2. グループの追加	(省略可能) ユーザーの管理を容易にするため、グループマネージャの「編集(Edit)」メニューから「追加(Add)」を選択してグループを追加する。通常は 1 回だけ行う作業。	77ページの「グループを追加する方法」
3. ユーザーアカウントの追加	新しいユーザーアカウントを追加する <code>Admintool</code> の「ユーザー」メインウィンドウを使用して、ユーザーアカウントを追加する。	79ページの「新しいユーザーアカウントを追加する方法」
4. 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。	80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」
5. 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 <code>mount</code> コマンドによって手作業でマウントする。	82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」

ユーザー情報データシート

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以下のような形式で記録しておくくと便利です。

役割によるアクセス制御を使用している場合、ユーザーアカウントの役割、プロファイル、承認も記録しておく必要があります。詳細は、『Solaris のシステム管理 (第 2 巻)』の「役割によるアクセス制御」を参照してください。

項目	記入欄
ユーザー名	_____
ユーザー ID	_____
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ります。
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-directory/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前

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前

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 (プロファイルシェルと呼ぶ) を起動しなければなりません。シェルの種類に合わせて、`pfsh` (Bourne シェル)、`pfersh` (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の詳細は、58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-directory/skel/user-type/.*
```

6. `ls -la` コマンドで、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。

例 — ユーザー初期設定ファイルのカスタマイズ

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。

```
# mkdir /export/skel/enduser
# cp /etc/skel/local.cshrc /export/skel/enduser/.cshrc
(.cshrc の編集 -- 69ページの「コード例 — .cshrc ファイルの例」を参照)
# chmod 744 /export/skel/enduser/.*
```

▼ Admintool を起動する方法

1. **Admintool** を使用するために必要な次の条件がそろっていることを確認します。
 - ビットマップディスプレイモニター。Admintool ソフトウェアは、Sun ワークステーションの標準ディスプレイモニターなど、ビットマップ画面のコンソールを使用するシステムでだけ使用できます。

- CDE などの X Window 環境が動作していること。
 - sysadmin グループのメンバーであること (グループ 14)。
- コンソールとして ASCII 端末を使用するシステムで管理作業を行う場合、Admintool ではなく Solaris のコマンドを使用してください。詳細は、`useradd(1M)` のマニュアルページを参照してください。

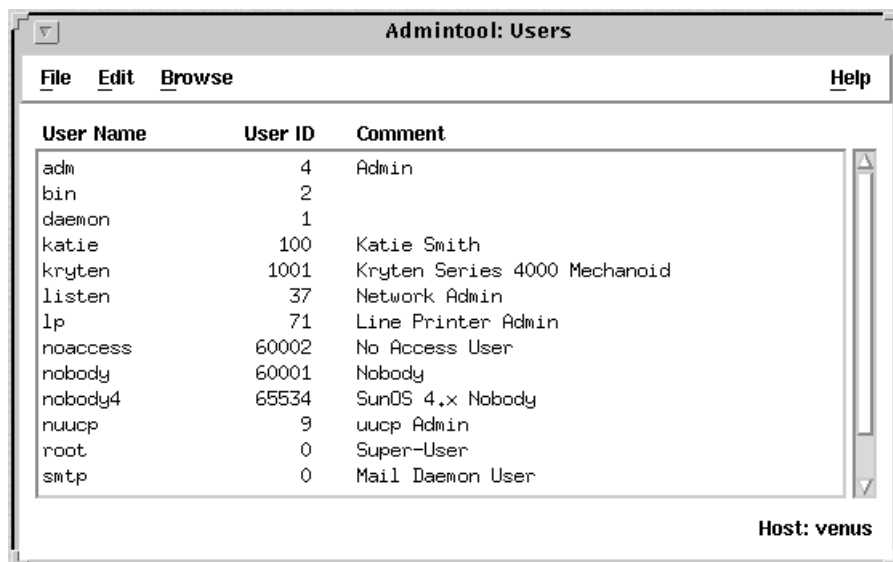
2. Admintool を起動します。

```
$ admintool &
```

「Admintool:ユーザー (Users)」メインウィンドウが表示されます。

例 — Admintool を起動する

「Admintool:ユーザー (Users)」メインウィンドウを使用して、ユーザーアカウント情報を管理できます。



▼ グループを追加する方法

1. Admintool を起動します。

詳細は 76ページの「Admintool を起動する方法」を参照してください。

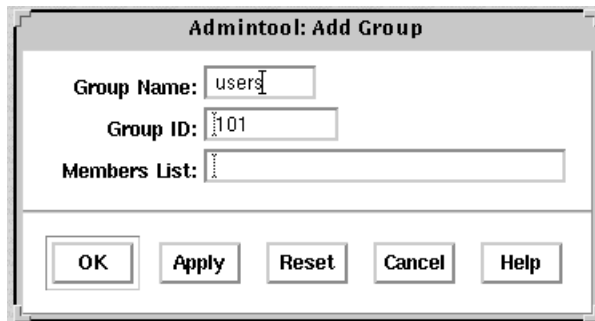
2. 「ブラウズ (Browse)」メニューから「グループ (Group)」を選択します。

「Admintool:グループ (Groups)」ウィンドウが表示されます。

3. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。
「追加 (Add)」ウィンドウにはいくつかのフィールドがあります。各フィールドについての説明を表示するには、「ヘルプ (Help)」ボタンをクリックしてください。
4. 新しいグループの名前を「グループ名 (**Group Name**)」テキストボックスに入力します。
5. 新しいグループのグループ ID を「グループ ID (**Group ID**)」テキストボックスに入力します。
グループ ID は固有でなければなりません。
6. (省略可能) 「メンバーリスト (**Members List**)」にユーザー名を入力します。
ユーザーのリストをグループに追加します。ユーザー名はカンマで区切ります。
7. 「了解 (**OK**)」をクリックします。
「Admintool:グループ (Groups)」ウィンドウに表示されるグループのリストに、新しいグループが追加されます。

例 — グループを追加する

次の例は、グループ名が users、グループ ID が 101 のグループを追加する場合です。



▼ 新しいユーザーアカウントを追加する方法

1. (省略可能) 74ページの「ユーザー情報データシート」のユーザー情報データシートに記入します。
2. **Admintool** を起動します。
詳細は 76ページの「Admintool を起動する方法」を参照してください。
3. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。
「追加 (Add)」ウィンドウが表示されます。
4. 「追加 (**Add**)」ウィンドウの各項目を入力します。
各フィールドについての説明を表示するには、「ヘルプ (Help)」ボタンをクリックしてください。
5. 「了解 (**OK**)」をクリックします。
「Admintool: ユーザー (Users)」のメインウィンドウに表示されるユーザーアカウントのリストに、新しいユーザーアカウントが追加されます。

次に進む手順

ユーザーのホームディレクトリを作成した場合、ユーザーのシステムがリモートでマウントできるように、そのディレクトリを共有しなければなりません。詳細は、80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。

ディスク容量に制限がある場合、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、そのユーザーのためのディスク割り当てを設定できます。ディスク割り当てを設定する方法については、『Solaris のシステム管理 (第 2 巻)』の「ディスク割り当ての管理」を参照してください。

例 — 新しいユーザーアカウントを追加する

次の例は、ユーザー `kryten` をシステムに追加します。

Admintool: Add User

USER IDENTITY

User Name:

User ID:

Primary Group:

Secondary Groups:

Comment:

Login Shell:

ACCOUNT SECURITY

Password:

Min Change: days

Max Change: days

Max Inactive: days

Expiration Date:

(dd/mm/yy)

Warning: days

HOME DIRECTORY

Create Home Dir:

Path:

▼ ユーザーのホームディレクトリを共有する方法

1. 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ります。
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一覧表示します。

```
# 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、次の手順を決定します。

ユーザーのホームディレクトリを含む ファイルシステムの状態	次の手順
すでに共有されている	次の確認手順へ進む
共有されていない	81ページの手順 6

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
```

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。習慣上、このファイルシステムは `/export/home` になる。

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。詳細は、82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の共有

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

( share -F nfs /export/home という行が追加される)

# 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

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。詳細は、80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。習慣上、 <i>/export/home</i> にユーザーのホームディレクトリが含まれるが、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
<i>/export/home/user-name</i>	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

/etc/vfstab ファイルにエントリを追加する方法については、第 36 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. *mount* コマンドを使って、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てください。

例 — ユーザーのホームディレクトリのマウント

```
# vi /etc/vfstab
(venus:/export/home/ripley - /export/home/ripley
nfs - yes rw を追加)
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on Fri ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on Fri ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
```

(続く)

```

/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home/ripley on venus:/export/home/ripley /read/write/remote on ...
    
```

ユーザーアカウントの管理

表 3-2 作業マップ：ユーザーアカウントの管理

作業	説明	手順の説明
グループの変更	「グループ」ウィンドウの「編集 (Edit)」メニューから「変更 (Modify)」を選択して、グループ名またはグループ内のユーザーを変更する。	85ページの「グループを変更する方法」
グループの削除	「グループ」ウィンドウの「編集 (Edit)」メニューから「削除 (Delete)」を選択して、グループを削除する。	86ページの「グループを削除する方法」
ユーザーアカウントの変更	<p>ユーザーアカウントを無効にする</p> <p>一時的にユーザーアカウントを無効にしたい場合、「変更 (Modify)」ウィンドウの「パスワード (Password)」メニューでそのユーザーアカウントをロックする。</p> <p>ユーザーのパスワードを変更する</p> <p>ユーザーのパスワードを変更したい場合、「変更 (Modify)」ウィンドウの「パスワード (Password)」メニューを使用する。</p>	<p>88ページの「ユーザーアカウントを無効にする方法」</p> <p>90ページの「ユーザーのパスワードを変更する方法」</p>

表 3-2 作業マップ：ユーザーアカウントの管理 続く

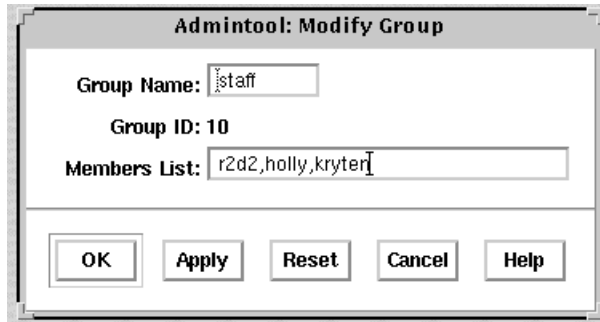
作業	説明	手順の説明
	パスワード有効期限を変更する 定期的にユーザーのパスワードを変更させたい場合、「変更 (Modify)」ウィンドウでユーザーのパスワード有効期限フィールドを変更する（「アカウントのセキュリティ設定(Account Security)」カテゴリ）。	91ページの「ユーザーアカウントのパスワード有効期限を変更する方法」
ユーザーアカウントの削除	「ユーザー」ウィンドウの「編集(Edit)」メニューから「削除 (Delete)」を選択して、ユーザーアカウントを削除する。	93ページの「ユーザーアカウントを削除する方法」

▼ グループを変更する方法

1. **Admintool** を起動して、「ブラウザ (**Browse**)」メニューからグループを選択します。
 詳細は 76ページの「Admintool を起動する方法」を参照してください。
2. 「**Admintool: グループ (Groups)**」ウインドウから変更するグループエントリを選択します。
3. 「**編集 (Edit)**」メニューから「**変更 (Modify)**」を選択します。
 指定したグループエントリを含む「**変更 (Modify)**」ウィンドウが表示されます。
4. グループ名またはグループ内のユーザーを変更します。
 ユーザー名はカンマで区切らなければなりません。各フィールドについての説明を表示するには、「**ヘルプ (Help)**」ボタンをクリックしてください。
5. 「**了解 (OK)**」をクリックします。
 メインウインドウに表示されるグループ情報が更新されます。

例 — グループを変更する

次の例は、ユーザー r2d2、holly、および kryten を、staff グループに追加します。



▼ グループを削除する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「グループ (**Groups**)」を選択します。
詳細は 76ページの「Admintool を起動する方法」を参照してください。
2. 「グループ」ウインドウから削除したいグループエントリを指定します。
3. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
ウインドウが表示され、削除してよいかどうかをたずねられます。
4. 「了解 (**OK**)」をクリックします。
「グループ」ウインドウからそのグループエントリが削除されます。

▼ ユーザーアカウントを変更する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細は 76ページの「Admintool を起動する方法」を参照してください。
2. 「ユーザー (**Users**)」ウインドウから変更するユーザーアカウントエントリを選択します。
3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したユーザーアカウントエントリを含む「変更 (**Modify**)」ウインドウが表示されます。

4. ユーザーアカウントを変更します。

各フィールドについての説明を表示するには、「ヘルプ (Help)」ボタンをクリックしてください。パスワードやパスワード有効期限の変更など、「アカウントのセキュリティ設定 (Account Security)」フィールドの任意の項目を変更することができます。詳しい手順については、以下の項を参照してください。

- 88ページの「ユーザーアカウントを無効にする方法」
- 90ページの「ユーザーのパスワードを変更する方法」
- 91ページの「ユーザーアカウントのパスワード有効期限を変更する方法」

5. 「了解 (OK)」をクリックします。

6. 「ユーザー (Users)」ウィンドウで、変更したユーザーアカウントのエントリをダブルクリックして、変更が行われていることを確認してください。確認後、そのまま「取消し (Cancel)」をクリックしてウィンドウを閉じてください。

例 — ユーザーアカウントを変更する

次の例は、lp に rimmer ユーザーアカウントを二次グループとして追加した場合です。

Admintool: Modify User

USER IDENTITY

User Name:

User ID: 1002

Primary Group:

Secondary Groups:

Comment:

Login Shell: /bin/csh

ACCOUNT SECURITY

Password:

Min Change: days

Max Change: days

Max Inactive: days

Expiration Date: (dd/mm/yy)

Warning: days

HOME DIRECTORY

Path:

▼ ユーザーアカウントを無効にする方法

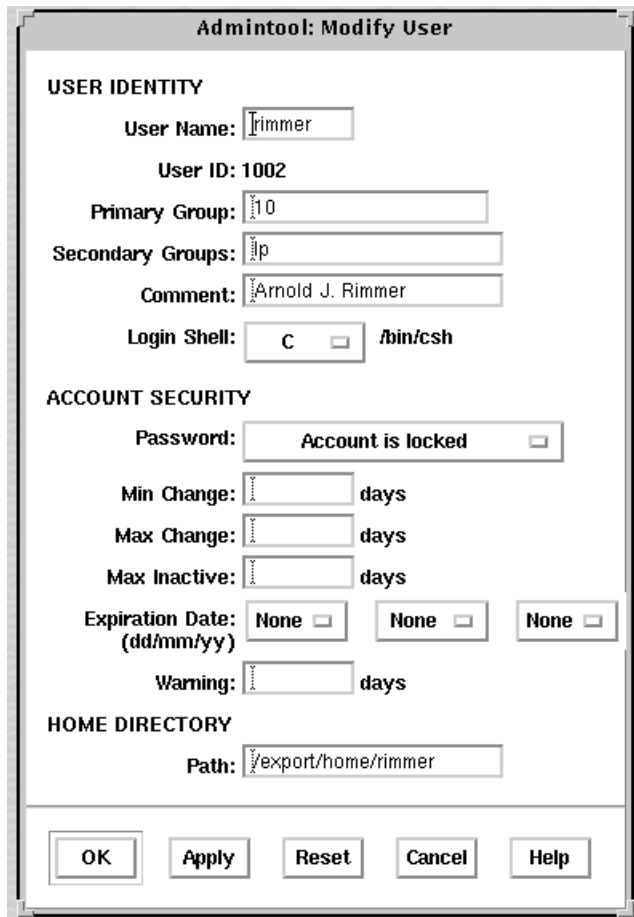
注 - パスワードの状態を「通常のパスワード設定 (Normal Password)」または「最初にログインするまで未設定 (Cleared until first login)」に変更することによって、ユーザーアカウントを有効にすることができます。

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細は 76 ページの「Admintool を起動する方法」を参照してください。
2. 無効にするユーザーアカウントエントリを選択します。

3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したユーザーアカウントエントリを含む「変更 (**Modify**)」ウィンドウが表示されます。
4. 「パスワード (**Password**)」メニューから「アカウントを凍結 (**Account is Locked**)」を選択します。
これによってパスワードの状態が凍結され、ユーザーアカウントが無効になります。
5. 「了解 (**OK**)」をクリックします。
6. 無効にしたユーザーアカウントでログインして、そのユーザーアカウントが無効になっていることを確認してください。

例 — ユーザーアカウントを無効にする

次の例は、`rimmer` ユーザーアカウントを無効にしています。



▼ ユーザーのパスワードを変更する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細は 76ページの「Admintool を起動する方法」を参照してください。
2. パスワード変更が必要なユーザーアカウントエントリを選択します。
3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したユーザーアカウントエントリを含む「変更 (Modify)」ウィンドウが表示されます。

4. 「パスワード (Password)」メニューから「通常のパスワード設定 (Normal password)」を選択します。
5. 「了解 (OK)」をクリックします。

例 — パスワードの変更

これは、ユーザーのパスワードを変更するために使用されるポップアップウィンドウです。このウィンドウは、「ユーザーを追加 (Add User)」ウィンドウまたは「ユーザーを変更 (Modify User)」ウィンドウから利用できます。



▼ ユーザーアカウントのパスワード有効期限を変更する方法

1. **Admintool** を起動します。「ブラウズ (Browse)」メニューから「ユーザー (Users)」を選択します。
詳細は 76 ページの「Admintool を起動する方法」を参照してください。
2. パスワード有効期限の変更が必要なユーザーアカウントエントリを選択します。
3. 「編集 (Edit)」メニューから「変更 (Modify)」を選択します。
指定したユーザーアカウントエントリを含む「変更 (Modify)」ウィンドウが表示されます。
4. パスワードの有効期限に関係のある以下のフィールドを変更します。
 - 最短有効日数 (Min Change)

- 最長有効日数 (Max Change)
- 最長非使用日数 (Max Inactive)
- 有効期限 (Expiration Date)
- 警告 (Warning)

「アカウントのセキュリティ設定 (Account Security)」カテゴリの一部であるパスワード有効期限フィールドについての説明を表示するには、「ヘルプ (Help)」ボタンをクリックしてください。

5. 「了解 (OK)」をクリックします。

例 — ユーザーアカウントのパスワード有効期限の変更

次の例では、ユーザーは少なくとも 1 日 (Min Change) はパスワードを変更せずに保持しなければなりません。また、パスワードを 60 日 (Max Change) ごとに変更しなければなりません。アカウントを 10 日間 (Max Inactive) 以上使用しなかった場合、パスワードを変更しなければなりません。

Admintool: Modify User

USER IDENTITY

User Name: kryten

User ID: 1001

Primary Group: 10

Secondary Groups: staff

Comment: Kryten Series 4000 Mecha

Login Shell: C /bin/csh

ACCOUNT SECURITY

Password: Normal Password...

Min Change: 1 days

Max Change: 60 days

Max Inactive: 10 days

Expiration Date: (dd/mm/yy) None None None

Warning: days

HOME DIRECTORY

Path: /export/home/kryten

OK Apply Reset Cancel Help

▼ ユーザーアカウントを削除する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細は 76 ページの「Admintool を起動する方法」を参照してください。
2. メインウィンドウから削除するユーザーアカウントエントリを選択します。
3. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
「削除 (Delete)」ウィンドウが表示され、そのユーザーアカウントを削除してもよいかどうかをたずねられます。

4. (省略可能) チェックボックスをクリックして、ユーザーのホームディレクトリとその内容を削除します。
5. ユーザーアカウントを削除する準備が出来ていれば「了解 (OK)」をクリックします。そのユーザーアカウントエントリが、「**Admintool: ユーザー (Users)**」メインウィンドウから削除されます。

例 — ユーザーアカウントを削除する

ユーザー kryten のアカウントと /export/home/kryten ディレクトリが削除されます。



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フェース (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 SolveSM へのアクセス

Solaris ユーザー登録が完了すると、Solaris Solve 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ことができます。

基本的に、Solaris ユーザー登録を完了して、Solaris Solve にアクセスする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解決

この節では、Solaris ユーザー登録に関連した問題を解決する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 3-3 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、96ページの「Solaris ユーザー登録をやり直す方法」を参照のこと。
登録プロセスをやり直したい。	96ページの「Solaris ユーザー登録をやり直す方法」を参照のこと。

▼ 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. uprofs ファイルを削除します。

```
% rm uprofs
```


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

表 3-4 に、Solaris ソフトウェア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方法を示します。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。

表 3-4 ユーザー登録を無効にする方法

ユーザー登録を無効にするとき	方法	参照先
Solaris ソフトウェアをインストールする前	<ul style="list-style-type: none">■ 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、DISABLE=1 という行を持つ、solregis という名前のファイルを作成する finish スクリプトを作成して実行する。	『Solaris 8 のインストール (上級編)』 solregis(1) のマニュアルページ
Solaris ソフトウェアをインストールした後	<ul style="list-style-type: none">■ pkgrm コマンドを使用して SUNWsregu パッケージを削除する。■ /etc/default ディレクトリに solregis ファイルを追加する (カスタム JumpStart インストールのみ)。	第 21 章 『Solaris 8 のインストール (上級編)』 solregis(1) のマニュアルページ

サーバーとクライアントサポートの管理

ここでは、Solaris オペレーティング環境においてサーバーとクライアントのサポートを管理する方法について説明します。次の章で構成されています。

第 5 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。サポートを追加できるいろいろなシステムタイプと、選択する場合のガイドラインを説明します。

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) に関する情報を記載します。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。

- 101ページの「サーバーとクライアントとは」
- 102ページの「サポートとは」
- 102ページの「システムタイプの概要」

注 - AutoClient™ とディスクレスシステムは、Solaris 8 リリースではサポートされません。既存のクライアントシステムの管理については、『*Solstice AdminSuite 2.3 管理者ガイド*』を参照してください。

サーバーとクライアントとは

ネットワーク上のシステムは、次の 2 つに分類できます。

- サーバー - ネットワーク上の他のシステムにサービスを提供するシステムです。ファイルサーバー、ブートサーバー、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

のシステムにファイルシステムとインストールソフトウェア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です。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。

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。

サポートとは

「システムにサポートを提供する」とは、他のシステムを動作させるために必要な適切なソフトウェアとサービスを提供することです。これには次のものが含まれます。

- システムをネットワークに認識させる (ホスト名とイーサネットアドレス情報)。
- 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。
- ディスク領域が限られているか、まったくないシステムに、オペレーティングシステム (OS) のサービスを提供する。

システムタイプの概要

システムタイプは基本的に、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。表 5-1 に、各システムタイプの相違点を要約します。

表 5-1 システムタイプの概要

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	ルート (/) /usr /export/home	あり	なし	低	高
JavaStation™	なし	なし	/home	低	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、/export/swap、/export/home の各ファイルシステム。これらのファイルシステムは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します。
- 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。

- 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 4 つの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。

JavaStation クライアント

JavaStation は、システム管理が不要になるように設計されたクライアントです。このクライアントは Java™ を最適化します。つまり、JavaStation クライアントはネットワークの利点を活用して、Java アプリケーションとそのサービス、完全に統合されたシステムとネットワーク管理を提供します。JavaStation にはローカルの管理が必要ありません。つまり、ブート、管理、データの格納は、サーバーが処理します。

システムのシャットダウンとブート

ここでは、Solaris 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。次の章で構成されています。

第 7 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 8 章	実行レベルとブートファイルについて説明します。
第 9 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 10 章	SPARC システムのブート手順について説明します。
第 11 章	IA システムのブート手順について説明します。
第 12 章	ブートプロセスの概要を説明します。SPARC システムおよび IA システムのブートに使用するプラットフォーム固有のハードウェアについても説明します。

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資源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ブ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。

- 107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」
- 108ページの「シャットダウンとブートについての参照先」
- 109ページの「シャットダウンとブートの用語」
- 110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」
- 110ページの「システムのブートに関するガイドライン」
- 111ページの「再構成用ブートの実行」
- 111ページの「システムをシャットダウンする場合」
- 112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 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。

DHCP によるネットワーク経由のシステムのブート

今回のリリースの Solaris には、ネットワーク経由でシステムをブートするための DHCP (Dynamic Host Configuration Protocol) 機能が追加されました。以前の RARP/bootparams に基づいたネットワークブートも利用できます。

DHCP ブートを使用するには、DHCP サーバーがネットワークにインストールされ、構成されていなければなりません。DHCP サーバーの設定については、『Solaris のシステム管理 (第 3 巻)』の「DHCP サービスの設定」を参照してください。

SPARC システムをネットワーク経由でブートする方法については、151ページの「SPARC: システムをネットワーク経由でブートする方法」を参照してください。IA システムをネットワーク経由でブートする方法については、166ページの「IA: システムをネットワーク経由でブートする方法」を参照してください。

IA: Solaris ブートフロッピーディスクを使用しない CD-ROM からのブート

ブートフロッピーディスクを使用しないブート機能をサポートしている IA システムでは、Solaris ブートフロッピーディスクがなくても、ローカルに接続された CD-ROM から Intel 版 Solaris 8 を直接ブートできます。

この機能をサポートしていないシステムでは、Solaris ブートフロッピーディスクを利用できます。

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、IA システムのブートについては、『Solaris 8 インストールガイド (Intel 版)』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。

手順	参照先
SPARC または IA のシステムのシャット ダウン	第 9 章
SPARC システムのブート	第 10 章
IA システムのブート	第 11 章
電源管理ソフトウェアによる SPARC システ ムの管理	『電源管理システム ユーザーマニュアル』

シャットダウンとブートの用語

ここ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。
- 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指すもので、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ます。
 - 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ます。
 - 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します。

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` コマンドを使用します。`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資源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。
- どちらのコマンド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 – すべてのシステム資源を使用でき、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 資源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。
 - 実行レベルについての詳細は、第 8 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。IA 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 IA 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 12 章を参照してください。

再構成用ブートの実行

再構成用ブートは、システムに新しいハードウェアを追加したときに行います。表 7-1 に、どの場合にどの再構成手順を使用するかを示します。

表 7-1 再構成手順

システム再構成の内容	参照先
ディスクの追加	第 30 章または第 31 章
周辺装置の追加	312ページの「周辺デバイスを追加する方法」

システムをシャットダウンする場合

表 7-2 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 7-2 システムのシャットダウン

管理作業	以下の実行レベルに変更	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 9 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 9 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 (シングルユーザーモード)。	第 9 章
/etc/system などのシステム構成ファイルを修正する。	112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 24 章
ブート失敗の原因となっている重要なシステムファイルを修正する。	112ページの「システムをブートする場合」を参照。	なし
カーネルデバッガ (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 9 章
ハング状態から回復する。または、クラッシュダンプを強制する。	112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 9 章を参照してください。

システムをブートする場合

表 7-3 に、システム管理作業とそれに伴って必要となるブートの種類を示します。

表 7-3 システムのブート

リポート前に実行した管理作業	使用するブート種類	SPARC のブート手順の参照先	IA 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した。	システムの電源を再投入する。	第 9 章	第 9 章
/etc/system ファイル内のカーネルパラメータを変更した。	システムを実行レベル 3 にリポートする (NFS 資源を共有できる状態のマルチユーザーモード)	147ページの「SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法」	162ページの「IA: システムを実行レベル 3 (マルチユーザー状態) にする方法」
ファイルシステムを保守した (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 Ctrl-d を押して、システムを実行レベル 3 に戻す。	148ページの「SPARC: システムを実行レベル S (シングルユーザー状態) にする方法」	163ページの「IA: システムを実行レベル S (シングルユーザー状態) にする方法」
/etc/system などのシステム構成ファイルを修正した。	対話式ブート。	149ページの「SPARC: システムを対話式でブートする方法」	164ページの「IA: 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した (または、システムからハードウェアを削除した)。	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投入する)。	394ページの「SPARC: 二次ディスクを接続してブートする方法」	第 31 章
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した。	kabd をブートする。	157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」	171ページの「IA: 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正した。	回復ブート。	153ページの「SPARC: システムを復元するためにブートする方法」	167ページの「IA: システムを復元するためにブートする方法」
ハング状態から回復した。または、クラッシュダンプを強制した。	回復ブート。	156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」	172ページの「IA: 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」

システムブートの例については、第 10 章または第 11 章を参照してください。

実行レベルとブートファイル (手順)

この章では、システムのシャットダウンとブートに関するガイドラインを示し、実行レベルとブートファイル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。

- 117ページの「システムの実行レベルを確認する方法」
- 122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」
- 124ページの「実行制御スクリプトを追加する方法」
- 125ページの「実行制御スクリプトを無効にする方法」

この章で説明する主な内容は次のとおりです。

- 115ページの「実行レベル」
- 117ページの「/etc/inittab ファイル」
- 121ページの「実行制御スクリプト」
- 125ページの「実行制御スクリプトのまとめ」

実行レベル

システムの「実行レベル」(init 状態ともいう)は、ユーザーが利用できるサービスと資源を定義します。システムが1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 (表 8-1 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 8-1 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	すべて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	ユーザーがログインできる状態ですべての使用可能なファイルシステムにアクセスする。
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全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 資源を共有した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。NFS の資源共有が可能な状態。
4	マルチユーザー状態 (予備)		このレベルは現在使用されていない。
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	リブート状態	リブート	システムをシャットダウンして実行レベル 0 にした後、マルチユーザー状態 (または、 <code>inittab</code> ファイルに指定されたデフォルトのレベル) でリブートする。

▼ システムの実行レベルを確認する方法

`who -r` コマンドを使用してシステムの実行レベル情報を表示すれば、システムの実行レベルがわかります。ただし、実行レベル 0 を除きます。

```
$ who -r
```

例—システムの実行レベルを確認する

```
$ who -r
.          run-level 3  Sep  1 14:45    3      0  S
$
```

実行レベル 3	現在の実行レベル。
Sep 1 14:45	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。
0	最後にリブートしてからこの実行レベルになった回数。
S	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

表 8-2 に、`inittab` エントリの各フィールドを要約します。

表 8-2 `inittab` ファイルのフィールド

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、 <code>initdefault</code> 、 <code>sysinit</code> 、 <code>boot</code> 、 <code>bootwait</code> 、 <code>wait</code> 、および <code>respawn</code> 。
<i>process</i>	実行するコマンド。

例 — デフォルトの `inittab` ファイル

以下の例は、デフォルトの `inittab` ファイルです。

```
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
```

(続く)

```
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. シングルユーザーモードを定義します。
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` ファイルを読み取って、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についての詳細は、`init(1M)` のマニュアルページを参照してください。

表 8-3 に、実行レベル 3 の `action` フィールドで使用するキーワードについて説明します。

表 8-3 実行レベル 3 の `action` キーワードの説明

キーワード	指定されたプロセスの実行方法
<code>powerfail</code>	システムが電源切断シグナルを受信したときだけプロセスを実行する。
<code>wait</code>	指定されたプロセスの終了を待つ。
<code>respawn</code>	プロセスがまだ起動されていない場合は起動する。プロセスがすでに起動されている場合は、 <code>inittab</code> ファイルの検索を続ける。

表 8-4 に、実行レベル 3 で実行されるプロセス (またはコマンド) について説明します。

表 8-4 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 <code>powerfail</code> 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>/var</code> 、 <code>/var/adm</code> のファイルシステムをマウントしてチェックする。
<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 (マルチユーザーモード) に移行する。

表 8-4 実行レベル 3 のコマンドの説明 続く

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

実行制御スクリプト

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。これらのスクリプトは、`/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
K07dmi          S70uucp        S75cron        S91afbinit
K07snmpdx      S71ldap.client S75flashprom   S91ifbinit
```

K28nfs.server	S71rpc	S75savecore	S92volmgt
README	S71sysid.sys	S76nscd	S93cacheos.finish
S01MOUNTFSYS	S72autoinstall	S80PRESERVE	S94ncalogd
S05RMTMPFILES	S72inetsvc	S80lp	S95IIim
S20syssetup	S72slpd	S80spc	S95amiserv
S21perf	S73cachefs.daemon	S85power	S95ocfserv
S30sysid.net	S73nfs.client	S88sendmail	S99audit
S40llc2	S74autofs	S88utmpd	S99dtlogin
S47asppp	S74syslog	S89bdconfig	
S69inet	S74xntpd	S90wbem	

/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、システムプロセスが終了 (kill) します。名前が S で始まるスクリプトを実行すると、システムプロセスが起動されます。

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、表 8-5 から表 8-10 に要約を示します。

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

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) 機能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. 機能を停止します。

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

- 機能を再開します。

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

- pgrep コマンドを使用して、サービスが停止または起動しているかを確認します。

```
# 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 サーバーの機能を停止するには、次のように入力します。

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
#
```

NFS サービスを再開するには、次のように入力します。

```
# /etc/init.d/nfs.server start
# pgrep -f nfs
141
143
245
247
# pgrep -f nfs -d, | xargs ps -fp
root  141    1 40   Jul 31 ?    0:00 /usr/lib/nfs/statd
root  143    1 80   Jul 31 ?    0:01 /usr/lib/nfs/lockd
root  245    1 34   Jul 31 ?    0:00 /usr/lib/nfs/nfsd -a 16
root  247    1 80   Jul 31 ?    0:02 /usr/lib/nfs/mountd
```

実行制御スクリプトを追加する

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て、`rcn.d` ディレクトリ内の適切なファイルへのリンクを作成します。

実行制御スクリプトの命名法についての詳細は、`/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。以下に、実行制御スクリプトの追加方法を説明します。

▼ 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. `ls` コマンドを使用して、スクリプトが指定されたディレクトリ内のファイルにリンクされているかどうかを確認します。

```
# 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

```
# cp xyz /etc/init.d
# 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、スクリプト名の先頭にドット (.) を付けます。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` というスクリプト名を変更しますが、元のスクリプトも保存しています。

```
# cd /etc/rc2.d  
# mv S100datainit _S100datainit
```

実行制御スクリプトのまとめ

表 8-5 /sbin/rc0 スクリプト

スクリプト名	説明
/sbin/rc0	以下の作業を実行する。 <ul style="list-style-type: none"> ■ 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

表 8-6 /sbin/rc1 スクリプト

スクリプト名	説明
/sbin/rc1	/etc/rc1.d ディレクトリ内のスクリプトを実行して以下の作業を実行する。 <ul style="list-style-type: none"> ■ 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 ■ システムをシングルユーザーモードに移行

表 8-7 /sbin/rc2 スクリプト

スクリプト名	説明
/sbin/rc2	<p>/etc/rc2.d ディレクトリ内のスクリプトを実行して以下の作業を実行する。</p> <ul style="list-style-type: none"> ■ 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 内に保存する。 ■ /tmp ディレクトリにあるすべてのファイルの削除 ■ システムアカウントの設定 ■ デフォルトのルータの設定 ■ NIS ドメインと ifconfig ネットワークマスクの設定 ■ インストール媒体またブートサーバー (/ .PREINSTALL または / AUTOINSTALL のどちらかが存在する場合) からシステムをリブートする。 ■ 必要に応じて、inetd、rpcbind、named を起動する。 ■ Kerberos のクライアント側デーモン kerbd を起動する。 ■ NIS デーモン (ypbind) および NIS+ デーモン (rpc.nisd) を起動する。どちらを起動するかは、システムが NIS または NIS+ のどちら用に設定されているか、またシステムがクライアントとサーバーのどちらであるかによって異なる。 ■ keyserd、statd、lockd、xntpd、utmpd の起動 ■ すべての NFS エントリのマウント ■ ncsd (ネームサービスキャッシュデーモン) の起動 ■ automount、cron、LP 印刷サービス、sendmail、utmpd、vold の各デーモンの起動

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。

表 8-8 /sbin/rc3 スクリプト

スクリプト名	説明
/sbin/rc3	<p>/etc/rc3.d ディレクトリ内のスクリプトを実行して以下の作業を行う。</p> <ul style="list-style-type: none"> ■ sharetab のクリーンアップ ■ nfsds の起動 ■ mountd の起動 ■ ブートサーバーの場合は、rarpd、rpc.bootparamd、rpld を起動する。 ■ snmpdx を起動する (Solstice Enterprise Agent™ プロセス)。

表 8-8 /sbin/rc3 スクリプト 続く

表 8-9 /sbin/rc5 スクリプトと /sbin/rc6 スクリプト

スクリプト名	説明
/sbin/rc5 および /sbin/rc6	<p>/etc/rc0.d/k* ディレクトリ内のスクリプトを実行して以下の作業を行う。</p> <ul style="list-style-type: none"> ■ 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

表 8-10 /sbin/rcS スクリプト

スクリプト名	説明
/sbin/rcS	<p>/etc/rcS.d ディレクトリ内のスクリプトを実行して、システムを実行レベル S に移行する。以下の作業は、これらのスクリプトが実行する。</p> <ul style="list-style-type: none"> ■ 最小限のネットワークの確立 ■ 必要なら、/usr をマウントする。 ■ システム名の設定 ■ / および /usr ファイルシステムのチェック ■ 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 ■ シングルユーザーモードでマウントされる他のファイルシステムをチェックしてマウントする。

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。

- 132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」
- 132ページの「サーバーをシャットダウンする方法」
- 136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」
- 138ページの「すべてのデバイスの電源を落とす方法」

この章で説明する主な内容は次のとおりです。

- 129ページの「システムをシャットダウンする場合」
- 130ページの「システムをシャットダウンするには」
- 131ページの「デバイスの電源を落とす場合」
- 131ページの「システムのダウンについてユーザーに通知する」

使用可能な実行レベルについての概要は、第 8 章を参照してください。

システムをシャットダウンする場合

Solaris ソフトウェアは、電子メールとネットワーク資源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か、一部のシステムサービスしか提供しない中間の実行レベルまで移行する必要があります。たとえば次のような場合です。

- ハードウェアを追加または削除する。
- 予告済みの停電に備える。
- 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についての詳細は、第 7 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、『電源管理システム ユーザーマニュアル』を参照してください。

システムをシャットダウンするには

システムをシャットダウンする第 1 の方法は、`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「正常にシャットダウン」します。つまり、すべてのファイルシステムに対する変更は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。

システムのアボー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、正常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。システムの復元手順については、第 10 章または第 11 章を参照してください。

表 9-1 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 9-1 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、およびサーバー資源をマウントしているシステムに、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。
<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ロン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ので、シャットダウンにかかる時間は短くて済む。

表 9-1 シャットダウンコマンド 続く

コマンド	説明	用途
reboot	ディスクを同期させ、ブート命令を uadmin システムコールに渡す実行可能ファイル。実際にプロセッサを停止するのは、uadmin システムコールである。	推奨されない。代わりに、init コマンドを使用する。
halt	ディスクを同期させ、プロセッサを停止する実行可能ファイル。	/etc/rc0 (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のマウントを解除する) スクリプトを実行しないので、推奨されない。

注 - この章およびこのマニュアル全体を通して、`/usr/ucb/shutdown` コマンドではなく `/usr/sbin/shutdown` コマンドを使用します。

デバイスの電源を落とす場合

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。
- 予測可能な停電や自然災害 (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。

シャットダウン手順を実行してから、すべてのデバイスの電源を落としてください。

システムのダウンについて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およびシステム資源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ておく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知るには、`who` コマンドを使用します。`who` コマンドは、システムの現在の実行レベルを知りたい場合にも使用できます。システム上のどのユーザーに通知する必要があるかを知るには、`who` コマンドを使用します。`who(1)` コマンドは、117ページの「システムの実行レベルを確認する方法」で説明されているシステムの現在の実行レベルを知りたい場合にも使用できます。

▼ 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. `who` コマンドを使用して、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

`who` コマンドの出力例を以下に示します。

```
$ who
holly 1      console 2      May  7 07:30
kryten     pts/0 2      May  7 07:35 3 (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

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どうか調べます。

```
# who
```

ログインしているすべてのユーザーが表示されます。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もできます。

3. shutdown (1M) コマンドを使用してシステムをシャットダウンします。

```
# shutdown -iinit-state -ggrace-period -y
```

<code>-iinit-state</code>	システムをデフォルトの S 以外の init 状態にする。0、1、2、5、6 のいずれかを指定できる。
<code>-ggrace-period</code>	シャットダウンするまでの時間 (秒) を指定する。デフォルトは 60 秒。
<code>-y</code>	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。

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、**Ctrl-d** を押してデフォルトの実行レベルに戻ります。

7. 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する方法を以下の表に要約します。

移行後の実行レベル	SPARC システム上の場合	IA システム上の場合
実行レベル S (シングルユーザーモード)	#	#
実行レベル 0 (電源切断状態)	ok または >	type any key to continue
実行レベル 3 (リモート資源が共有されたマルチユーザー状態)	hostname console login:	hostname console login:

SPARC: 例 — システムを実行レベル S にする (サーバー)

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 (シングルユーザーモード) にしています。

```
# who
root      console      Jul 14 13:53
# shutdown -g180 -y

Shutdown started.      Wed Jul 14 13:55:55 MDT 1999

Broadcast Message from root (console) on earth Wed Jul 14 13:55:56...
The system earth will be shut down in 3 minutes
.
.
.
Broadcast Message from root (console) on earth Wed Jul 14 13:58:28...
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.
Jul 14 13:59:15 earth /usr/sbin/vold[376]: problem unmounting /vol;
Print services stopped.
Jul 14 13:59:16 earth syslogd: going down on signal 15
Killing user processes: done.

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
```

(続く)

```
(or give root password for system maintenance): xxx
Entering System Maintenance Mode ...
#
```

SPARC: 例 — システムを実行レベル 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      Jul 14 14:01
rimmer    pts/0          Jul 14 14:03   (starbug)
pmorph    pts/1          Jul 14 14:04   (bluemidget)
# shutdown -i0 -g300 -y
Shutdown started.   Wed Jul 14 14:05:03 MDT 1999

Broadcast Message from root (console) on earth Wed Jul 14 14:05:03...
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、138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 — システムをリブートして実行レベル 3 にする (サーバー)

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      Jul 14 14:14
rimmer    pts/0          Jul 14 14:15   (starbug)
pmorph    pts/1          Jul 14 14:15   (bluemidget)
# shutdown -i6 -g120 -y
Shutdown started.      Wed Jul 14 14:16:08 MDT 1999

Broadcast Message from root (console) on earth Wed Jul 14 14:16:08...
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、すべてのファイル資源が使用でき、ユーザーがログインできる実行レベル 3 に戻ることになるでしょう。システムをマルチユーザー状態に移行する手順については、第 10 章または第 11 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

1. スーパーユーザーになります。
2. `init(1M)` コマンドを使用してシステムをシャットダウンします。


```
# init run-level
```

run-level

新しい実行レベル

3. 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する方法を以下の表に要約します。

移行後の実行レベル	SPARC システムの場合	IA システムの場合
実行レベル S (シングルユーザーモード)	#	#
実行レベル 2 (マルチユーザー状態)	#	#
実行レベル 0 (電源切断状態)	ok または >	type any key to continue
実行レベル 3 (リモート資源が共有されたマルチユーザー状態)	hostname console login:	hostname console login:

IA: 例 — システムを実行レベル 0 にする (スタンドアロン)

次の例では、init コマンドを使用して、スタンドアロンの IA 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 に移行してすべてのデバイスの電源を落とす場合は、138 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 — システムを実行レベル S にする (スタンドアロン)

次の例では、init コマンドとを使用して、スタンドアロンの SPARC システムを実行レベル S (シングルユーザー状態) 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、すべてのファイル資源が使用でき、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状態に移行する手順については、第 10 章または第 11 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方法

1. 以下の表を参考にして、該当するシャットダウン作業を実行します。

サーバーをシャットダウンする場合	スタンドアロンシステムをシャットダウンする場合
132ページの「サーバーをシャットダウンする方法」を参照	136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。必要なら、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。
CPU の電源が入ると、システムは実行レベル 3 になります。

SPARC: システムのブート (手順)

この章では、OpenBoot™ PROM モニターを使う手順と、SPARC システムを異なる実行レベルにブートする手順を説明します。

この章で説明する手順は次のとおりです。

- 142ページの「SPARC: ok プロンプトに切り換える方法」
- 142ページの「SPARC: システムの PROM リリースを確認する方法」
- 143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」
- 145ページの「SPARC: システムをリセットする方法」
- 147ページの「SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法」
- 148ページの「SPARC: システムを実行レベル S (シングルユーザー状態) にする方法」
- 149ページの「SPARC: システムを対話式でブートする方法」
- 151ページの「SPARC: システムをネットワーク経由でブートする方法」
- 153ページの「SPARC: システムを復元するためにブートする方法」
- 155ページの「SPARC: 復元を目的としてシステムを停止する方法」
- 156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」
- 157ページの「SPARC: 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」

ブート手順の概要については、第 12 章を参照してください。ブートの問題を解決する方法については、『Solaris のシステム管理 (第 2 巻)』の「リブートが失敗した場合の対処方法」を参照してください。

IA システムをブートする手順については、第 11 章を参照してください。

SPARC: ブート PROM の使用方法

システム管理者は、通常は PROM レベルを使ってシステムをブートします。ただしシステムをマルチユーザー状態にする前に、どのデバイスからブートするかを設定したり、ハードウェア診断プログラムを実行するなど、システム動作方法を変更する必要がある場合があります。

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には、デフォルトのブートデバイスを変更しなければなりません。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には、ネットワークブートの方針を変更しなければなりません。

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: ok プロンプトに切り換える方法

システムを停止すると、PROM モニターは `>` または `ok` を表示します。

SPARC システムで、`>` プロンプトから `ok` プロンプトに切り換えるには、次のコマンドを入力します。

```
> n
ok
```

この節で示す例では、すべて `ok` プロンプトを使います。

▼ SPARC: システムの PROM リリースを確認する方法

システムの PROM リリースレベルを `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 #nnnnnnnn.
Ethernet address 8:0:20:a5:d1:3b, Host ID: nnnnnnnn.
```

ハードウェア構成情報が表示され、その中に **PROM** のリリース番号があります。
PROM リリースレベルが、ROM Rev 番号で示されます。

▼ SPARC: デフォルトのブートデバイスを変更する方法

1. スーパーユーザーになります。
2. `init(1M)` コマンドを使って、システムを停止します。

```
# init 0
```

3. > **PROM** プロンプトが表示されたら、`n` を入力して **Return** キーを押します。

```
> n
ok
```

ok PROM プロンプトが表示されます。

4. `setenv` コマンドを使って、`boot-device` の設定を変更します。

```
ok setenv boot-device device [n]
```

`boot-device` ブートするデバイスを設定するパラメータを指定する。

`device[n]` `boot-device` の値 (`disk` または `net`) を指定する。`n` は
ディスク番号。

ディスク番号を確認したい場合は、`probe-scsi-all` コマンドを使います。

5. `printenv` コマンドを使って、デフォルトのブートデバイスの変更を確認します。

```
ok printenv boot-device
```

6. reset コマンドを使って新しい boot-device 値を保存します。

```
ok reset
```

新しい boot-device 設定が、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-III 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #nnnnnnnn.
Ethernet address 8:0:20:a5:d3:4b, Host ID: nnnnnnnn.

Boot device: disk File and args:
SunOS Release 5.8 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
```

(続く)


```
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 #nnnnnnnn.
Ethernet address 8:0:20:a3:d54:4b, Host ID: nnnnnnnn.

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

表 10-1 に、この章で説明するブート方法の要約を示します。

表 10-1 ブート方法の説明

システムのブート方法	このブート方法が必要な場合	参照先
実行レベル 3 (NFS 資源を共有するマルチユーザー状態)	システムを停止するか、また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。これはデフォルトのブートレベルで、すべての資源が利用でき、そしてユーザーがシステムにログインできる。	147ページの「SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法」
実行レベル S (シングルユーザー状態)	ファイルシステムのバックアップなどのなんらかのシステム保守作業を行なった後。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。	148ページの「SPARC: システムを実行レベル S (シングルユーザー状態) にする方法」
対話式でブート	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。このブート方法を使えば、システムファイルまたはカーネルに問題がある場合、プロンプトが表示されたときに、これらのファイルに対して別のパス名を指定すれば簡単に復元できる。他のシステムプロンプトにデフォルトの設定を使う。	149ページの「SPARC: システムを対話式でブートする方法」
ネットワーク経由でブート	システムをネットワーク経由でブートする。この手順は、ブートサーバー上で必要な設定が完了していることを前提とする。	151ページの「SPARC: システムをネットワーク経由でブートする方法」
ローカル CD-ROM またはネットワークからブートして復元する	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。また、この形式のブートはオペレーティングシステムの新しいリリースをインストール (あるいは、アップグレード) するのにも使用する。	153ページの「SPARC: システムを復元するためにブートする方法」
kadb を使ってブート	オペレーティングシステムのコアダンプを保存して、システムの障害を追跡する場合。	155ページの「SPARC: 復元を目的としてシステムを停止する方法」

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。この後に示す手順では、ok PROM プロンプトからさまざまな状態でブートする方法を説明します。

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 8 章を参照してください。

▼ SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法

1. ブートコマンドを使って実行レベル 3 にします。

```
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。

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 (マルチユーザー状態) にする

次に、システムを実行レベル 3 にするメッセージの例を示します。

```
ok boot
SPARCstation 10 (1 X 390Z50)
ROM Rev. 2.14, 32 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Rebooting with command:
Boot device: /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,8...
SunOS Release 5.8 Version Generic 32-bit
Copyright (c) 1983-2000 by Sun Microsystems, Inc.
configuring IPv4 interfaces: le0.
Hostname: earth
```

(続く)

```
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdisk/c0t3d0s7: is clean.
NIS domainname is Solar.COM
starting rpc services: rpcbind keyserver ypbind done.
Setting netmask of le0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/
4: gateway earth
syslog service starting.
Print services started.
volume management starting.
The system is ready.

earth console login:
```

▼ SPARC: システムを実行レベル S (シングルユーザー状態) にする方法

1. `boot -s` コマンドを使って、システムを実行レベル **S** にします。

```
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. `who -r` コマンドを使って、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      May  2 07:39      3      0  S
```

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 (マルチユーザー状態) にする

次に、実行レベル S になるシステムの表示例を示します。

```
ok boot -s
.
.
.
SunOS Release 5.8 Version Generic 32-bit
Copyright (c) 1983-2000 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 5.8  generic August 1999
# who -r
.      run-level S      Jul 14 11:37      S      0  ?
(システム保守作業を行う)
# <Control-d> を押す
```

▼ SPARC: システムを対話式でブートする方法

1. `boot -a` コマンドを使って、システムを対話式でブートします。

```
ok boot -a
```

2. 下の表に示すように、システムプロンプトに答えてください。

システムの表示	操作
Enter filename [kernel/unix]:	ブートに使用する別の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カーネルを使う (/platform/'uname -m'/kernel/unix)。
Name of default directory for modules [/platform/'uname -m'/kernel /kernel /usr/kernel]:	ブートに使用する別の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カーネルを使う (/platform/'uname -m'/kernel/unix)。
Name of system file [/etc/system]:	代替のシステムの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 ファイルを使う。
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形式を使う。つまり、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 を使う。
Enter physical name of root device [physical_device_name]:	代替の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う。

3. 上の表の質問に回答するためのプロンプトが表示されない場合、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 ([] で囲まれています) 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 5.8 Version Generic 64-bit
Copyright (c) 1983-2000 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.
earth console login:
```

▼ SPARC: システムをネットワーク経由でブートする方法

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経由でブートできます。たとえば、スタンドアロンシステムがローカルディスクからブートできない場合、スタンドアロンシステムを一時的にネットワーク経由でブートできます。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、143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、2つのネットワーク構成ブートの方法、RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol) または DHCP (Dynamic Host Configuration Protocol) を選択できます。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。

注 - DHCP ネットワークブートを使用するには、Sun Ultra システムの PROM のバージョンが 3.25 以降でなければなりません。

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、どちらを使用するかは、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、PROM レベルで永久に保存できます。つまり、この設定はシステムをリブートしても変更されません。次の `nvalias` コマンドの例は、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 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。これらのコマンドを使用する方法については、『*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. ネットワーク経由でブートする方法を決定し、次の中から **1** つを選択します。
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経由でブートするには、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 設定を変更している場合、`boot net` と入力するだけで、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経由でブートするには、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。デフォルトで DHCP でブートするように PROM 設定を変更している場合だけ、`boot net:rarp` と指定しなければなりません。

▼ SPARC: システムを復元するためにブートする方法

この手順は、`/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必要です。

システムのデバイス名を調べたい場合は、第 26 章を参照してください。

1. **Solaris** インストール CD またはネットワークからブートしているかどうかによって、次のそれぞれの手順に従ってください。

ブート方法	手順
Solaris インストール CD	<ol style="list-style-type: none">1. Solaris インストール CD を CD キャディに挿入する。2. CD キャディを CD-ROM ドライブに挿入する。3. インストール CD からブートして、シングルユーザーモードにする。 <pre>ok boot cdrom -s</pre>
インストールサーバーまたはリモート CD ドライブが利用できる場合は、ネットワークからブートする。	次のコマンドを使う。 <pre>ok boot net -s</pre>

2. 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
```

3. 新しくマウントしたディレクトリに移動します。

```
# cd /a/directory
```

4. 端末タイプを設定します。

```
# TERM=sun
# export TERM
```

5. 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
```

6. ルート (/) ディレクトリに移動します。

```
# 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:
```

SPARC: 例 — システムを復元するためにブートする

次に、ローカル CD-ROM からブートした後、重要なシステムファイルを修復する例として、/etc/passwd を使用した場合は示します。

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=sun
# export TERM
# vi passwd
(無効なエントリを削除する)
```

```
# cd /  
# umount /a  
# init 6
```

▼ SPARC: 復元を目的としてシステムを停止する方法

1. システムのアボ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。

2. sync コマンドを使って、ディスクを同期させます。

```
ok sync
```

3. syncing file systems. . . というメッセージが表示されたら、システムのアボートキーシーケンスをもう一度押します。
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。
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 — 復元のために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: クラッシュダンプを強制してシステムをリブートする

問題を対処するために、オペレーティングシステムのクラッシュダンプを保存しておく必要があります。savecore 機能とその設定方法については、『Solaris のシステム管理 (第 2 巻)』の「システムクラッシュ情報の生成と保存」で説明しています。この節では、savecore 機能が有効な場合に、システムをリブートする方法だけを説明します。

▼ 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

1. 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。
モニターに ok PROM プロンプトが表示されます。
2. ok プロンプトで、sync コマンドを使ってディスクを同期させ、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
```

▼ 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。
モニターに、ok PROM プロンプトが表示されます。
2. ok プロンプトで、sync コマンドを使ってディスクを同期させ、クラッシュダンプを書き出します。

```
> n  
ok sync
```

3. syncing file systems. . .メッセージが表示されたら、もう一度システムのアボートキーシーケンスを入力します。
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ガ (kadb) を使用してブートしていることを確認します。

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

SPARC: 例 — カーネルデバッガ (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
```

IA: システムのブート (手順)

この章では、IA システムをブートする手順を説明します。

この章で説明する手順は次のとおりです。

- 162ページの「IA: Solaris Device Configuration Assistant をブートする方法」
- 162ページの「IA: システムを実行レベル 3 (マルチユーザー状態) にする方法」
- 163ページの「IA: システムを実行レベル S (シングルユーザー状態) にする方法」
- 164ページの「IA: システムを対話式でブートする方法」
- 166ページの「IA: システムをネットワーク経由でブートする方法」
- 167ページの「IA: システムを復元するためにブートする方法」
- 171ページの「IA: 復元を目的としてシステムを停止する方法」
- 171ページの「IA: 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」
- 172ページの「IA: 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」

ブート手順の概要については、第 12 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 10 章を参照してください。

IA: システムのブート

表 11-1 に、この章で説明するブート方法の要約を示します。

表 11-1 ブート方法の説明

システムのブート方法	このブート方法が必要な場合	参照先
Solaris Device Configuration Assistant (デバイス構成用補助) を実行する	システムのハードウェア構成を変更した後。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、間違っって構成したハードウェアを構成し直すなど、デバイスやブートに関連する作業を行うことができる。	162ページの「IA: Solaris Device Configuration Assistant をブートする方法」
実行レベル 3 (マルチユーザー状態)	システムを停止するか、また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。これはデフォルトのブートレベルで、すべての資源が利用でき、そしてユーザーがシステムにログインすることができる。	162ページの「IA: システムを実行レベル 3 (マルチユーザー状態) にする方法」
実行レベル S (シングルユーザー状態)	ファイルシステムのバックアップなどのなんらかのシステム保守作業を行なった後。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。	163ページの「IA: システムを実行レベル S (シングルユーザー状態) にする方法」
対話式でブート	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。この形式のブートを使えば、システムファイルまたはカーネルに問題がある場合、プロンプトが表示されたときに、これらのファイルに対して別のパス名を指定すれば簡単に復元できる。他のシステムプロンプトにデフォルトの設定を使う。	164ページの「IA: システムを対話式でブートする方法」
ネットワーク経由でブート	システムをネットワーク経由でブートする。この手順は、ブートサーバー上で必要な設定が完了していることを前提とする。	166ページの「IA: システムをネットワーク経由でブートする方法」
ローカル CD-ROM またはネットワークからブートして復元する	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。また、この形式のブートは、オペレーティングシステムの新しいリリースをインストール (またはアップグレード) するのにも使用する。	167ページの「IA: システムを復元するためにブートする方法」

表 11-1 ブート方法の説明 続く

システムのブート方法	このブート方法が必要な場合	参照先
Solaris カーネルデバugg (kadb) の実行	システムの問題の原因を調べる。	171ページの「IA: カーネルデバugg (kadb) を使ってシステムをブートする方法」
クラッシュダンプの作成	システムの問題の原因を調べるために、オペレーティングシステムのコアダンプを保存する。	172ページの「IA: 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」

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、電源のオン・オフ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。

IA: 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) 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 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 Installation CD

次の節では、Solaris Device Configuration Assistant フロッピーディスクを挿入して、Configuration Assistant 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 からのブートをサポートしている場合は、Solaris Installation CD を挿入して、Configuration Assistant をブートすることも可能です。

▼ IA: Solaris Device Configuration Assistant をブートする方法

1. **Solaris Device Configuration** フロッピーディスクまたは **Solaris Installation CD** 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、任意のキーまたはリセットボタン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、電源スイッチを押してシステムを起動します。
3. **Configuration Assistant** の最初のメニューが、数分後に表示されます。

▼ IA: システムを実行レベル 3 (マルチユーザー状態) にする方法

1. `Type any key to continue` プロンプトが表示されている場合は、任意のキーまたはリセットボタン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、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** キーを押してください。
5 秒以内に選択しないと、システムは自動的に実行レベル 3 になります。
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:
```

IA: 例 — システムを実行レベル 3 (マルチユーザー状態) にする

```
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:

```

▼ IA: システムを実行レベル S (シングルユーザー状態) にする方法

1. Type any key to continue プロンプトが表示されている場合は、任意のキーまたはリセットボタン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、電源スイッチを押してシステムを起動します。数分後、「Current Boot Parameters」メニューが表示されます。
2. `b -s` と入力し、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と、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。

IA: 例 — システムを実行レベル S (シングルユーザー状態) にする

```
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
```

▼ IA: システムを対話式でブートする方法

1. Type any key to continue プロンプトが表示されている場合は、任意のキーまたはリセットボタン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、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. **Solaris** パーティションにアクティブを示すマークがついていない場合、その **Solaris** パーティションをリストから選択して、**Enter** キーを入力します。**30** 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と、システムは自動的に実行レベル 3 でブートします。
4. 下の表で説明するように、システムプロンプトに応答します。

システムの表示	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て Enter キーを押すか、そのまま Enter キーを押してデフォルトの modules ディレクトリパスを使用する。
Name of system file [etc/system]:	代替のシステムファイル名を入力して Enter キーを押すか、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形式を使う。つまり、ローカルディスクからのブートの場合は UFS、ネットワークのブートの場合は NFS を使う。
Enter physical name of root device [physical_device_name]:	代替のデバイス名を入力して、Enter キーを押すか、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う。

IA: 例 — システムを対話式でブートする

デフォルトを選択 ([] で囲まれています) した例を示します。

```

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

```

(続く)

```
<<< 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 5.8 Version Generic 32-bit
Copyright (c) 1983-2000 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:
```

▼ IA: システムをネットワーク経由でブートする方法

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経由でブートできます。たとえば、スタンドアロンシステムがローカルディスクからブートできない場合、このスタンドアロンシステムを一時的にネットワーク経由でブートできます。

Device Configuration Assistant (デバイス構成用補助) の「Boot Tasks」メニューにある新しいメニュー「Set Network Configuration Strategy」を使用して、適切なブート方法を選択できます。

1. ネットワーク経由でブートする方法 (**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。
2. **Device Configuration Assistant** (デバイス構成用補助) フロッピーディスクまたは **Installation CD** を適切なドライブに挿入します

3. Type any key to continue プロンプトが表示されている場合は、任意のキーまたはリセットボタン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、電源スイッチを押してシステムを起動します。
4. 「**Solaris Device Configuration Assistant**」画面で「**F2_Continue**」を押して、デバイスを走査します。
デバイスの識別が実行され、識別されたデバイスを表示する画面が表示され
ます。
5. 「**Identified Devices**」画面で「**F2_Continue**」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。
6. 「**Boot Solaris**」メニューから「**F4_Boot Tasks**」を押します。
7. 「Set Network Configuration Strategy」を選択して、「**F2_Continue**」
を押します。
8. 「RARP」または「DHCP」を選択して、「**F2_Continue**」を押します。
新しいネットワーク構成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構成方法は、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. 「**Boot Solaris**」メニューから「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。

▼ IA: システムを復元するためにブートする方法

重要なシステム資源を修復するには、次の手順に従ってシステムをブートします。
この例では、Solaris 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
法を説明しています。

以下の説明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については、第 26 章を参照してください。

Solaris Installation CD またはネットワークのどちらからブートしているかによって、対応する手順に従ってください。

1. **Solaris Installation CD** (またはネットワーク) 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) フロッピーディスクまたは **Installation CD** 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、任意のキーまたはリセットボタン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、電源スイッチを押してシステムを起動します。
 - c. 「**Solaris Device Configuration Assistant**」画面で、**F2** キー (**F2_Continue**) を押します。
デバイスが識別され、識別されたデバイスを示す画面が表示されます。
 - d. 「**Identified Devices**」画面で、**F2** キー (**F2_Continue**) を押します。
ブート可能なドライバが読み込まれます。
 - e. 「**Boot Solaris**」メニューから **CD** または **NET** をデバイスとして選択します。次に **F2** キー (**F2_Continue**) 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。
2. 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
```

3. 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
```


4. 必要に応じて、エディタを使って passwd ファイルに必要な変更を加えます。

```
# vi passwd
```

5. ルート (/) ディレクトリに移動します。

```
# cd /
```

6. /a ディレクトリのマウントを解除します。

```
# umount /a
```

7. システムをリブートします。

```
# init 6
```

8. システムが実行レベル **3** でブートしたことを確認します。

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:
```

IA: 例 — システムを復元する目的でブートする

```
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.
```

(続く)

```

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
[ ] 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
.

```

(続く)

```
.  
# cd /a/etc  
# vi passwd  
(無効なエントリを削除する)  
# cd /  
# umount /a  
# init 6
```

▼ IA: 復元を目的としてシステムを停止する方法

可能であれば、次のコマンドのどちらか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」というメッセージが表示されたら、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。

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。

▼ IA: 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、任意のキーまたはリセットボタン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、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。**r**. 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 でブートし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。

IA: 例 — システムをカーネルデバッグ (`kadb`) 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 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)
```

IA: クラッシュダンプを強制してシステムをリブートする

問題に対処するために、オペレーティングシステムのコアダンプを保存しておく必要があります。savecore 機能とその設定方法については、『Solaris のシステム管理 (第 2 巻)』の「システムクラッシュ情報の生成と保存」で説明しています。この節では、savecore 機能が有効な場合に、システムをリブートする方法だけを説明します。

▼ IA: 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します。

注 - カーネルデバッガ 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 (CDE または OpenWindows) 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。

```
Press <F1-a>
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。

ブートプロセス (参照情報)

この章では、SPARC や IA システムでブートするのに使用するハードウェア、および各プラットフォームのブートプロセスの概要を説明します。

この章の内容は次のとおりです。

- 175ページの「SPARC: ブート PROM」
- 176ページの「SPARC: ブートプロセス」
- 177ページの「IA: PC BIOS」
- 177ページの「IA: ブートサブシステム」
- 182ページの「IA: ブートプロセス」

システムのブート手順については、第 10 章または、第 11 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、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など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 [SPARC] - 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。詳細は、ご購入先にお問い合わせください。

SPARC: ブートプロセス

表 12-1 で、ブートプロセスについて説明します。

表 12-1 ブートプロセスの説明

ブートフェーズ	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。 2. PROM は一次ブートプログラム bootblk 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 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ことにある。
ブートプログラム	3. bootblk プログラムは二次ブートプログラム ufsboot を検索して実行し、それをメモリーに読み込む。 4. ufsboot プログラムが読み込まれた後、ufsboot プログラムはカーネルを読み込む。
カーネルの初期設定	5. カーネルが自身を初期設定し、ファイルの読み込みに ufsboot 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、/ufsboot プログラムの対応づけを解除し、それ自身のリソースを使って動作を続ける。
init	6. カーネルがユーザープロセスを作成し、/sbin/init プロセスを起動する。このプロセスは /etc/inittab ファイルを読んで他のプロセスを起動する。 7. /sbin/init プロセスが実行制御 (rc) 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 (/sbin/rc*) はファイルシステムをマウントして検査し、いろいろなプロセスを開始し、システム保守作業を実行する。

IA: 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 BIOS があるかどうか調べ、あれば、それぞれをロードして、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。

IA: 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えます。

- 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。何も指定しなければ、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、Device Configuration Assistant (デバイス構成用補助) にアクセスできます。
Configuration Assistant を使用すると、Solaris システムを別のブートデバイスからブートしたり、新しいハードウェアの構成や構成変更が行え、またデバイスやブートに関連する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できます。

表 12-2 に、主な IA ブートインタフェースの目的を示します。以降の節では、各ブートサブシステムを詳細に説明し、例を示します。

表 12-2 ブートサブシステム

ブートサブシステム	このサブシステムメニューの目的
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。
二次ブートサブシステム	Solaris リリースをブートするときに使用。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。 <ol style="list-style-type: none"> 1. Solaris Device Configuration Assistant ブートフロッピーディスクまたは Solaris 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 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」	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。このメニューにはブートオプションが表示される。

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。

以下の節では、各サブシステムの画面の例を示します。

IA: Solaris のブート

デバイスを識別する段階では、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資源の編集などの追加作業を可能にする。

ブート中に、Configuration Assistant は次の処理を行います。

- 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す。
- 自動ブートやプロパティ設定の編集、ネットワーク構成方針の選択などの追加作業を可能にする。

各段階のデバイス識別の例を次に示します。デバイス出力は、各システム構成によって異なります。

IA: デバイス識別段階で表示される画面

Configuration Assistant がシステム上のデバイスを識別するときに、いくつかのメニューが表示されます。

IA: Configuration Assistant 画面

この画面は、Configuration Assistant をブートしてメニューにアクセスするたびに表示されます。Configuration Assistant 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はこのメニューを省略します。

```
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.
```

(続く)

```
> 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
```

IA: Bus Enumeration 画面

この画面は、Configuration Assistant 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 ...
```

IA: 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 Configuration Assistant 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 ...
```

IA: 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。このメニューからは、Boot Solaris メニューに移動できます。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、付加的な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
```

IA: 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。

IA: Boot Solaris メニュー

Solaris ブートメニューからは、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.
```

(続く)

```

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

```

IA: 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 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したい場合は、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:

```

IA: ブートプロセス

表 12-3 で、ブートプロセスについて説明します。

表 12-3 ブートプロセスの説明

ブートフェーズ	説明
BIOS	<p>1. システムの電源をオンにすると、PC 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、エラーメッセージが表示されて、復元オプションが示される。</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. 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. 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さがし、利用可能なパーティションのメニューを表示する。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> <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 をさがし、実行する。Configuration Assistant 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取り、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。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カーネルが読み込むと、二次ブートプログラムのマッピングを解除し、それ自身の資源を使用してブートを継続する。</p>

表 12-3 ブートプロセスの説明 続く

ブートフェーズ	説明
init	9.カーネルはユーザープロセスを作成し、/sbin/init プロセスを起動する。このプロセスは /etc/inittab ファイルを読み込んで他のプロセスを起動する。 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。

取り外し可能な媒体の管理

ここでは、取り外し可能な媒体を Solaris オペレーティング環境で使用方法について説明します。次の章で構成されています。

第 14 章	自動マウントと手作業によるマウントの比較を含めて、CD とフロッピーディスクの使用方法についての概要を説明します。
第 15 章	コマンド行から CD を使用する手順と、ボリューム管理の起動と終了について説明します。
第 16 章	コマンド行からフロッピーディスクをフォーマットして使用する手順を説明します。
第 17 章	コマンド行から PCMCIA メモリカードをフォーマットして使用する手順を説明します。
第 18 章	Solaris 環境によって、フロッピーディスクおよび CD へのアクセスを容易にするための特別なマウントポイントとシンボリックリンクがどのように作成されるかについて、詳しく説明します。

CD とフロッピーディスクの使用方法 (概要)

この章では、フロッピーディスクと CD を Solaris 環境で使用するためのガイドラインを示します。

この章の内容は次のとおりです。

- 187ページの「取り外し可能な媒体の管理についての参照先」
- 188ページの「取り外し可能な媒体の機能と利点」
- 188ページの「自動マウントと手作業によるマウントの比較」
- 189ページの「フロッピーディスクと CD で行える操作」

取り外し可能な媒体の管理についての参照先

取り外し可能な媒体を管理する手順については、次を参照してください。

- 第 15 章
- 第 16 章
- 第 17 章

共通デスクトップ環境のファイルマネージャで取り外し可能な媒体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。

取り外し可能な媒体の機能と利点

Solaris 環境には、ユーザーとソフトウェア開発者用に、フロッピーディスクと CD 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。

- フロッピーディスクと CD 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。(手作業によるマウントと自動マウントの比較については、表 14-1 を参照してください。)
- スーパーユーザーでなくても、フロッピーディスクと CD にアクセスできるようにします。
- ネットワーク上の他のシステムが、システムに挿入されたフロッピーディスクと CD に自動的にアクセスできるようにします(第 15 章と第 16 章を参照してください)。

自動マウントと手作業によるマウントの比較

次の表 14-1 は、手作業によるマウント (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) の手順を比較したものです。

表 14-1 手作業によるマウントと自動マウントの比較

手順	手作業によるマウント	自動マウント
1	媒体を挿入する	媒体を挿入する
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、volcheck コマンドを使用する
3	媒体デバイスの位置を確認する	ボリューム管理は、CD およびフロッピーディスクを手作業でマウントする場合の大半の作業を、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	

表 14-1 手作業によるマウントと自動マウントの比較 続く

手順	手作業によるマウント	自動マウント
6	mount 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	媒体上のファイル进行操作する	媒体上のファイル进行操作する
9	スーパーユーザーになる	
10	媒体デバイスのマウントを解除する	
11	媒体を取り出す	媒体を取り出す
12	スーパーユーザーを終了する	

フロッピーディスクと CD で行える操作

ボリューム管理を使用すると、手作業によるマウントの場合と同様にフロッピーディスクおよび CD にアクセスできますが、処理ははるかに容易になり、スーパーユーザーになる必要もなくなります。フロッピーディスクと CD は、操作を容易にするために、覚えやすい位置にマウントされます。

表 14-2 フロッピーディスクと CD 上のデータにアクセスする方法

アクセスするデータ	操作	ファイルを検索する場所
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、volcheck と入力する。	/vol/dev/aliases/floppy0
フロッピーディスク上の raw データ	フロッピーディスクを挿入して、volcheck と入力する。	/vol/dev/aliases/floppy0
CD 上のファイル	CD を挿入して、数秒間待つ。	/cdrom/cdrom0

表 14-2 フロッピーディスクと CD 上のデータにアクセスする方法 続く

システムに複数のフロッピーディスクや CD-ROM ドライブがある場合は、そのアクセスポイントについて、次の表 14-3 を参照してください。

表 14-3 フロッピーディスクと CD にアクセスする場所

媒体デバイス	ファイルシステムにアクセスするための場所	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	/floppy/floppy0	/vol/dev/aliases/floppy0
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

コマンド行での **CD** の使用方法 (手順)

この章では、Solaris 環境においてコマンド行から CD を使用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。

- 193ページの「CD をロードする方法」
- 193ページの「CD の内容を調べる方法」
- 194ページの「CD から情報をコピーする方法」
- 195ページの「CD が使用中かどうかを調べる方法」
- 196ページの「CD を取り出す方法」
- 196ページの「他のシステム上の CD にアクセスする方法」
- 198ページの「ローカル CD を他のシステムで使用可能にする方法」
- 201ページの「音楽 CD を演奏するようにシステムを設定する方法」
- 202ページの「新しい CD-ROM ドライブ用にシステムを準備する方法」
- 203ページの「ボリューム管理を終了させる方法」
- 203ページの「ボリューム管理を再起動する方法」

CD の使用方法

表 15-1 作業マップ: 一般的な CD の使用方法

作業	説明	手順の説明
CD のロード	CD-ROM ドライブに CD を挿入する。	193ページの「CD をロードする方法」
CD の内容の確認	省略可能。 CD の内容を調べるには、/cdrom の下にある適切なディレクトリを調べる。	193ページの「CD の内容を調べる方法」
ファイルまたはディレクトリのコピー	省略可能。ファイルシステムの任意の場所からコピーをするのと同様に、CD からファイルまたはディレクトリをコピーする。	194ページの「CD から情報をコピーする方法」
CD が使用中かどうかの確認	省略可能。CD を取り出す前に、CD が使用中かどうかを確認する。	195ページの「CD が使用中かどうかを調べる方法」
CD の取り出し	作業が終了したら、CD-ROM ドライブから CD を取り出す。	196ページの「CD を取り出す方法」

CD の名前について

CD を操作する場合、名前または以下の表 15-2 に示す代替名によって、CD を識別できます。説明を簡単にするため、このマニュアル中の説明では `cdrom0` を使用しますが、実際の作業では CD 名または別の代替名を使用することもできます。

表 15-2 CD の識別方法

CD	代替名
最初の CD-ROM ドライブ	<code>cdrom0</code>
2 番目の CD-ROM ドライブ	<code>cdrom1</code>
3 番目の CD-ROM ドライブ	<code>cdrom2</code>

▼ CD をロードする方法

CD を挿入します。ランプの点滅が消えるとすぐに (およそ 5~10 秒)、CD は /cdrom にマウントされます。CD がマウントされたことを確認するには、193ページの「CD の内容を調べる方法」で説明する作業を実行してください。

注 - ほとんどの CD 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 をボリューム管理によってマウントできます。ただし、第 18 章で説明するように、UFS CD はアーキテクチャに依存するため、専用のアーキテクチャ上で使用する必要があります。CD 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 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 のラベルで確認してください。

▼ CD の内容を調べる方法

ls -L コマンドを使用して、/cdrom ディレクトリの内容を表示します。

```
$ ls -L [-l] /cdrom/cdrom0
```

-L 出力にシンボリックリンクを含める。

-l 詳細表示。出力にアクセス権と所有権を含める。

例 — CD の内容を調べる

次の例は、最初の CD-ROM ディレクトリである /cdrom/cdrom0 にロードされた CD の内容を示します。

```
$ ls -L -l /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
```

▼ CD から情報をコピーする方法

他のファイルシステムの場合と同様に、CD のファイルとディレクトリにもアクセスできます。ただし、所有権とアクセス権については注意が必要です。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(これは、CD 上のファイルには書き込み権がないためです)。ユーザー自身がアクセス権を変更する必要があります。

1. **CD** がマウントされていることを確認します。

```
$ ls /cdrom
```

ls コマンドは、マウントされた CD の内容を表示します。内容が表示されない場合は、193ページの「CD をロードする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

コピーするもの	使用するコマンド
ファイル	cp
ディレクトリ	cp -r

例 — CD から情報をコピーする

次の例は、cp コマンドを使用して1つのファイルを /cdrom/solstice_sysmgt_2_3 ディレクトリからシステムの現在のディレクトリ (「.」で表す) へコピーします。

```
$ cp /cdrom/solstice_sysmgt_2_3/README .
$ ls -l
-r--r--r--  1 pmorph  users      4618 May  9 08:09 README
```

ファイルやディレクトリを CD からユーザーのファイルシステムへコピーした場合、そのユーザーが、コピーしたファイルやディレクトリの所有者になりますが、アクセス権は CD 上のアクセス権が保持されたままです。

-r--r--r--

コピーしたファイルやディレクトリのアクセス権を変更するには、`chmod` コマンドを使用します。`chmod` コマンドの使用方法については、『Solaris のシステム管理 (第 2 巻)』の「ファイルのセキュリティの適用手順」を参照して下さい。

▼ CD 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. **CD** にアクセスしているプロセスを確認します。

`fuser(1M)` コマンドは、指定する CD に現在アクセス中のプロセスを表示します。

```
# fuser -u [-k] /cdrom/cdrom0
```

-u	CD のユーザーを表示する。
-k	CD にアクセス中のプロセスを終了させる。

`fuser` コマンドは、終了したプロセスすべてを必ず識別できるわけではありません。確認するには、`-u` オプションを付けて、もう一度このコマンドを実行して下さい。

例 — CD が使用中かどうかを調べる

以下の例では、プロセス `6400c` と `6399c` が `/cdrom/cdrom0` ディレクトリにアクセスしており、プロセスの所有者はそれぞれ `root` と `smith` です。

```
# fuser -u /cdrom/cdrom0  
/cdrom/cdrom0: 6400c(root) 6399c(smith)
```

各プロセスを別々に (スーパーユーザー権限で) 終了するか、あるいは `-k` オプションを付けて `fuser` コマンドを使用できます。このオプションはファイルシステムにアクセスしているすべてのプロセスを終了させます。

```
# fuser -u -k /cdrom/cdrom0
/cdrom/cdrom0: 6400c(root)Killed 6399c(smith)Killed
```

▼ CD を取り出す方法

1. **CD** 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 CD 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、CD は「使用中」であることを忘れないでください。CD のすべてのユーザーを検出したかどうかを確認できない場合は (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)、195ページの「CD が使用中かどうかを調べる方法」に説明されている `fuser` コマンドを使用してください。

2. **CD** を取り出します。

```
# eject cdrom0
```

▼ 他のシステム上の CD にアクセスする方法

他のシステム上の CD を各自のファイルシステムに手作業でマウントすることによって、その CD にアクセスできます。ただしこれは、他のシステムが、198ページの「ローカル CD を他のシステムで使用可能にする方法」の指示に従って CD-ROM をエクスポートしている場合にかぎります。

1. マウントポイントとして指定する既存のディレクトリを選択するか、あるいは作成します。

```
$ mkdir directory
```

directory

他のシステムの CD 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

2. マウントしたい **CD** の名前を確認します。

```
$ showmount -e system-name
export list for system-name:
/cdrom/sol_8_sparc (everyone)
```

3. スーパーユーザー権限で、**CD** をマウントします。

```
# mount -F nfs -o ro system-name:/cdrom/cd-name local-mount-point
```

<i>system-name</i>	マウントする CD を持つシステムの名前
<i>cd-name</i>	マウントしたい CD の名前
<i>local-mount-point</i>	リモート CD 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. **CD** が実際にマウントされたかどうかを確認するには、`ls` コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /cdrom
```

例 — 他のシステム上の **CD** にアクセスする

次の例では、リモートシステム `mars` 上で、`sol_8_sparc` という名前の CD を、ローカルシステムの `/cdrom` ディレクトリにマウントしています。

```
$ showmount -e starbug
export list for starbug:
/cdrom/sol_8_sparc (everyone)
$ su
Password: password
# mount -F nfs -o ro starbug:/cdrom/sol_8_sparc /cdrom
# exit
$ ls /cdrom
cdrom0      sol_8_sparc
```

▼ ローカル CD を他のシステムで使用可能にする方法

システムを設定して、その CD-ROM をエクスポートすることができます。つまり、これらのドライブ上の CD (音楽 CD を除く) を、他のシステムで使用できるようにすることができます。CD-ROM ドライブがエクスポートされると、他のシステムは、196ページの「他のシステム上の CD にアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれる CD にアクセスできます。

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
```

デーモンが実行されていると、`/usr/lib/nfs/nfsd` の行は、上記のとおりに表示されます。デーモンが実行されていないと、`grep nfsd` の行だけが表示されます。

3. 以下のうちの該当する手順に進みます。

条件	次の手順
nfsd が実行されている場合	199ページの手順 8
nfsd が実行されていない場合	198ページの手順 4

4. nfsd がエクスポートするダミー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。このディレクトリにはファイルは含まれない。この目的は、NFS デーモンを「呼び起こして」、エクスポートされた CD-ROM を認識させることにある。

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、このエントリを参照して、エクスポートされたフロッピーディスクドライブを認識します。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 **CD** を取り出します。

```
# eject cdrom0
```

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 cdrom*
```

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で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. CD をロードします。

ここでロードする CD は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。

CD にアクセスするために、リモートユーザーは、196ページの「他のシステム上の CD にアクセスする方法」の指示に従って、名前によりその CD をマウントする必要があります。

13. CD が実際に他のシステムで使用できるかどうかを確認するには、share コマンドを使用してください。

CD が使用可能な場合は、その共有の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"
- /Sol_7_sparc ro ""
```

例 —ローカル CD を他のシステムで使用可能にする

次の例では、ローカルシステムの CD-ROM ドライブにロードされた CD を、ネットワーク上の他のシステムで使用できるようにしています。

```
# 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_7_sparc/s5   ro   ""
-           /cdrom/sol_7_sparc/s4   ro   ""
-           /cdrom/sol_7_sparc/s3   ro   ""
-           /cdrom/sol_7_sparc/s2   ro   ""
-           /cdrom/sol_7_sparc/s1   ro   ""
-           /cdrom/sol_7_sparc/s0   ro   ""
#

```

▼ 音楽 CD を演奏するようにシステムを設定する方法

Solaris システムに接続された CD-ROM から音楽 CD を演奏することができます。音楽 CD を演奏するには、パブリックドメインソフトウェアである Workman にアクセスし、外部スピーカまたはヘッドホンを CD-ROM ドライブに接続する必要があります。システムハードウェアに接続されたスピーカは動作しません。

システムを設定したら、音楽 CD を CD-ROM ドライブに挿入するだけで、その CD を演奏することができます。Workman のコントロールパネルは、デスクトップに自動的に表示されます。

1. スーパーユーザーになります。

2. /etc/rmmount.conf を編集します。

以下の例に示すように、# Actions の下、cdrom 記述の前に次の行を追加します。

```

# Actions
action cdrom action_workman.so path/workman workman-options

```

<i>path</i>	Workman ソフトウェアを置いたディレクトリ
<i>workman-options</i>	Workman ソフトウェアによって許可されたオプション

例 — 音楽 CD を演奏するようにシステムを設定する

Workman ソフトウェアをサポートするように変更された `/etc/rmmount.conf` ファイルの例を示します。

```
# @(#)rmmount.conf 1.3      96/05/10 SMI
#
# Removable Media Mounter configuration file.
#
# File system identification
ident hsfs ident_hsfs.so cdrom
ident ufs ident_ufs.so cdrom floppy rmscsi pcmem
ident pcfs ident_pcfs.so floppy rmscsi pcmem

# Actions
action cdrom action_workman.so /usr/dist/exe/workman
action cdrom action_filemgr.so
action floppy action_filemgr.so
action rmscsi action_filemgr.so

# File System Sharing
share cdrom*
share floppy*
```

▼ 新しい CD-ROM ドライブ用にシステムを準備する方法

ボリューム管理が新しい CD-ROM ドライブを認識するようにするには、`/reconfigure` ファイルを作成し、システムをリブートする必要があります。

1. スーパーユーザーになります。
2. `/reconfigure` というファイルを作成します。

```
# touch /reconfigure
```

3. システムをリブートします。

ボリューム管理の設定

通常はボリューム管理を使用しないで、フロッピーディスクまたは CD を管理したいことがあります。ここでは、ボリューム管理を終了してその後再起動する方法を説明します。

▼ ボリューム管理を終了させる方法

1. フロッピーディスクまたは **CD** が使用中でないことを確認します。
フロッピーディスクまたは CD を使用中のすべてのユーザーを確認したかどうかを調べるには、195ページの「CD が使用中かどうかを調べる方法」に説明されている `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.
```

コマンド行でのフロッピーディスクのフォーマットと使用方法 (手順)

この章では、Solaris 環境において、コマンド行からフロッピーディスクをフォーマットして使用方法を説明します。この章で説明する手順は次のとおりです。

- 210ページの「UFS フロッピーディスクをフォーマットする方法」
- 213ページの「UFS ファイルシステムをフロッピーディスク上に作成する方法」
- 215ページの「DOS フロッピーディスクをフォーマットする方法」
- 218ページの「フロッピーディスクをロードする方法」
- 221ページの「フロッピーディスクの内容を調べる方法」
- 221ページの「フロッピーディスクから情報をコピーまたは移動する方法」
- 223ページの「フロッピーディスクへ情報をコピーまたは移動する方法」
- 224ページの「フロッピーディスクが使用中かどうかを調べる方法」
- 225ページの「フロッピーディスクを取り出す方法」
- 226ページの「他のシステム上のフロッピーディスクにアクセスする方法」
- 228ページの「ローカルフロッピーディスクを他のシステムで使用可能にする方法」

フロッピーディスクのフォーマット

表 16-1 作業マップ: フロッピーディスクのフォーマット方法

作業	説明	手順の説明
フォーマットされていないフロッピーディスクのロード	フロッピーディスクをドライブに挿入して、 <code>volcheck</code> コマンドを実行する。	218ページの「フロッピーディスクをロードする方法」
フロッピーディスクのフォーマット	UFS 用にフロッピーディスクをフォーマットする。 DOS 用にフロッピーディスクをフォーマットする。	210ページの「UFS フロッピーディスクをフォーマットする方法」 215ページの「DOS フロッピーディスクをフォーマットする方法」
UFS ファイルシステムの作成	<i>palatino</i> の場合のみ。 省略可能。フロッピーディスクをファイルの保存に使用する場合は、UFS ファイルシステムを作成する。文字の保存に使用する場合に、この手順は不要。	213ページの「UFS ファイルシステムをフロッピーディスク上に作成する方法」
フロッピーディスクの取り出し	フォーマットが終了したら、すぐに使用する場合でも、必ずそのフロッピーディスクを取り出す。	225ページの「フロッピーディスクを取り出す方法」

フロッピーディスクの名前について

フロッピーディスクを操作する場合は、名前、または次の表 16-2 の代替名によって識別できます。説明を簡単にするために、このマニュアル中の説明では `floppy0` を使用しますが、実際の作業ではフロッピーディスク名または別の代替名も使用できます。

表 16-2 フロッピーディスクを識別する方法

フロッピーディスク	フロッピーディスクの代替名
最初のフロッピーディスクドライブ	floppy0
2 番目のフロッピーディスクドライブ	floppy1
3 番目のフロッピーディスクドライブ	floppy2

注 - 名前の付いていない (つまり「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。

ハードウェアの考慮点

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。次の表はこの制限を要約したものです。

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
IA 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、IA システム上の UFS には使用できません。IA システム上でフォーマットされたフロッピーディスクの場合も同様です。これは、SPARC と IA 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、IA 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 のいずれ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。205ページの「フロッピーディスクのフォーマット」を参照してください。

Solaris システム (SPARC または IA のいずれか) では、7つの異なる密度のフロッピーディスクをフォーマットできます (適切なドライブが必要です)。

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"	拡張密度	2.88M バイト
3.5"	高密度 (HD)	1.44M バイト
3.5"	中密度 (DD)	1.2M バイト
3.5"	低密度	720M バイト
5.25"	高密度 (HD)	1.2M バイト
5.25"	中密度 (DD)	720K バイト
5.25"	低密度	360K 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、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ます。1.44M バイトドライブに、フロッピーディスクを、たとえば 720K バイトにフォーマットするように指示できます。しかし、720K バイトドライブに、フロッピーディスクを 1.44M バイト用にフォーマットするよう指示することはできません。つまり、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。

ドライブに対して、フロッピーディスクをデフォルト以外の密度にフォーマットするよう指示するには、次の作業に示すように、`fdformat` コマンドを使用してください。この場合、次の表 16-3 から適切な密度オプションを使用するようにしてください。

表 16-3 密度のオプション

フロッピーディスクの フォーマット密度	ドライブがデフォルト密 度を使用する場合	fdformat コマンドに使用する密 度オプション
2.88M バイト	2.88M バイト	-E
1.44M バイト	2.88M バイト	-H
1.44M バイト	1.44M バイト	なし
1.2M バイト	1.44M バイト	-t nec -M
720K バイト	1.44M バイト	-D または -t dos -D
1.2M バイト	1.2M バイト	なし
720K バイト	1.2M バイト	-D
720K バイト	720K バイト	なし
360 バイト	720K バイト	-D

fdformat コマンドのオプションについては、マニュアルページを参照するか、fdformat -z を入力してください。この -z オプションを使用すると、コマンドのすべてのオプションが表示されます。

ドライブのデフォルト密度がわからない場合は、デフォルト設定 (つまり、密度オプションのない状態) によってフォーマット処理を開始して、設定メッセージに注意してください。このメッセージは、次のように表示されます。

```
Formatting 1.44 M in /vol/dev/rdiskette0/unformatted
Press return to start formatting floppy.
```

確認メッセージには、ドライブのデフォルト密度が示されます。たとえば、上の例では、ドライブのデフォルト密度は 1.44M バイトです。密度が目的のものでない場合は、Ctrl-c キーを使用してフォーマット処理を中止し、フォーマットをやりなおしてください。

▼ UFS フロッピーディスクをフォーマットする方法

最初の説明にあるように、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、SPARC システムでしか使用できません。また、IA 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、IA システムでしか使用できません。



注意 - フロッピーディスクをフォーマットすると、既存の内容はすべて消去されます。

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていないフロッピーディスクを挿入すると、フォーマット用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ておきたい場合は、フォーマットウィンドウが表示されたときに、そのフォーマットウィンドウを終了してください。

2. フロッピーディスクが書き込み可能になっていることを確認します。

3.5 インチと 5.25 インチのどちらのフロッピーディスクでも、書き込み保護は、左下または右下の角にある小さなタブによって設定されます。タブの背後に四角い穴が見える場合、フロッピーディスクは書き込み保護になっています。この穴がタブによって覆われている場合、フロッピーディスクは書き込み可能です。(フロッピーディスクを取り出して調べる必要がある場合は、`eject floppy` とシェルに入力してください。)

3. フロッピーディスクを挿入します。

フロッピーディスクが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

```
$ fdformat -v -U [density-options convenience-options]
```

`-v` フロッピーディスクが正しくフォーマットされたかどうかを確認する。

`-U` フロッピーディスクがマウントされている場合は、それを解除する。

`density-options` ドライブ密度が 1.44M バイトの場合、密度オプションは次のようになる。

- なし— 1.44M バイトにフォーマットする。
 - D 720K バイトにフォーマットする。
- 密度オプションについては、表 16-3 を参照。

convenience-options

- e フォーマットが終了すると、フロッピーディスクを取り出す。
- f フォーマットの前に確認を要求しない。
- b *label* フロッピーディスクに名前を付ける。名前は、8 文字以下にする必要がある。大文字小文字は区別されない。
- z `fdformat` コマンドのすべてのオプションを一覧表示するが、フロッピーディスクはフォーマットしない。

注 - 720K バイト (DD) 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たときに、`-v` オプションが指定されていない場合は `fdformat` の処理は中止されません。`-v` オプションがされている場合には、`fdformat` はフロッピーディスクをフォーマットしますが、検査でエラーが検出されて次のメッセージが出力されます。`fdformat: check diskette density, I/O error`

`fdformat` コマンド (`-f` オプションを使用していない場合) は、確認メッセージを表示して、実行されるフォーマットのタイプを示します。

```
Formatting 1.44 M in /vol/dev/rdiskette0/unformatted
Press return to start formatting floppy.
```

5. 目的に応じて、以下のいずれかのキーを使用します。

目的	使用するキー
フォーマットのタイプの確認	Return キー (前の手順で <code>-f</code> オプションを使用していない場合。使用した場合は、確認は不要。)
フォーマットの取り消し	Ctrl-c キー

フォーマット処理中には、ドットが連続して表示されます。検査処理中には、ドットの後に V が連続して表示されます。表示が停止すると、フォーマットは完了です。

これでフロッピーディスクを、tar や cpio などの raw データ操作に使用することができます。

例—UFS フロッピーディスクをフォーマットする

次に、UFS フォーマットのいくつかの例を示します。最初の例では、1.44M バイトのフロッピーディスクを 1.44M バイトドライブでフォーマットしています。

```
$ fdformat -v -U
Formatting 1.44 M in /vol/dev/rdiskette0/unformatted
Press return to start formatting floppy. [ Return キー ]
.....
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

次の例では、同じジョブを実行していますが、フロッピーディスクに `myfiles` という名前を割り当てています。

```
$ fdformat -v -U -b myfiles
Formatting 1.44 M in /vol/dev/rdiskette0/unformatted
Press return to start formatting floppy. [ Return キー ]
.....
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

次の例では、720K バイトフロッピーディスクを、1.44M バイトドライブでフォーマットして、それに `myfiles` という名前を付けています。


```
newfs: construct a new file system
/vol/dev/aliases/floppy0: (y/n)? y
```

状態メッセージが表示されて、ファイルシステムとフロッピーディスクのフォーマットの詳細を示します。

これで、フロッピーディスクを SPARC システムで使用することができます。ただし、ボリューム管理にフロッピーディスクを認識させるには、次の手順で説明するように、`volrmmount` コマンドを実行する必要があります。

4. フロッピーディスクが挿入されたことをボリューム管理に通知するために、`-i` オプションを指定して `volrmmount` コマンドを呼び出します。

```
$ volrmmount -i floppy0
```

5. `/floppy` ディレクトリで `ls` コマンドを使用して、**UFS** ファイルシステムがフロッピーディスク上に作成されていることを確認します。

`floppy0` サブディレクトリが表示されれば、UFS ファイルシステムがフロッピーディスクに保存され、正しくマウントされたということです。

```
$ ls /floppy
floppy0
```

例 — UFS ファイルシステムをフロッピーディスクに作成する

```
$ volcheck -v
media was found
$ /usr/sbin/newfs -v /vol/dev/aliases/floppy0
newfs: construct a new file system /vol/dev/aliases/floppy0: (y/n)? y
mkfs -F ufs /vol/dev/aliases/floppy0 2880 18 2 8192 1024 16 10 5 2048
t 0 -1 8 15
/vol/dev/aliases/floppy0: 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,
$ volrmmount -i floppy0
$ ls /floppy
```

(続く)

```
floppy0
```

▼ DOS フロッピーディスクをフォーマットする方法

SPARC システムおよび IA システムのどちらでも、DOS フロッピーディスクをフォーマットできます。両者の手順は同じですが、SPARC の場合はフロッピーディスク上に SunOS ファイルシステムが作成されるのに対して、IA の場合はファイルシステム上に DOS ファイルシステム (MS-DOS または NEC-DOS) が作成されます。



注意 - フロッピーディスクをフォーマットすると、既存の内容はすべて消去されます。

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていないフロッピーディスクを挿入すると、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たから、そのフォーマットウィンドウを終了してください。

2. フロッピーディスクが書き込み保護になっていないことを確認します。

3.5 インチと 5.25 インチのどちらのフロッピーディスクでも、書き込み保護は、左下または右下の角にある小さなタブによって設定されます。タブの背後に四角い穴が見える場合、フロッピーディスクは書き込み保護になっています。この穴がタブで覆われている場合、フロッピーディスクは書き込み可能です。(フロッピーディスクを取り出して調べる必要がある場合は、`eject floppy` とシェルに入力してください。)

3. フロッピーディスクを挿入します。

フロッピーディスクが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

```
$ fdformat -v -U [density-options convenience-options]
```

-v	フロッピーディスクが正しくフォーマットされたかどうかを確認する。
-U	フロッピーディスクがマウントされている場合は、それを解除する。
<i>density-options</i>	ドライブ密度が 1.44M バイトの場合、密度オプションは次のようになる。
-d	1.44M バイトで MS-DOS 用にフォーマットする。
-d -D	720K バイトで MS-DOS 用にフォーマットする。
-t nec -M	1.2M バイトで NEC-DOS 用にフォーマットする。
	密度オプションについては、表 16-3 を参照。
<i>convenience-options</i>	
-e	フォーマットが終了すると、フロッピーディスクを取り出す。
-f	フォーマットの前に確認を要求しない。
-b <i>label</i>	フロッピーディスクに名前を付ける。名前は、8 文字以下にする必要がある。大文字小文字は区別されない。
-z	fdformat コマンドのすべてのオプションを一覧表示するが、フロッピーディスクはフォーマットしない。

注 - 720K バイト (DD) 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た場合、fdformat は、-v オプションが指定されないかぎり、処理を中止しない。-v オプションを使用すると、fdformat は、フロッピーディスクをフォーマットするが、検査でエラーが検出されて、次のメッセージによって通知される。fdformat: check diskette density, I/O error

fdformat コマンドは、確認メッセージを表示して、実行されるフォーマットのタイプを示します。

フロッピーディスクの使用方法

表 16-4 作業マップ: 一般的なフロッピーディスクの使用方法

作業	説明	手順の説明
フロッピーディスクのロード	フロッピーディスクをドライブに挿入して、 <code>volcheck</code> コマンドを実行する。	218ページの「フロッピーディスクをロードする方法」
フロッピーディスクの内容の確認	省略可能。フロッピーディスクの内容を調べるには、 <code>/diskette</code> の下にある適切なディレクトリを調べる。	221ページの「フロッピーディスクの内容を調べる方法」
ファイルの交換	省略可能。フロッピーディスクとファイルシステムの間でファイルまたはディレクトリをコピーする。	221ページの「フロッピーディスクから情報をコピーまたは移動する方法」 223ページの「フロッピーディスクへ情報をコピーまたは移動する方法」
フロッピーディスクが使用中かどうかの確認	省略可能。フロッピーディスクを取り出す前に、フロッピーディスクが使用中かどうかを調べる。	224ページの「フロッピーディスクが使用中かどうかを調べる方法」
フロッピーディスクの取り出し	終了したら、フロッピーディスクをドライブから取り出す。	225ページの「フロッピーディスクを取り出す方法」

▼ フロッピーディスクをロードする方法

1. フロッピーディスク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不確かな場合は、フロッピーディスクを挿入して、219ページの手順3で説明されているように、コンソールの状態メッセージをチェックしてください。フロッピーディスクをフォーマットする必要がある場合は、210ページの「UFS フロッピーディスクをフォーマットする方法」、ま

または 215ページの「DOS フロッピーディスクをフォーマットする方法」を参照してください。

2. フロッピーディスクを挿入します。

フロッピーディスクがドライブに完全に挿入されたことを確認してください。ドライブにふたがある場合は、それを閉じてください。

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どちらかが表示されます。

media was found

ボリューム管理がフロッピーディスクを検出して、それを /floppy ディレクトリにマウントしようとする。

フロッピーディスクが正しくフォーマットされると、エラーメッセージはコンソールに表示されない。

フロッピーディスク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、次のエラーメッセージがコンソールに表示される。

```
fd0: unformatted diskette or no diskette in
the drive
```

```
fd0: read failed (40 1 0)
```

```
fd0: bad format
```

フロッピーディスク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。フォーマットの方法は、210ページの「UFS フロッピーディスクをフォーマットする方法」(UFS 用)と215ページの「DOS フロッピーディスクをフォーマットする方法」(DOS 用)を参照。

no media was found

ボリューム管理は、フロッピーディスクを検出しなかった。フロッピーディスク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、フロッピーディスクをチェックする。損傷の可能性がある。フロッピーディスクを手作業でマウントしてみることもできる。

4. フロッピーディスクの内容を一覧表示して、フロッピーディスクがマウントされていることを確認します。

```
$ ls /floppy
floppy0 myfiles
```

前述したように、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。この場合は 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、または正しくフォーマットされていないかのいずれか

です。これを調べるには、mount コマンドを実行して、/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。コンソールのエラーメッセージを確認してください。

▼ フロッピーディスクの内容を調べる方法

/floppy の下にあるディレクトリの一部がシンボリックリンクであるため、ls コマンドに -L オプションを付けて使用してください。

```
$ ls -L [-l] floppy0
```

-L 出力にシンボリックリンクを含める。

-l 詳細表示。アクセス権と所有権を出力に含める。

例 — フロッピーディスクの内容を調べる

次の例では、floppy0 で識別される最初のフロッピードライブのフロッピーディスクの内容を一覧表示しています。

```
$ ls -L -l /floppy/floppy0
-rwxrwxrwx 1 smith staff 362284 Nov 16 20:54 text.doc
-rwxrwxrwx 1 smith staff 24562 Nov 16 12:20 art.gif
```

▼ フロッピーディスクから情報をコピーまたは移動する方法

フロッピーディスクを挿入すると、他のファイルシステムの場合と同様に、そのファイルとディレクトリにアクセスできます。ただし、所有権とアクセス権については注意が必要です。たとえば、フロッピーディスクのファイルの所有者ではない場合、フロッピーディスク上でそれを変更することはできません。また、ファイルシステムにファイルをコピーした場合、そのファイルの所有者にはなりますが、書き込み権はありません (フロッピーディスクでも書き込み権がなかったためです)。アクセス権は各自で変更する必要があります。

1. フロッピーディスクがフォーマットされてマウントされていることを確認します。

```
$ ls /floppy
floppy0  diskette-name
```

フロッピーディスクが正しくフォーマットされてマウントされていれば、その名前とシンボリックリンクが、/floppy の下に表示されます。

/floppy ディレクトリに何も表示されない場合、フロッピーディスクはマウントされていません。218ページの「フロッピーディスクをロードする方法」を参照してください。フロッピーディスクがフォーマットされていない場合もあります。210ページの「UFS フロッピーディスクをフォーマットする方法」または215ページの「DOS フロッピーディスクをフォーマットする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

コピーするもの	使用するコマンド
ファイル	cp
ディレクトリ	cp -r

3. ls コマンドを使用して、コピーまたは移動の結果を確認します。

例 — フロッピーディスクから情報をコピーまたは移動する

最初の例では、ファイル (readme.doc) をフロッピーディスクから現在のディレクトリ (「.」で示される) に移動しています。2番目の例では、ファイル (readme2.doc) をフロッピーディスクから現在のディレクトリにコピーしています。3番目の例では、ディレクトリ (morefiles) とその下にあるすべてのものを、フロッピーディスクから現在のディレクトリにコピーしています。

```
$ mv /floppy/floppy0/readme.doc .
$ cp /floppy/floppy0/readme2.doc .
$ cp -r /floppy/floppy0/morefiles .
```

▼ フロッピーディスクへ情報をコピーまたは移動する方法

1. フロッピーディスクが書き込み保護になっていないことを確認します。
3.5 インチと 5.25 インチのどちらのフロッピーディスクでも、書き込み保護は、左下または右下の角にある小さなタブによって設定されます。タブの背後に四角い穴が見える場合、フロッピーディスクは書き込み保護されています。この穴がタブで覆われている場合、フロッピーディスクは書き込み可能です。
2. フロッピーディスクがフォーマットされてマウントされていることを確認します。

```
$ ls /floppy
floppy0  diskette-name
```

フロッピーディスクが正しくフォーマットされてマウントされていれば、その名前とシンボリックリンクの `floppy0` が、`/floppy` の下に表示されます。

`/floppy` ディレクトリに何も表示されない場合、フロッピーディスクはマウントされていません。218ページの「フロッピーディスクをロードする方法」に示された作業を参照してください。フロッピーディスクがフォーマットされていない場合もあります。210ページの「UFS フロッピーディスクをフォーマットする方法」、または 215ページの「DOS フロッピーディスクをフォーマットする方法」参照してください。

3. ファイルまたはディレクトリを移動またはコピーします。

作業	使用するコマンド
ファイルのコピー	cp
ディレクトリのコピー	cp -r
ファイルまたはディレクトリの移動	mv

4. ls コマンドを使用して、移動またはコピーの結果を確認します。

例 — 情報をフロッピーディスクにコピーまたは移動する

最初の例では、ファイル (readme.doc) を現在のディレクトリから、最初のフロッピードライブ (/floppy/floppy0) にロードされたフロッピーディスクに移動しています。2 番目の例では、ファイル (readme2.doc) を現在のディレクトリから、2 番目のフロッピードライブ (/floppy/floppy1) にロードされたフロッピーディスクにコピーしています。3 番目の例では、ディレクトリ (morefiles) とその下にあるすべてのものを、/home/smith/ ディレクトリから最初のフロッピードライブにコピーしています。

```
$ mv readme.doc /floppy/floppy0
$ cp readme2.doc /floppy/floppy1
$ cp -r /home/smith/morefiles /floppy/floppy0
```

▼ フロッピーディスク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. fuser コマンドを起動します。

fuser コマンドは、指定された CD に現在アクセスしているプロセスを表示します。

```
# fuser -u [-k] floppy0
```


- u フロッピーディスクを使用中のユーザーを表示する。
- k フロッピーディスクにアクセスしているプロセスを終了させる。

例 — フロッピーディスクが使用中かどうかを調べる

次の例では、プロセス 6400c と 6399c が /floppy/floppy0 ディレクトリにアクセスしていて、プロセス所有者はそれぞれ root と smith です。

```
# fuser -u /floppy/floppy0  
/floppy/floppy0: 6400c(root) 6399c(smith)
```

プロセスは、root 権限によって個別に終了させることも、fuser コマンドに -k オプションを付けて使用して、ファイルシステムにアクセス中のすべてのプロセスを終了させることもできます。fuser コマンドは、終了したプロセスの一部を識別できないことがあります。確認するには、-u オプションを付けてもう一度このコマンドを実行してください。

```
# fuser -u -k /floppy/floppy0  
/floppy/floppy0: 6400c(root)Killed 6399c(smith)Killed
```

▼ フロッピーディスクを取り出す方法

1. フロッピーディスク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フロッピーディスク中のファイルやディレクトリにアクセスしている場合、フロッピーディスクは「使用中」であることを忘れないでください。

フロッピーディスクのすべてのユーザーを検出したかどうか不確かな場合は(デスクトップツールの背後に隠れたシェルでアクセスしている可能性がある場合は)、224ページの「フロッピーディスクが使用中かどうかを調べる方法」に説明されている fuser コマンドを使用してください。

2. フロッピーディスクを取り出します。

```
# eject floppy0
```

SPARC システムでは、フロッピーはドライブから物理的に取り出されますが、IA システムでは手でフロッピーディスクを取り出す必要があります。ウィンドウを実行している場合は、フロッピーディスクを取り出すように指示する画面メッセージに注意してください。

フロッピーディスクが使用中の場合は、次のメッセージが表示されます。

```
/vol/dev/rdiskette0/noname: Device busy
```

この場合は、225ページの手順 1 に戻って、誰もフロッピーディスクを使用していないことを確認してから、もう一度それを取り出してください。

フロッピーディスクが詰まった場合は、紙クリップを伸ばした先端を、ドライブ前面の小さな穴に 3 センチほど挿入し、手で取り出してください。

▼ 他のシステム上のフロッピーディスクにアクセスする方法

他のシステム上のフロッピーディスクを各自のファイルシステムに手作業でマウントすることによって、そのフロッピーディスクにアクセスできます。ただしこれは、他のシステムが、228ページの「ローカルフロッピーディスクを他のシステムで使用可能にする方法」の指示に従ってフロッピーディスクドライブをエクスポートしている場合にかぎります。

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、あるいは作成します。

```
$ mkdir directory
```

directory

他のシステムのフロッピーディスク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

2. マウントしたいフロッピーディスクの名前を検索します。

手作業でリモートフロッピーディスクをマウントする場合は、ローカルフロッピーディスクで使用できる `floppy0` または `floppy1` 変数を使用することができません。フロッピーディスクの名前を使用する必要があります。この名前を検索するには、リモートシステムの `/floppy` ディレクトリで `ls` コマンドを使用してください。オートマウンタが実行されている場合は、マウントしたいフロッ

ピーディスクのシステムに `cd` コマンドで移動してから、`ls` コマンドを使用できます。オートマウンタが実行されていない場合は、リモートからログインするなどの別の方法を使用する必要があります。

3. スーパーユーザーとして、フロッピーディスクをマウントします。

```
# mount -F nfs -o rw system-name:/floppy/diskette-name local-mount-point
```

<i>system-name</i>	マウントするフロッピーディスクが存在するシステムの名前
<i>diskette-name</i>	マウントしたいフロッピーディスクの名前
<i>local-mount-point</i>	リモートフロッピーディスク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. フロッピーディスクが実際にマウントされたかどうかを確認するには、`ls` コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /floppy
```

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

次の例では、`myfiles` という名前の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
```

▼ ローカルフロッピーディスクを他のシステムで使用可能にする方法

システムを設定して、そのフロッピーディスクをエクスポートすることができます。つまり、これらのドライブ上のフロッピーディスクを、他のシステムが使用できるようになります。フロッピーディスクドライブがエクスポートされると、他のシステムは、226ページの「他のシステム上のフロッピーディスクにアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれるフロッピーディスクにアクセスできます。

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. 以下の表から該当する手順を選択します。

条件	次の手順
nfsd が実行されている場合	229ページの手順 8
nfsd が実行されていない場合	228ページの手順 4

4. nfsd がエクスポートするダミー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」、エクスポートされたフロッピーディスクを認識させることを目的としています。

5. 次のエントリを `/etc/dfs/dfstab` 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、このエントリを参照して、エクスポートされたフロッピーディスクドライブを認識します。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. 現在ドライブ内にあるフロッピーディスクを取り出します。

```
# eject floppy0
```

9. `root` の書き込み権を `/etc/rmmount.conf` に割り当てます。

```
# chmod 644 /etc/rmmount.conf
```

10. 次の行を `/etc/rmmount.conf` に追加します。

```
# File System Sharing
share floppy*
```

上記の行によって、システムのフロッピーディスクドライブにロードされたすべてのフロッピーディスクが共有されます。

11. `/etc/rmmount.conf` 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. フロッピーディスクをロードします。

```
--- フロッピーディスクを挿入する ---  
# volcheck -v  
media was found
```

ここでロードするフロッピーディスクは他のシステムで使用できるようになります。フロッピーディスクにアクセスするために、リモートユーザーは、226ページの「他のシステム上のフロッピーディスクにアクセスする方法」の指示に従って、名前によりそのフロッピーディスクをマウントする必要があります。

13. フロッピーディスクが実際に他のシステムで使用できるかどうかを確認するには、`share` コマンドを使用してください。

フロッピーディスクが使用可能な場合は、その共有の設定が表示されます。(共有されるダミーディレクトリも表示されます。)

```
# share  
- /dummy ro "dummy dir to wake up NFS daemon"  
- /myfiles rw ""
```

例 — ローカルフロッピーディスクを他のシステムで使用可能にする

次の例では、ローカルシステムの CD-ROM ドライブにロードされたフロッピーディスクすべてを、ネットワーク上の他のシステムで使用できるようにしています。

```
# 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  ""
```


コマンド行での **PCMCIA** メモリーカードの 使用方法 (手順)

この章では、Solaris 環境においてコマンド行から PCMCIA メモリーカードをフォーマットして使用するために必要な作業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。

- 236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」
- 238ページの「UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法」
- 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」
- 244ページの「PCMCIA メモリーカードをロードする方法」
- 246ページの「PCMCIA メモリーカードの内容を調べる方法」
- 247ページの「PCMCIA メモリーカードから情報をコピーまたは移動する方法」
- 248ページの「PCMCIA メモリーカードへ情報をコピーまたは移動する方法」
- 249ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」
- 250ページの「PCMCIA メモリーカードを取り出す方法」
- 251ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」
- 253ページの「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方法」

PCMCIA メモリーカードのフォーマット

表 17-1 作業マップ: PCMCIA メモリーカードのフォーマット方法

作業	説明	手順の説明
フォーマットされていない PCMCIA メモリーカードのロード	IPCMCIA メモリーカードをドライブに挿入し、volcheck コマンドを実行する。	244ページの「PCMCIA メモリーカードをロードする方法」
PCMCIA メモリーカードのフォーマット	PCMCIA メモリーカードを、UFS 用にフォーマットする。 PCMCIA メモリーカードを、DOS 用にフォーマットする。	236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」
UFS ファイルシステムの作成	UFS の場合のみ。省略可能。PCMCIA メモリーカードをファイルの保存に使用する場合は、UFS ファイルシステムを作成する。文字の保存に使用する場合には、この手順は不要。	238ページの「UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法」
PCMCIA メモリーカードの取り出し	PCMCIA メモリーカードのフォーマットを終了したら、すぐに使用する場合でも、必ずそれを取り出す。	250ページの「PCMCIA メモリーカードを取り出す方法」

PCMCIA メモリーカードの名前について

PCMCIA メモリーカードを操作する場合は、名前、または次の表 17-2 に示す代替名によってそれらを識別できます。説明を簡単にするために、このマニュアル中の説明では pcmem0 を使用しますが、実際の作業では PCMCIA メモリーカード名または別の代替名も使用できます。

表 17-2 PCMCIA メモリーカードを識別する方法

PCMCIA カード	代替名
最初の PCMCIA ドライブ	pcmem0
2 番目の PCMCIA ドライブ	pcmem1
3 番目の PCMCIA ドライブ	pcmem2

注 - 名前の付いていない (つまり「ラベル」がない) PCATA ドライブには、noname というデフォルト名が割り当てられます。

ハードウェアの考慮点

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いくつかの制限があります。次の表はこの制限を要約したものです。

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
IA 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、IA システム上の UFS には使用できません。IA システム上でフォーマットされた PCMCIA メモリーカードの場合も同様です。これは、SPARC と IA 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 のいずれかのファイ

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。234ページの「PCMCIA メモリーカードのフォーマット」を参照してください。

`fdformat` コマンドのオプションについては、`fdformat (1)` を参照するかまたは `fdformat -z` と入力してください。この `-z` オプションを使用すると、`fdformat` コマンドのすべてのオプションが表示されます。

▼ UFS PCMCIA メモリーカードをフォーマットする方法

最初の説明にあるように、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、SPARC システムでしか使用できません。また、IA システムでフォーマットされた UFS PCMCIA メモリーカードは、Intel 版 Solaris を実行する IA システムでしか使用できません。



注意 - PCMCIA メモリーカードをフォーマットすると、既存の内容はすべて消去されます。

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていない PCMCIA メモリーカードを挿入すると、フォーマット用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ておきたい場合は、フォーマットウィンドウが表示されたときに、そのフォーマットウィンドウを終了してください。

2. PCMCIA メモリーカードが書き込み可能になっていることを確認します。

書き込み保護は、PCMCIA メモリーカードの端にある小さいスライドスイッチによって設定します。

3. PCMCIA メモリーカードを挿入します。

PCMCIA メモリーカードが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

```
$ fdformat -v -U [convenience-options]
```

-v PCMCIA メモリーカードが正しくフォーマットされたかどうかを確認する。

-U PCMCIA メモリーカードがマウントされている場合は、それを解除する。

convenience-options

-e フォーマットが終了すると、PCMCIA メモリーカードを取り出す。

-f フォーマットの前に確認を要求しない。

-b *label* PCMCIA メモリーカードに名前を付ける。名前は、8文字以下にする。大文字と小文字は区別されない。

-z fdformat コマンドのすべてのオプションを一覧表示するが、PCMCIA メモリーカードはフォーマットしない。

fdformat コマンドの `-f` オプションを使用していなければ、実行されるフォーマットのタイプと確認メッセージが表示されます。

```
Formatting in /vol/dev/aliases/pcm0  
Press return to start formatting pcm0.
```

5. 目的に応じて、以下のいずれかのキーを使用します。

目的	使用するキー
フォーマットのタイプの確認	Return キー (前の手順で -f オプションを使用していない場合。使用した場合は、確認は不要。)
フォーマットの取り消し	Ctrl-c キー

フォーマットの進行中は、ドットが連続して表示されます。検査の進行中は、ドットの後に V が連続して表示されます。表示が停止すると、フォーマットは完了です。

これで PCMCIA メモリーカードを、tar や cpio などの raw データ操作に使用できます。

例 — UFS PCMCIA メモリーカードをフォーマットする

次に、UFS フォーマットの例を示します。

```
$ fdformat -v -U
Formatting in /vol/dev/aliases/unformatted
Press return to start formatting pcmem0. [ Return キー ]
.....
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

次の例では、同じジョブを実行していますが、PCMCIA メモリーカードに myfiles という名前を割り当てています。

```
$ fdformat -v -U -b myfiles
Formatting in /vol/dev/aliases/unformatted
Press return to start formatting pcmem0. [ Return キー ]
.....
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

▼ UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法

UFS ファイルシステムを作成する手順は、IA システムおよび SPARC システムのどちらでフォーマットされた UFS PCMCIA メモリーカードでも同じですが、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、SPARC システム

でしか使用できません。また、IA システムでフォーマットされた UFS PCMCIA メモリーカードは、Solaris を実行する IA システムでしか使用できません。

1. **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します。
236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」で説明した手順に従ってください。
2. `newfs` コマンドと、ボリューム管理ディレクトリへの絶対パス名を使用して、**PCMCIA** メモリーカード上に **UFS** ファイルシステムを作成します。

```
$ /usr/sbin/newfs -v /vol/dev/aliases/pcm0
```

`-v` 状態メッセージを出力する。

`/vol/dev/aliases/pcm0` PCMCIA メモリーカードの位置を示す。

`newfs` コマンドは、ファイルシステムの作成を確認するメッセージを表示しません。

3. ファイルシステムの作成を確認します。

```
newfs: construct a new file system \
/vol/dev/aliases/pcm0: (y/n)? y
```

状態メッセージが表示されて、ファイルシステムと PCMCIA メモリーカードのフォーマットの詳細を示します。

```
mkfs -F ufs /vol/dev/aliases/pcm0 2848 8 2 8192 1024 16 \
10 60 2048 t 0 -1 8 -1
/vol/dev/aliases/pcm0: 2848 sectors in 128 cylinders of \
2 tracks, 8 sectors
1.0MB in 8 cyl groups (16 c/g, 0.12MB/g, 64 i/g)
```

(続く)

```
super-block backups (for fsck -F ufs -o b=#) at:
    32, 304, 544, 816, 1056, 1328, 1568, 1840
```

これで、PCMCIA メモリーカードを SPARC システムで使用できます。ただし、ボリューム管理が PCMCIA メモリーカードを認識する前に、次の手順で説明するように、`volrmmount` コマンドを使用する必要があります。

4. メモリーカードが挿入されたことをボリューム管理に通知するために、`-i` オプションを指定して `volrmmount` コマンドを使用します。

```
$ volrmmount -i pcmem0
```

PCMCIA メモリーカードは、これで `/pcmem/pcmem0` にマウントされます。

5. **UFS** ファイルシステムが **PCMCIA** メモリーカード上に作成されたかどうかを確認するには、`/pcmem` ディレクトリで `ls` コマンドを使用します。

`pcmem0` サブディレクトリが表示されれば、UFS ファイルシステムが PCMCIA メモリーカードに作成され、正しくマウントされたことになります。

```
$ ls /pcmem
pcmem0
```

例 — UFS ファイルシステムを PCMCIA メモリーカード上に作成する

```
$ volcheck -v
media was found
$ /usr/sbin/newfs -v /vol/dev/aliases/pcmem0
newfs: construct a new file system \

    /vol/dev/aliases/pcmem0: (y/n)? y
mkfs -F ufs /vol/dev/aliases/pcmem0 ...

$ volrmmount -i pcmem0
```



```
media was found
```

▼ DOS PCMCIA メモリーカードをフォーマットする方法

SPARC システムおよび IA システムのどちらでも、DOS PCMCIA メモリーカードをフォーマットできます。両者の手順は同じですが、SPARC の場合は PCMCIA メモリーカード上に SunOS ファイルシステムが作成されるのに対して、IA の場合はファイルシステム上に DOS ファイルシステム (MS-DOS または NEC-DOS) が作成されます。



注意 - PCMCIA メモリーカードをフォーマットすると、既存の内容はすべて消去されます。

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていない PCMCIA メモリーカードを挿入すると、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。

2. **PCMCIA** メモリーカードが書き込み保護されていないことを確認します。

書き込み保護は、PCMCIA メモリーカードの端にある小さいスライドスイッチによって設定します。

3. **PCMCIA** メモリーカードを挿入します。

PCMCIA メモリーカードが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

```
$ fdformat -v -U [density-options convenience-options]
```

-v	PCMCIA メモリーカードが正しくフォーマットされたかどうかを確認する。
-U	PCMCIA メモリーカードがマウントされている場合は、それを解除する。
<i>density-options</i>	ドライブ密度が 1.44M バイトの場合、 <i>density-options</i> は次のとおり。
-d	MS-DOS 用にフォーマットする。
-t nec -M	1.2M バイトで NEC-DOS 用にフォーマットする。
	密度オプションについては、 <code>fdformat(1)</code> のマニュアルページを参照。
<i>convenience-options</i>	
-e	フォーマットが終了すると、PCMCIA メモリーカードを取り出す。
-f	フォーマットの前に確認を要求しない。
-b label	PCMCIA メモリーカードに名前を付ける。名前は、8 文字以下にする。大文字と小文字は区別されない。
-z	<code>fdformat</code> コマンドのすべてのオプションを一覧表示するが、PCMCIA メモリーカードはフォーマットしない。

注 - 720K バイト (DD) 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たときに、`-v` オプションが指定されていない場合は `fdformat` の処理は中止されません。`-v` オプションが指定されている場合には、`fdformat` はフロッピーディスクをフォーマットしますが、検査でエラーが検出されて次のメッセージ出力されます。`fdformat: check diskette density, I/O error`

実行されるフォーマットのタイプと、確認メッセージが表示されます。

```
Formatting 1.44 M in /vol/dev/rdiskette0/unformatted
Press return to start formatting floppy.
```

5. 目的に応じて、以下のいずれかのキーを使用します。

目的	使用するキー
フォーマットのタイプの確認	Return キー (前の手順で <code>-f</code> オプションを使用していない場合。使用した場合は、確認は不要。)
フォーマットの取り消し	Ctrl-c キー

フォーマット処理中には、ドットが連続して表示されます。検査処理中には、ドットの下に V が連続して表示されます。表示が停止すると、フォーマットは完了したことになります、DOS システム用に使用することができます。

6. メモリーカードが挿入されたことをボリューム管理に通知するために、`-i` オプションを指定して `volrmmount` コマンドを使用します。

```
$ volrmmount -i pcmem0
```

ボリューム管理は、PCMCIA メモリーカードを `/pcmem/pcmem0` の下にマウントします。

PCMCIA メモリーカードの使用方法

表 17-3 作業マップ: PCMCIA メモリーカードの使用方法

作業	説明	手順の説明
PCMCIA メモリーカードのロード	PCMCIA メモリーカードをドライブに挿入して、 <code>volcheck</code> コマンドを実行する。	244ページの「PCMCIA メモリーカードをロードする方法」
PCMCIA メモリーカードの内容の確認	省略可能。PCMCIA メモリーカードの内容を調べるには、 <code>/PCMCIAmemorycard</code> の下にある適切なディレクトリを調べる。	246ページの「PCMCIA メモリーカードの内容を調べる方法」

表 17-3 作業マップ: PCMCIA メモリーカードの使用法 続く

作業	説明	手順の説明
ファイルの交換	省略可能。PCMCIA メモリーカードとファイルシステムの間でファイルまたはディレクトリをコピーする。	247ページの「PCMCIA メモリーカードから情報をコピーまたは移動する方法」 248ページの「PCMCIA メモリーカードへ情報をコピーまたは移動する方法」
PCMCIA メモリーカードが使用中かどうかの確認	省略可能。PCMCIA メモリーカードを取り出す前に、PCMCIA メモリーカードが使用中であるかどうかを調べる。	249ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」
PCMCIA メモリーカードの取り出し	終了したら、PCMCIA メモリーカードを取り出す。	250ページの「PCMCIA メモリーカードを取り出す方法」

▼ PCMCIA メモリーカードをロードする方法

1. PCMCIA メモリーカード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不確かな場合は、PCMCIA メモリーカードを挿入して、以下の243ページの「PCMCIA メモリーカードの使用法」で説明されているように、コンソールの状態メッセージをチェックしてください。

PCMCIA メモリーカードをフォーマットする必要がある場合は、236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

2. PCMCIA メモリーカードを挿入します。

PCMCIA メモリーカードがドライブに完全に挿入されたことを確認してください。ドライブにふたがある場合は、それを閉じてください。

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

ボリューム管理が PCMCIA メモリーカードを検出して、それを /pcmem ディレクトリにマウントしようとする。

PCMCIA メモリーカードが正しくフォーマットされていると、エラーメッセージはコンソールに表示されない。

PCMCIA メモリーカード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、次のエラーメッセージがコンソールに表示される。

```
fd0: unformatted diskette or no diskette in
the drive
```

```
fd0: read failed (40 1 0)
```

```
fd0: bad format
```

PCMCIA メモリーカード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。フォーマットの方法は、236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」(UFS 用)と 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」(DOS 用)を参照。

no media was found

ボリューム管理は、PCMCIA メモリーカードを検出しなかった。PCMCIA メモリーカード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、PCMCIA メモリーカードをチェックする。損傷の可能性がある。PCMCIA メモリーカードを手作業でマウントしてみることもできる。

4. 内容をリスト表示して、**PCMCIA** メモリーカードがマウントされていることを確認します。

```
$ ls /pcmem/pcmem0
pcmem0 myfiles
```

前述したように、`pcmem0` は PCMCIA メモリーカードの実際の名前へのシンボリックリンクです。この場合は `myfiles` が実際の名前です。PCMCIA メモリーカードに名前はないけれども正しくフォーマットされている場合は、`unnamed_floppy` と示されます。

`/pcmem` ディレクトリに何も表示されない場合は、PCMCIA メモリーカードがマウントされていないか、または正しくフォーマットされていないかのいずれかです。これを調べるには、`mount` コマンドを実行して、`/pcmem` 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)。

```
/pcmem/name on /vol/dev/diskette0/name ...
```

という行が表示されない場合、PCMCIA メモリーカードはマウントされていません。コンソールのエラーメッセージを確認してください。

▼ PCMCIA メモリーカードの内容を調べる方法

`/pcmem` の下にあるディレクトリの一部がシンボリックリンクであるため、`ls` コマンドに `-L` オプションを付けて使用してください。

```
$ ls -L [-l] pcmem0
```

`-L` 出力にシンボリックリンクを含む。

`-l` 長形式。アクセス権と所有権を出力に含む。

例 — PCMCIA メモリーカードの内容を調べる

次の例では、`pcmem0` で識別される最初のフロッピードライブの PCMCIA メモリーカードの内容を一覧表示しています。

```
$ ls -L -l /pcmem/pcmem0
-rwxrwxrwx 1 smith staff 362284 Nov 16 20:54 text.doc
-rwxrwxrwx 1 smith staff 24562 Nov 16 12:20 art.gif
```

▼ PCMCIA メモリーカードから情報をコピーまたは移動する方法

PCMCIA メモリーカードを挿入すると、他のファイルシステムの場合と同様に、そのファイルおよびディレクトリにアクセスできます。ただし、所有権とアクセス権については注意が必要です。たとえば、PCMCIA メモリーカードのファイルの所有者ではない場合、PCMCIA メモリーカード上でそれを変更することはできません。また、ファイルシステムにファイルをコピーした場合、そのファイルの所有者にはなりますが、書き込み権はありません (PCMCIA メモリーカードでも書き込み権がなかったためです)。アクセス権は各自で変更する必要があります。

1. **PCMCIA** メモリーカードがフォーマットされてマウントされていることを確認します。

```
$ ls /pcmem
pcmem0 PCMCIA memorycard-name
```

PCMCIA メモリーカードが正しくフォーマットされてマウントされていれば、その名前とシンボリックリンクが、/pcmem の下に表示されます。

/pcmem ディレクトリに何も表示されない場合、PCMCIA メモリーカードはマウントされていません。244ページの「PCMCIA メモリーカードをロードする方法」を参照してください。PCMCIA メモリーカードがフォーマットされていないこともあります。236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

コピーするもの	使用するコマンド
ファイル	cp
ディレクトリ	cp -r

3. コピーまたは移動の結果を確認するには、ls コマンドを使用します。

例 — PCMCIA メモリーカードから情報をコピーまたは移動する

最初の例では、ファイル (readme.doc) を PCMCIA メモリーカードから現在のディレクトリ (「.」で示される) に移動しています。2 番目の例では、ファイル (readme2.doc) を PCMCIA メモリーカードから現在のディレクトリにコピーしています。3 番目の例では、ディレクトリ (morefiles) とその下にあるすべてのものを、PCMCIA メモリーカードから現在のディレクトリにコピーしています。

```
$ mv /pcmem/pcmem0/readme.doc .  
$ cp /pcmem/pcmem0/readme2.doc .  
$ cp -r /pcmem/pcmem0/morefiles .
```

▼ PCMCIA メモリーカードへ情報をコピーまたは移動する方法

1. **PCMCIA** メモリーカードが書き込み保護になっていないことを確認します。
書き込み保護は、PCMCIA メモリーカードの端にある小さなスライドスイッチによって設定されます。
2. **PCMCIA** メモリーカードがフォーマットされてマウントされていることを確認します。

```
$ ls /pcmem  
pcmem0 PCMCIA memorycard-name
```

PCMCIA メモリーカードが正しくフォーマットされてマウントされていれば、その名前とシンボリックリンクの `pcmem0` が、`/pcmem` の下に表示されます。

`/pcmem` ディレクトリに何も表示されない場合、PCMCIA メモリーカードはマウントされていません。244ページの「PCMCIA メモリーカードをロードする方法」を参照してください。PCMCIA メモリーカードがフォーマットされていないこともあります。236ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 241ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

3. ファイルまたはディレクトリを移動またはコピーします。

作業	使用するコマンド
ファイルのコピー	cp
ディレクトリのコピー	cp -r
ファイルまたはディレクトリの移動	mv

4. ls コマンドを使用して、移動またはコピーの結果を確認します。

例 — 情報を PCMCIA メモリーカードにコピーまたは移動する

最初の例では、ファイル (readme.doc) を現在のディレクトリから、最初のカードドライブ (/pcmem/pcmem0) にロードされた PCMCIA メモリーカードに移動しています。2 番目の例では、ファイル (readme2.doc) を現在のディレクトリから、2 番目のカードドライブ (/pcmem/pcmem1) にロードされた PCMCIA メモリーカードにコピーしています。3 番目の例では、ディレクトリ (morefiles) とその下にあるすべてのものを、/home/smith/ ディレクトリから最初のカードドライブにロードされた PCMCIA メモリーカードにコピーしています。

```
$ mv readme.doc /pcmem/pcmem0
$ cp readme2.doc /pcmem/pcmem1
$ cp -r /home/smith/morefiles /pcmem/pcmem0
```

▼ PCMCIA メモリーカード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. fuser コマンドを起動します。

fuser コマンドは、指定された CD に現在アクセスしているプロセスを表示します。

```
# fuser -u [-k] pcmem0
```

-u	PCMCIA メモリーカードを使用中のユーザーを表示する。
-k	PCMCIA メモリーカードにアクセスしているプロセスを終了させる。

例 — PCMCIA メモリーカードが使用中かどうかを調べる

次の例では、プロセス 6400c と 6399c が /pcmem/pcmem0 ディレクトリにアクセスしていて、プロセス所有者はそれぞれ root と smith です。

```
# fuser -u /pcmem/pcmem0
/pcmem/pcmem0: 6400c(root) 6399c(smith)
```

プロセスは、root 権限によって個別に終了することも、fuser コマンドに -k オプションを付けて使用して、ファイルシステムにアクセス中のすべてのプロセスを終了することもできます。

```
# fuser -u -k /pcmem/pcmem0
/pcmem/pcmem0: 6400c(root)Killed 6399c(smith)Killed
```

fuser コマンドは、終了したプロセスの一部を識別できないことがあります。確認するには、-u オプションを付けてもう一度このコマンドを実行してください。

▼ PCMCIA メモリーカードを取り出す方法

1. **PCMCIA** メモリーカードが使用中でないことを確認します。

PCMCIA メモリーカードは、シェルまたはアプリケーションがそのファイルやディレクトリにアクセスしている場合「使用中」であることを忘れないでください。

PCMCIA メモリーカードのすべてのユーザーを検出したかどうか不確かな場合は (デスクトップツールの背後に隠れたシェルがアクセスしている可能性がある場合は)、249ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」に説明されている fuser コマンドを使用してください。

2. **PCMCIA** メモリーカードを取り出します。

```
# eject pcmem0
```

手で PCMCIA メモリーカードを取り出す必要があります。ウィンドウを実行している場合は、PCMCIA メモリーカードを取り出すように指示する画面メッセージに注意してください。

PCMCIA メモリーカードが使用中の場合は、次のメッセージが表示されます。

```
/vol/dev/pcmem/noname: Device busy
```

この場合は、手順 1 に戻って誰も PCMCIA メモリーカードを使用していないことを確認してから、もう一度それを取り出してください。

▼ 他のシステム上の PCMCIA メモリーカードにアクセスする方法

他のシステム上の PCMCIA メモリーカードを各自のファイルシステムに手作業でマウントすることによって、その PCMCIA メモリーカードにアクセスできます。ただしこれは、他のシステムが、253ページの「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方法」の指示に従って PCMCIA メモリーカードドライブをエクスポートしている場合にかぎります。

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、あるいは作成します。

```
$ mkdir directory
```

directory

他のシステムの PCMCIA メモリーカードのマウントポイントとして使用するディレクトリの名前

2. マウントしたい **PCMCIA** メモリーカードの名前を検索します。

手作業でリモート PCMCIA メモリーカードをマウントする場合は、ローカル PCMCIA メモリーカードで使用可能な `pcm0` または `floppy1` 変数を使用できません。正しい PCMCIA メモリーカード名を使用する必要があります。この名前を検索するには、リモートシステムの `/pcm0` ディレクトリで `ls` コマンドを使用してください。オートマウンタが実行されている場合は、マウントしたい PCMCIA メモリーカードのシステムに `cd` コマンドで移動してから、`ls` コマン

ドを使用できます。オートマウンタが実行されていない場合は、リモートからロ
グインするなどの別の方法を使用する必要があります。

3. スーパーユーザーとして、**PCMCIA** メモリーカードをマウントします。

```
# mount -F nfs system-name:/pcmem/PCMCIAmemory-card-name local-mount-point
```

<i>system-name</i>	マウントしたい PCMCIA メモリーカードが存在するシ ステムの名前
<i>PCMCIA memory-card-name</i>	マウントしたい PCMCIA メモリーカードの名前
<i>local-mount-point</i>	リモート PCMCIA メモリーカードのマウント先の 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. **PCMCIA** メモリーカードが実際にマウントされたかどうかを確認するには、`ls`
コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /pcmem
```

例 — 他のシステム上の **PCMCIA** メモリーカードにアクセスする

以下の例では、`myfiles` という名前の 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
```

▼ 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方法

システムを設定して、その PCMCIA メモリーカードをエクスポートすることができます。つまり、これらのドライブ上の PCMCIA メモリーカードを、他のシステムが使用できるようになります。PCMCIA メモリーカードドライブがエクスポートされると、他のシステムは、251ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれる PCMCIA メモリーカードにアクセスできます。

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. 以下のうちの該当する手順に進みます。

条件	次の手順
<code>nfsd</code> が実行されている場合	254ページの手順 8
<code>nfsd</code> が実行されていない場合	253ページの手順 4

4. `nfsd` がエクスポートするダミー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る。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」、エクスポートされた PCMCIA メモリーカード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` 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、このエントリを参照して、エクスポートされた PCMCIA メモリーカードドライブを認識します。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. 現在ドライブ内にある PCMCIA メモリーカードを取り出します。

```
# eject pcmem0
```

9. 書き込み権を `/etc/rmmount.conf` に割り当てます。

```
# chmod 644 /etc/rmmount.conf
```

10. 次の行を `/etc/rmmount.conf` に追加します。

```
# File System Sharing
share floppy*
```

上記の行によって、システムの PCMCIA メモリーカードドライブにロードされた PCMCIA メモリーカードが共有されます。

11. `/etc/rmmount.conf` 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. **PCMCIA** メモリーカードをロードします。

```
---PCMCIA メモリーカードを挿入する---  
# volcheck -v  
media was found
```

ここでロードした PCMCIA メモリーカードは他のシステムで使用できるようになります。PCMCIA メモリーカードにアクセスするために、リモートユーザーは、251ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」の指示に従って、名前によりその PCMCIA メモリーカードをマウントする必要があります。

13. **PCMCIA** メモリーカードが実際に他のシステムで使用できるかどうかを確認するには、`share` コマンドを使用してください。

PCMCIA メモリーカードが使用可能な場合は、その共有の設定が表示されます。(共有されるダミーディレクトリも表示されます。)

```
# share  
- /dummy ro "dummy dir to wake up NFS daemon"  
- /myfiles rw ""
```

例 — ローカル **PCMCIA** メモリーカードを他のシステムで使用可能にする

次の例では、ローカルシステムの CD-ROM ドライブにロードされた PCMCIA メモリーカードすべてを、ネットワーク上の他のシステムで使用できるようにしています。

```
# 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    ""
```


ボリューム管理の動作 (参照情報)

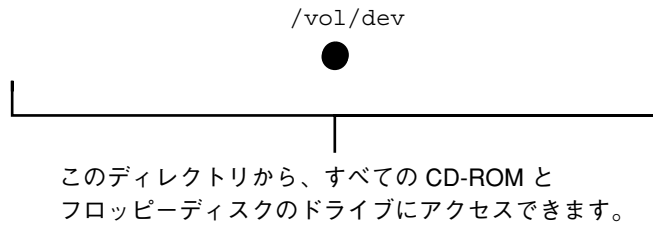
この章では、取り外し可能な媒体を格納するためにボリューム管理が作成するマウントポイントとシンボリックリンクについて説明します。

この章で説明する内容は次のとおりです。

- 257ページの「すべての取り外し可能な媒体のマウント」
- 258ページの「フロッピーディスクへのアクセス」
- 259ページの「CD へのアクセス」
- 260ページの「アクセスを容易にするためのマウントポイント」
- 262ページの「2 種類のシンボリックリンク」
- 263ページの「UFS フォーマットによる制限」

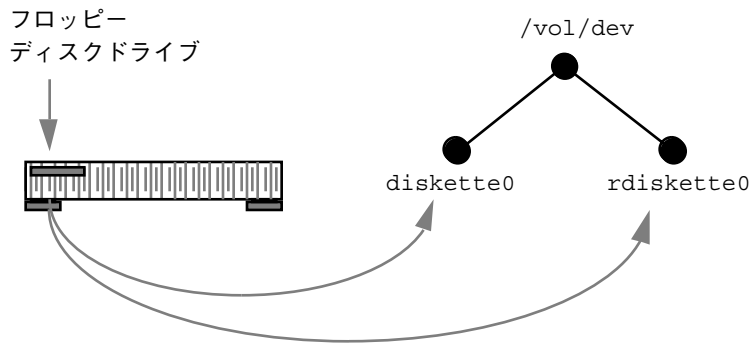
すべての取り外し可能な媒体のマウント

ボリューム管理は、`/vol/dev` の下にあるすべての CD-ROM とフロッピーディスクのドライブへのアクセスを提供します。



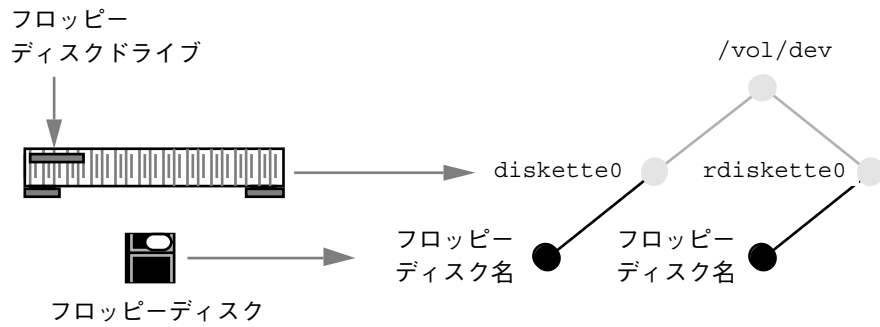
フロッピーディスクへのアクセス

ボリューム管理は、/vol/dev のサブディレクトリ、つまり `diskette0` と `rdiskette0` から、システムのフロッピーディスクドライブへのアクセスを提供します。



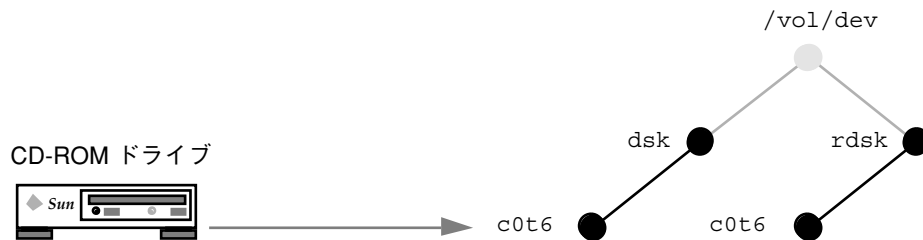
システムに 2 番目のフロッピーディスクドライブがある場合、ボリューム管理は、`diskette1` と `rdiskette1` という 2 番目の対のディレクトリを作成します。3 つ目のフロッピーディスクドライブの場合は、`diskette2` と `rdiskette2` という対が作成されます。ドライブが追加されるたびに、同様に対のディレクトリが作成されます。

`diskette` ディレクトリからはファイルシステムに、`rdiskette` ディレクトリからは raw データに、それぞれアクセスできます。フロッピーディスク自体は、ドライブディレクトリの下にあるサブディレクトリに示されます。この章の図では、ノードのいくつかは説明をわかりやすくするためにグレーで示されています。これは構造上の違いを示すものではありません。

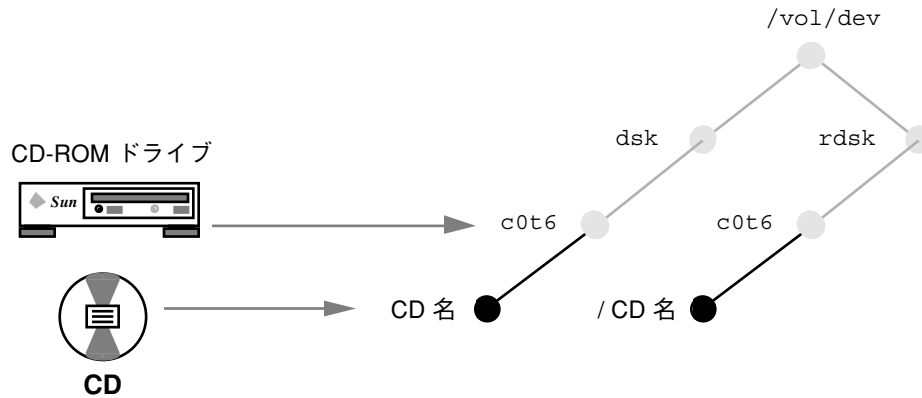


CD へのアクセス

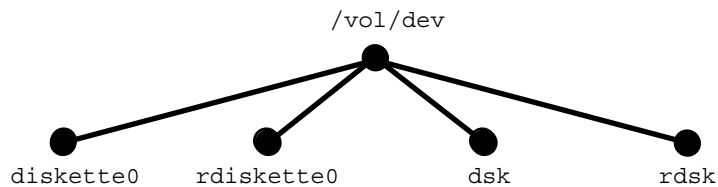
CD の場合は、フロッピーディスクの場合と似ていますが、ブロックディレクトリと raw ディレクトリにそれぞれ /dsk および /rdsk という名前が付けられている点が異なります。また、CD-ROM デバイスは、実際にはこれらよりも 1 つ下のディレクトリにあります。



上記の図では、そのディレクトリは c0t6 と命名されています。これは、特定システムのデバイス命名規則を反映しています。このため、使用するシステムによっては、ディレクトリが同じ形式であってもディレクトリ名が異なる場合があります。ただし、CD 自体は、それらのデバイスに属するディレクトリの下にマウントされるという点で、フロッピーディスクと似た規則に従っています。



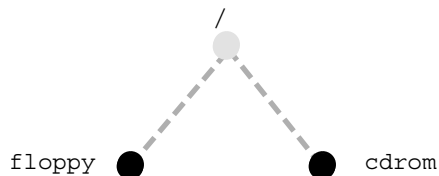
したがって、1つのフロッピーディスクドライブと1つのCD-ROMドライブを備えたシステムの `/vol/dev` ファイルシステムは、次のようになります。



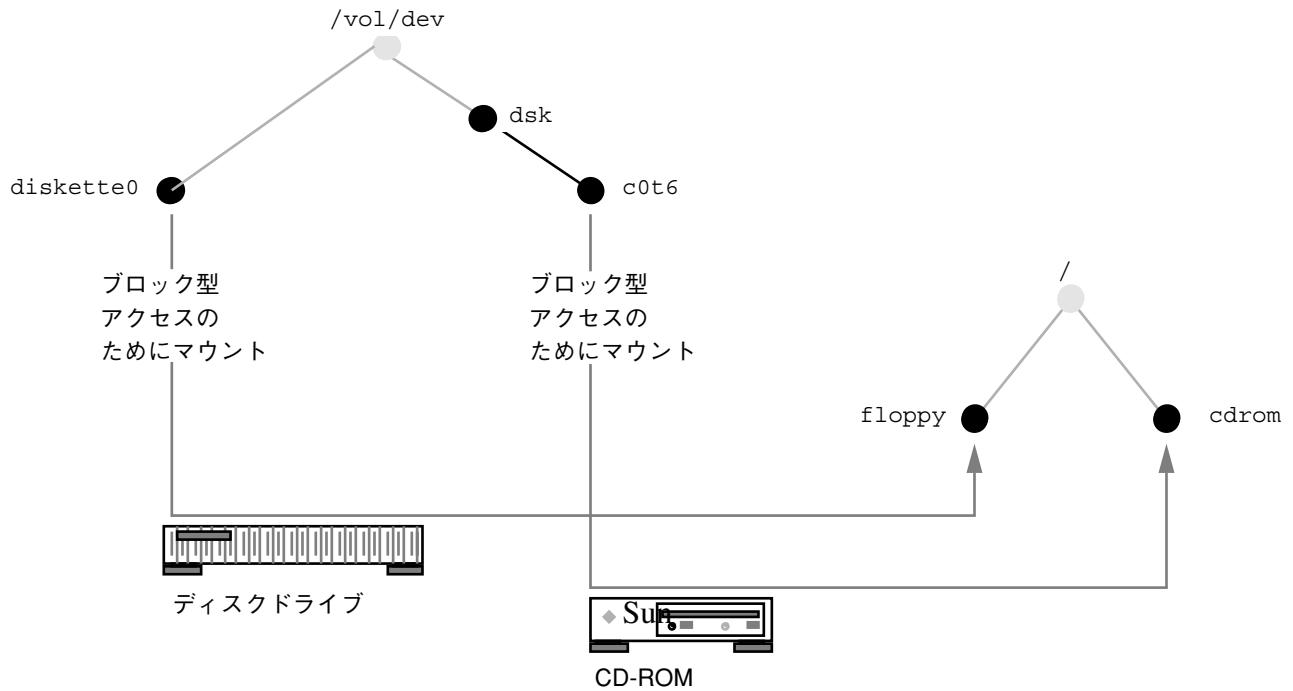
(実際には、`/vol/dev` には、`aliases` という名前のサブディレクトリが含まれますが、これについては後で説明します。)

アクセスを容易にするためのマウントポイント

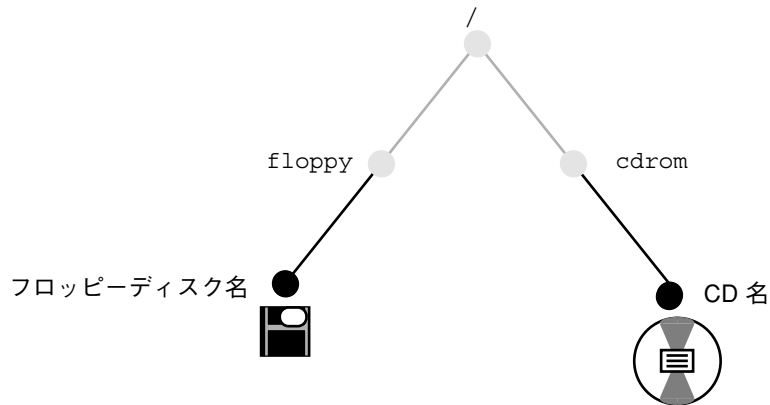
アクセスをより容易にするために、ボリューム管理は、`/floppy` および `/cdrom` という2つの特殊なマウントポイントを使用します。



ボリューム管理は、`/vol/dev/diskette0` および `/vol/dev/dsk/c0t6` というディレクトリを、`/floppy` および `/cdrom` にマウントします。



これらのマウントポイントがあるため、フロッピーディスクを挿入すると、`/floppy/diskette-name` でアクセスできます。同様に、CD を挿入すると、`/cdrom/cd-name` でアクセスできます。



ただし、これらのマウントポイントは、適切なフォーマットが必要です。フロッピーディスクがフォーマットされていればマウントは成功しますが、フォーマットされていないと、マウントは失敗して、フロッピーディスクは `/vol/dev/diskette0`

でしか使用できません。フロッピーディスクは、210ページの「UFS フロッピーディスクをフォーマットする方法」または 215ページの「DOS フロッピーディスクをフォーマットする方法」の説明に従ってフォーマットできます。

システムに複数のドライブがある場合は、`/floppy/floppy0`、`/floppy/floppy1`、`/cdrom/cdrom0` などの並列のディレクトリにマウントされます。

2 種類のシンボリックリンク

ボリューム管理は、さらに次の2つの個別のシンボリックリンクセットを作成します。

- ファイルシステムアクセス用
- raw デバイスアクセス用

ファイルシステムアクセス用シンボリックリンク

ファイルシステムアクセス用のシンボリックリンクは、ディレクトリ `/floppy/floppy0` と `/cdrom/cdrom0` を、最初のフロッピーディスクドライブに挿入されたフロッピーディスクと、最初の CD-ROM ドライブに挿入された CD にリンクします。

```
/floppy/floppy0 --> /floppy/name --> /vol/dev/diskette0/name
```

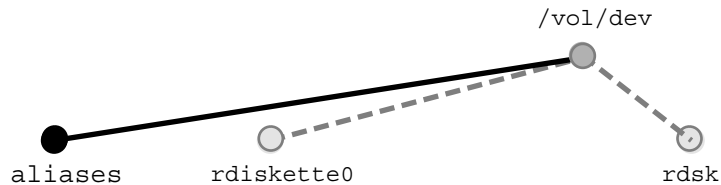
```
/cdrom/cdrom0 --> /cdrom/name --> /vol/dev/dsk/c0t6/CD-name
```

これらのリンクを使用すると、名前を知らないフロッピーディスクや CD にアクセスすることができます。リンク名 `floppy0` または `cdrom0` を名前の代わりに使用できます。

後続のドライブに挿入されたフロッピーディスクと CD は、表 14-3 に要約した命名規則に従います。

raw デバイスアクセス用シンボリックリンク

raw デバイスへのアクセスをさらに容易にするために、ボリューム管理は、`/vol/dev` の下に `aliases` ディレクトリを作成します。



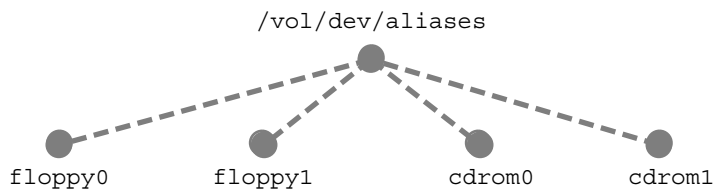
ボリューム管理は、aliases ディレクトリの下に、ブロック型アクセス用に使用されるものに似たシンボリックリンクのセットを作成します。つまり、raw データをアクセスする場合、これらのディレクトリは同等に扱うことができます。

```
/vol/dev/aliases/floppy0 --> /vol/dev/rdiskette0/diskette-name
```

```
/vol/dev/aliases/cdrom0 --> /vol/dev/rdsk/c0t6d0/cd-name
```

ファイルシステムアクセス用シンボリックリンクと同様、これらのリンクの目的は、raw データフロッピーディスクまたは CD の名前を知らなくても、アクセスできるようにすることにあります。つまり、/vol/dev/aliases/floppy0 および /vol/dev/aliases/cdrom0 の各リンク名を使用します。

上記の例は、フロッピーディスク用と CD 用にそれぞれ 1 つずつシンボリックリンクがある場合を示しています。システムに 2 つのフロッピーディスクまたは 2 つの CD がある場合は、次に示すように、それぞれに 1 つずつシンボリックリンクが作成されます。



UFS フォーマットによる制限

UFS フォーマットは、アーキテクチャが異なると互換性がないため、それらがフォーマットされたアーキテクチャ上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 CD は、IA システムでは認識されません。同様に、IA 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。

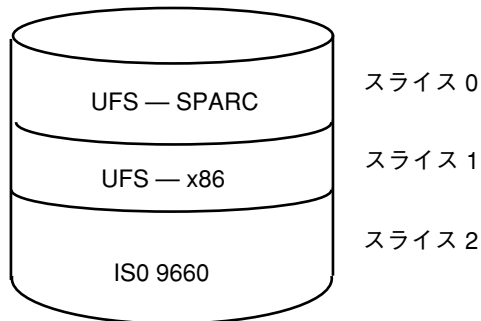
したがって、ボリューム管理は、SPARC システム上の IA UFS 媒体を認識してマウントすることはできません。また、IA システム上の SPARC UFS 媒体も認識できません。

ほとんどの CD は、ISO 9660 標準 (High Sierra File System — HSFS) に従ってフォーマットされます。これは、ボリューム管理をまったく制限しないため、CD において互換性が問題になることはほとんどありません。

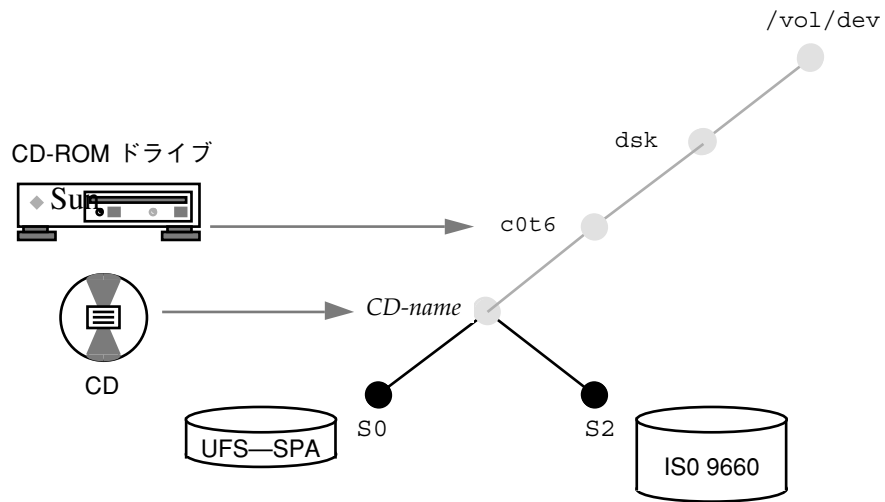
フロッピーディスクの場合、UFS の互換性の問題は、CD の場合よりも頻繁に起こる可能性があります。これは、ユーザーがフォーマットを設定できるためです。あるアーキテクチャで UFS フロッピーディスクをフォーマットした場合、それを別のアーキテクチャでは使用できないことに注意してください (210ページの「UFS フロッピーディスクをフォーマットする方法」を参照してください)。

混合 (ハイブリッド) フォーマットについて

いくつかの CD、特にインストール用 CD には、混合フォーマットが含まれます。つまり、一部が UFS で、一部 ISO 9660 です。異なるフォーマットを格納するために、CD は、実質的に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。さらに、CD をいくつかの異なるアーキテクチャで使用できるようにするために (異なる PROM アーキテクチャがシステムのブートに使用される可能性のあるインストールの場合などのために)、複数の UFS フォーマットが CD にロードされます。



この場合ボリューム管理は、ローカルシステムのアーキテクチャに固有ではない UFS フォーマットを無視して、適切な UFS スライスと ISO 9660 スライスをマウントします。



これらのスライスは、サブディレクトリとして、`/vol/dev/dsk/c0t6` と `/cdrom/cdrom0` の両方の下に現れます。

```
$ ls /cdrom/cdrom0
S0 S2
$ ls /vol/dev/dsk/c0t6
S0 S2
```


ソフトウェアの管理

ここでは、Solaris ソフトウェアのパッケージとパッチの管理について説明します。次の章で構成されています。

第 20 章	Solaris 環境における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 21 章	クライアントのタイプ別に、ソフトウェアパッケージをインストールしたり削除したりする手順について説明します。
第 22 章	Solaris 環境でのパッチの追加と削除についての概要と手順を説明します。

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、また、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と管理についての概要を示します。ここでは 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。

この章の内容は次のとおりです。

- 269ページの「ソフトウェア管理についての参照先」
- 270ページの「ソフトウェアパッケージの概要」
- 271ページの「ソフトウェア管理ツール」
- 272ページの「パッケージの追加または削除時の動作」
- 273ページの「パッケージを追加または削除する前に知っておくこと」
- 273ページの「クライアントソフトウェア管理に関するガイドライン」
- 274ページの「パッケージの削除に関するガイドライン」
- 275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」

ソフトウェア管理についての参照先

ソフトウェアを管理する手順については、次を参照してください。

手順	参照先
Solaris ソフトウェアのインストール	『Solaris 8 インストールガイド (SPARC 版)』または『Solaris 8 インストールガイド (Intel 版)』
Solaris 8 をインストールした後のソフトウェア追加と削除	第 21 章
インストール後の Solaris パッチの追加または削除	第 22 章
ソフトウェア管理の問題の解決	『Solaris のシステム管理 (第 2 巻)』の「ソフトウェア管理の問題の解決」

ソフトウェア管理における新機能

今回のリリースの Solaris には、新しいソフトウェア管理ツール Solaris Product Registry が含まれています。Solaris 8 リリースをインストールした後、Product Registry は Solaris Web Start 3.0 または Solaris パッケージ管理コマンドでインストールしたすべてのソフトウェアのリストを提供します。Product Registry を表示するには、コマンド行で `/usr/bin/prodreg` と入力します。

詳細は、『Solaris 8 インストールガイド (SPARC 版)』または『Solaris 8 インストールガイド (Intel 版)』の「Product Registry によるソフトウェアの追加と削除」を参照してください。

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 およびそのサードパーティベンダーは、ソフトウェアパッケージングと呼ばれる形式で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式を指します。) パッケージの最も単純な形式としては、定義済みフォーマットのファイルとディレクトリの集合を考えることができます。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。この 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て、パッケージをインストールまたは削除したり、インストールを検査する方法を提供するユーティリティがあります。

ソフトウェア管理ツール

システムにソフトウェアを追加したり、システムから削除するために、次の3つのツールがあります。

- パッケージコマンド (pkgadd、pkgrm、pkginfo)
- Admintool (システム管理ツール)

パッケージ情報を追加、削除、および表示するツール	説明
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んだり、オプションのファイルを設定してユーザーの対話操作や特別なチェック作業をなくしたり、また、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ことができる。すでにこれらのコマンドによるパッケージの追加と削除に詳しいユーザーの場合は、これらのコマンドを引き続き使用した方が簡単である。
Admintool	グラフィカルインタフェースツールの使用方法についての一般的な説明をオンライン表示する。ソフトウェアパッケージの命名規則やコマンド行オプションの使用方法にまだ慣れていないユーザー、また、一度に1つのシステムでしかソフトウェアの管理を行わないユーザーの場合は、Admintoolを使用して、ソフトウェアの追加、削除を行うことを推奨する。
Solaris Product Registry	製品を追加するためのインストーラを起動する。

次の表に、ソフトウェアを管理するとき、pkgadd コマンドや pkgrm コマンドよりも Admintool を使用する利点を説明します。

表 20-1 Admintool のソフトウェア管理機能

ソフトウェア管理作業	Admintool を使用して実行可能かどうか
スタンドアロンシステム、またはサーバーシステムでのパッケージの追加と削除	可能
インストールされたソフトウェアすべての簡単な表示	可能
インストール用媒体上のパッケージの簡単な表示と選択	可能
スプールディレクトリへのパッケージの追加	不可
管理ファイルを使用したユーザー対話操作の省略	不可

初期の Solaris リリースでは、ソフトウェアの追加、削除を行うグラフィカルユーザーインタフェースのツールとしてソフトウェアマネージャ (swmtool コマンドでアクセス) を使用しました。swmtool コマンドを Solaris 2.5 および以降の互換バージョンで使用すると、Admintool が起動します。

パッケージの追加または削除時の動作

pkgadd コマンドと pkgrm コマンド、または Admintool はソフトウェアを追加したり削除したりするのに使用します。Admintool は pkgadd コマンドと pkgrm コマンドのグラフィカルインタフェースです。

パッケージを追加する際、pkgadd ソフトウェアは、ファイルを解凍して、インストール用媒体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フォーマットで配布され、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 コマンドと pkgrm 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た製品について追跡を行います。pkgadd と pkgrm 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 および pkgrm の各コマンドは、システムにインストールされたすべてのソフトウェア製品の記録を追跡します。

パッケージを追加または削除する前に知っておくこと

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にあたっては、次のことを知っておく必要があります。

- パッケージの命名規則 – Sun パッケージは、SUNWvolr、SUNWadmap、SUNWab2m などのように、必ず接頭辞 SUNW で始まります。サードパーティのパッケージは、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 – Solaris Product Registry、Admintool、または pkginfo コマンドを使用して、システムにインストール済みのソフトウェアを表示できます。
- 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。（これ以降の節でクライアントソフトウェアの管理方法について詳しく説明します。）

クライアントソフトウェア管理に関するガイドライン

スタンドアロンシステムでのソフトウェアの管理は、パッケージのインストール用ツールと規則を理解すれば、かなり簡単なものです。ソフトウェアパッケージをシステムのローカルディスクにインストールするだけで、そのソフトウェアを使用できるようになります。しかし、クライアントシステム上でソフトウェアを管理する方法は、これよりも難しくなります。ソフトウェアがサーバーとクライアントに

分かれて置かれる場合には特に困難です。(たとえば、1つのソフトウェアに、クライアントのルートファイルシステムにインストールされるファイルを持つパッケージと、クライアントが通常サーバーからマウントする /usr ファイルシステムにインストールされるファイルを持つパッケージがある場合などです。)

パッケージの削除に関するガイドライン

pkgadd コマンドと pkgrm コマンドは、ソフトウェア製品用データベース内の情報を更新するため、パッケージを削除するときは、rm コマンドではなく、pkgrm コマンド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 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とは異なります。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には、removef コマンドを使用してください。これは、ソフトウェア製品用データベースを正しく更新します。詳細は、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をインストールしておきたい場合は(たとえば、複数バージョンの文書処理アプリケーションなど)、新しいバージョンを、すでにインストールされたパッケージとは異なるディレクトリに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、この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とベースディレクトリの設定についての詳細は、275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」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検索して、製品がすでにシステムにインストールされているかどうかを判断します。

パッケージ追加時のユーザーの対話操作を省略する

管理ファイルの使用

pkgadd コマンドを `-a` オプションを指定して実行するとき、pkgadd コマンドは、どのようにインストールを進めるかについての情報を持つ特殊な管理ファイルを参照します。通常、pkgadd 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、プロンプトを出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 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のディレクトリで管理ファイルを探します。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 は `/var/sadm/install/admin` ディレクトリで、指定された管理ファイルを探します。pkgadd コマンドには管理ファイル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。通常は pkgadd が提供するチェックとプロンプトを省略するには、管理ファイルを使用する前に、パッケージのファイルがどこにインストールされているのか、および、パッケージのインストールスクリプトをどのように実行するのかを知っておく必要があります。

次は、pkgadd 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出さないようにする管理ファイルの例です。

```
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の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できます。また、`pkgrm` コマンドでパッケージを削除するときの対話を省略できます。

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ができます。(これは、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指定する、代替ベースディレクトリを管理ファイルに設定します (`-basedir` キーワードを使用する)。詳細は、`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

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含まれます。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ユーザーに質問する (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) `request` スクリプトが含まれます。

インストールしたいパッケージが対話型パッケージであることを、インストール前に知っている場合、さらに、応答を格納しておいて、当該パッケージの将来のインストール時にユーザーの対話操作を省略したい場合は、`pkgask` コマンドを使用してユーザーの応答を保存できます。このコマンドについての詳細は、`pkgask(1M)` のマニュアルページを参照してください。

一度 `request` スクリプトが尋ねる質問への応答を格納すると、`pkgadd` コマンドに `-r` オプションを指定すれば、ユーザーの対話操作なしにパッケージをインストールできます。

ソフトウェアの管理 (手順)

この章では、Solaris コマンドと Admintool グラフィカルインタフェースを使用して、ソフトウェアパッケージをインストール、削除、管理する方法を説明します。

この章で説明する手順は次のとおりです。

- 278ページの「スタンドアロンシステムにパッケージを追加する方法」
- 281ページの「スプールディレクトリにパッケージを追加する方法」
- 284ページの「インストールされたパッケージが完全であるかどうかをチェックする方法」
- 284ページの「インストールされたすべてのパッケージに関する情報を表示する方法」
- 286ページの「パッケージに関する詳細を表示する方法」
- 287ページの「パッケージを削除する方法」
- 287ページの「スプールされたパッケージを削除する方法」
- 288ページの「Admintool を起動する方法」
- 290ページの「Admintool でパッケージを削除する方法」

ソフトウェアパッケージを処理するコマンド

表 21-1 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 21-1 パッケージの追加および削除を行うコマンド

コマンド	説明
prodreg(1)	インストーラでソフトウェアパッケージをインストールする
pkgadd(1M)	ソフトウェアパッケージをインストールする
pkgrm(1M)	ソフトウェアパッケージを削除する
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(1)	ソフトウェアパッケージ情報を表示する
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

パッケージの追加と削除における既知の問題

Solaris 2.5 より前に開発された一部のパッケージを追加、または削除する場合、いくつかの問題があることがわかっています。パッケージの追加または削除に失敗したり、ユーザー入力を要求されたときのユーザー応答内容が無視された場合は、次の環境変数を設定してください。

```
NONABI_SCRIPTS=TRUE
```

パッケージの追加

▼ スタンドアロンシステムにパッケージを追加する方法

1. スーパーユーザーとしてログインします。
2. すでにインストールされているパッケージの中で、追加しようとしているものと同じ名前のパッケージをすべて削除します。

これによってシステムは、追加および削除されたソフトウェアの正しい記録を保持できます。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したい場合もあります。この方法の概要については 274 ページの「パッケージの削除に関するガイドライン」を、手順については 287 ページの「パッケージを削除する方法」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...
```

<code>-a admin-file</code>	(省略可能) インストール時に <code>pkgadd</code> が参照する管理ファイルを指定する。(管理ファイルの使用方法についての詳細は、275 ページの「管理ファイルの使用」を参照。)
<code>-d device-name</code>	ソフトウェアパッケージの絶対パスを指定する。 <code>device-name</code> は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。パッケージのあるパスを指定しないと、 <code>pkgadd</code> コマンドはデフォルトのスプールディレクトリ (<code>/var/spool/pkg</code>) をチェックする。パッケージがそこにない場合、パッケージのインストールは失敗する。
<code>pkgid</code>	(省略可能) インストールされる 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、 <code>pkgadd</code> コマンドは、使用可能なすべてのパッケージをインストールする。

パッケージのインストール中に `pkgadd` に問題が起こると、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。複数のパッケージが指定されている場合は、`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。`pkgadd` は、他のパッケージのインストールを続けます。インストールを中止するには、`quit` と入力してください。

4. パッケージが正常にインストールされたことを確認するには、`pkgchk` コマンドを使用します。

```
# pkgchk -v pkgid
```

pkgchk が、エラーがないと判断した場合は、インストールされたファイルのリストが返されます。そうでない場合は、エラーについて報告されます。

例 — マウントされた CD からソフトウェアをインストールする

次の例は、SUNWaudio パッケージをマウントされた Solaris 8 CD からインストールするコマンドを示しています。この例では、pkgchk コマンドを使用して、パッケージファイルが正しくインストールされたかどうかを検査する方法も示しています。

```
# pkgadd -d /cdrom/sol_8_sparc/s0/Solaris_8/Product SUNWaudio
.
.
.
Installation of <SUNWaudio> was successful.
# pkgchk -v SUNWaudio
/usr
/usr/bin
/usr/bin/audioconvert
/usr/bin/audioplay
/usr/bin/audiorecord
```

例 — リモートパッケージサーバーからソフトウェアをインストールする

インストールしたいパッケージがリモートシステムから利用できるパッケージの場合は、そのパッケージ (パッケージ形式) を含む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、このためのコマンドを示しています。この例では、package-server という名前のリモートシステムが、/latest-packages ディレクトリにソフトウェアパッケージを持つものと想定しています。mount コマンドは、パッケージを /mnt にローカルにマウントし、pkgadd コマンドは、SUNWaudio パッケージをインストールしています。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWaudio
.
.
.
Installation of <SUNWaudio> 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作業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てください。

```
# pkgadd -d /net/package-server/latest-packages SUNWaudio
.
.
.
Installation of <SUNWaudio> 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 という名前前の管理ファイルを指定している点が異なります。このファイルは 275 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」に示されています。この例では、noask-pkgadd 管理ファイルが、デフォルト位置の /var/sadm/install/admin にあるものと想定しています。

```
# pkgadd -a noask-pkgadd -d /net/package-server/latest-packages SUNWaudio
.
.
.
Installation of <SUNWaudio> was successful.
```

スプールディレクトリの使用

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すると便利です。パッケージをデフォルトのスプールディレクトリである /var/spool/pkg にコピーする場合、pkgadd コマンドを使用するときに、パッケージのソース位置 (-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 ディレクトリを探して、コマンド行に指定されたすべてのパッケージを見つけます。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。

▼ スプールディレクトリにパッケージを追加する方法

1. サーバーまたはスタンドアロンシステムのスーパーユーザーとしてログインします。

- すでにスプールされているパッケージの中で、追加しようとしているものと同じ名前のパッケージをすべて削除します。
スプールされたパッケージを削除する方法については、287ページの「スプールされたパッケージを削除する方法」を参照してください。
- 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します。

```
# pkgadd -d device-name -s spooldir pkgid...
```

<code>-d device-name</code>	ソフトウェアパッケージの絶対パスを指定する。 <code>device-name</code> は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。
<code>-s spooldir</code>	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。 <code>spooldir</code> を指定する必要がある。
<code>pkgid</code>	(省略可能) インストールされる 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、 <code>pkgadd</code> は、使用可能なすべてのパッケージをコピーする。

- 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たことを確認するには、`pkginfo` コマンドを使用します。

```
$ pkginfo -d spooldir | grep pkgid
```

`pkgid` が正常にコピーされると、`pkginfo` コマンドは、それに関する 1 行の情報を返します。正常にコピーされないと、システムプロンプトが返されます。

例 — マウントされた CD からスプールディレクトリを設定する

次の例は、SUNWaudio および SUNWab2m の各パッケージを、マウントされた SPARC 版 Solaris 8 CD からデフォルトのスプールディレクトリ (`/var/spool/pkg`) にコピーするコマンドを示しています。

```
# pkgadd -d /cdrom/sol_8_sparc/s0/Solaris_8/Product -s /var/spool/pkg SUNWaudio
Transferring <SUNWaudio> package instance
```

例 — リモート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

インストールしたいパッケージがリモートシステムから利用できるパッケージの場合は、そのパッケージを (パッケージの形で) 含むディレクトリを手作業でマウントして、ローカルスプールディレクトリにそれ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` という名前のリモートシステムが、`/latest-packages` ディレクトリにソフトウェアパッケージを持つものと想定しています。`mount` コマンドは、パッケージを `/mnt` にローカルにマウントし、`pkgadd` コマンドは、`SUNWman` パッケージを `/mnt` からデフォルトのスプールディレクトリ (`/var/spool/pkg`) にコピーします。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

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作業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てください。

```
# pkgadd -d /net/package-server/latest-packages -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

例 — デフォルトのスプールディレクトリからパッケージをインストールする

次の例では、デフォルトのスプールディレクトリから `SUNWman` パッケージをインストールするコマンドを示しています (`pkgadd` にオプションを使用しないと、`/var/spool/pkg` で指定のパッケージが検索されます。)

```
# pkgadd SUNWman
.
.
.
Installation of <SUNWman> was successful.
```

パッケージのインストールの検査

`pkgchk` コマンドを使用すると、パッケージのインストールの完全性、パス名、ファイルの内容、およびファイル属性をチェックできます。オプションの詳細については、`pkgchk(1M)` のマニュアルページを参照してください。

`pkginfo` コマンドを使用して、システムにインストールされたパッケージに関する情報を表示してください。

▼ インストールされたすべてのパッケージに関する情報を表示する方法

`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します。

```
$ pkginfo
```

例 — インストールされたすべてのパッケージを表示する

次の例は、システムがスタンドアロンまたはサーバーのどちらかであるかに関係なく、ローカルシステムにインストールされたすべてのパッケージを表示する

`pkginfo` コマンドを示しています。出力には、基本カテゴリ、パッケージ名、およびパッケージの説明が示されています。

```
$ pkginfo
system      SUNWaccr      System Accounting, (Root)
system      SUNWaccu      System Accounting, (Usr)
system      SUNWadmap     System administration applications
system      SUNWadmc      System administration core libraries
.
.
.
```

▼ インストールされたパッケージが完全であるかどうかをチェックする方法

1. スーパーユーザーとしてシステムにログインします。
2. インストールされたパッケージの状態を、`pkgchk` コマンドによってチェックします。

```
# pkgchk [ -a -c -v ] pkgid ...
# pkgchk -dspooldir pkgid ...
```

- a pkgchk のデフォルトであるファイル属性と内容ではなく、ファイル属性 (つまりアクセス権) だけを検査するように指定する。
- c pkgchk のデフォルトであるファイル内容と属性ではなく、ファイル内容だけを検査するように指定する。
- v pkgchk による処理中、ファイル名を表示する詳細モードを指定する。
- d spooldir スプールディレクトリ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を空白で区切って指定する。pkgid を指定しないと、pkgchk は、システムにインストールされたすべてのソフトウェアパッケージをチェックする。これを省略すると、pkgchk は使用可能なすべてのパッケージを表示する。

例 — インストールされた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チェックする方法を示しています。

```
# pkgchk -c SUNWadmfw
```

pkgchk はエラーがないと判断すると、システムプロンプトに戻ります。そうでない場合は、エラーを表示します。

例 — インストールされた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チェックする方法を示しています。

```
# pkgchk -a SUNWadmfw
```

pkgchk はエラーがないと判断すると、システムプロンプトに戻ります。そうでない場合は、エラーを表示します。

例 — スプールディレクトリにインストールされた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チェック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 <SUNWadmcm>
## checking spooled package <SUNWsdml>
```

注 - スプールされたパッケージの検査は、パッケージがインストールされないときすべての情報を検査できないため、制限されます。

▼ パッケージに関する詳細を表示する方法

pkginfo -l 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します。

```
$ pkginfo -l pkgid ...
```

-l 出力を長形式で表示するように指定する。これには、パッケージに関する入手可能な情報すべてが含まれる。

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。この引数を省略すると、pkginfo は、使用可能なすべてのパッケージに関する情報を表示する。

例 — パッケージに関する詳細を表示する

```
$ pkginfo -l SUNWcar
  PKGINST: SUNWcar
    NAME: Core Architecture, (Root)
  CATEGORY: system
    ARCH: sparc.sun4u
  VERSION: 11.8.0,REV=1999.09.18.11.52
  BASEDIR: /
  VENDOR: Sun Microsystems, Inc.
    DESC: core software for a specific hardware platform group
  PSTAMP: humbolt19990821191439
  INSTDATE: Sep 18 1999 11:53
```

(続く)

```
HOTLINE: Please contact your local service provider
STATUS:  completely installed
FILES:   95 installed pathnames
         31 shared pathnames
         35 directories
         49 executables
         11307 blocks used (approx)
```

サーバーとスタンドアロンシステムからの パッケージの削除



注意 - インストールされたパッケージの削除には、`rm` コマンドではなく、必ず `pkgrm` コマンドを使用してください。 `rm` コマンドを使用すると、インストールされたパッケージのシステム記録に不具合が生じます。

▼ パッケージを削除する方法

1. システムにスーパーユーザーとしてログインします。
2. インストールされたパッケージを削除します。

```
# pkgrm pkgid...
```

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。これを省略すると、使用可能なパッケージすべてが表示される。

▼ スプールされたパッケージを削除する方法

1. スーパーユーザーとしてログインします。

- インストールされたパッケージを、`pkgrm -s` コマンドによってスプールディレクトリから削除します。

```
# pkgrm -s spooldir pkgid...
```

`-s spooldir`

パッケージがスプールされたスプールディレクトリの名前を指定する。

`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` を指定しないと、`pkgrm` は、ユーザーに対して、スプールディレクトリにある各パッケージを削除するように要求する。これを省略すると、使用可能なパッケージすべてが表示される。

Admintool を使用してソフトウェアパッケージを追加または削除する

Solaris オペレーティング環境には Admintool が組み込まれています。これは、ソフトウェアパッケージの追加と削除を含む、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。Admintool を使用すると、特に次のことを実行できます。

- 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た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 を起動する方法

1. インストールされたシステムにログインして、スーパーユーザーになります。シェルプロンプトに、次のように入力します。

```
$ su
```


UNIX の `sysadmin` グループ (グループ 14) のメンバーでない場合、Admintool でソフトウェアパッケージの追加または削除を行うには、システムのスーパーユーザーになる必要があります。

2. **CD** を **CD-ROM** ドライブにロードします。
ボリューム管理が自動的に CD をマウントします。

3. **Admintool** を起動します。

```
# admintool &
```

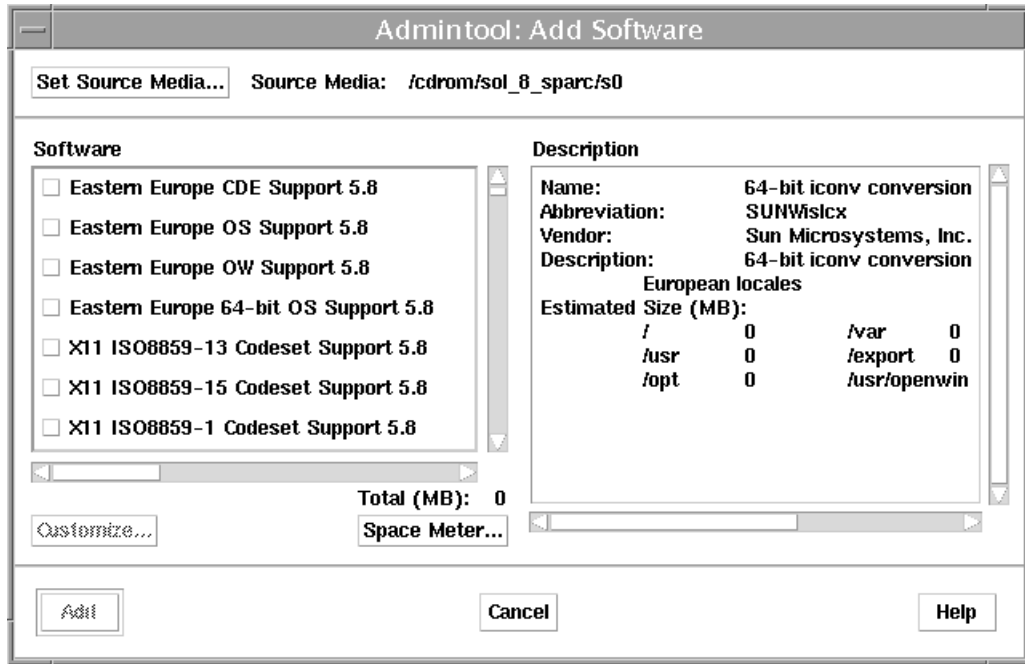
「ユーザー (Users)」ウィンドウが表示されます。

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。

「ソース媒体の設定 (Set Source Media)」ウィンドウが表示されます。必要であれば、インストール媒体へのパスを指定して、「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 SPARC 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したい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、インストールしたい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。
各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表示されます。
「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 でパッケージを削除する方法

1. インストールされたシステムにログインして、スーパーユーザーになります。
シェルプロンプトに、次のように入力します。

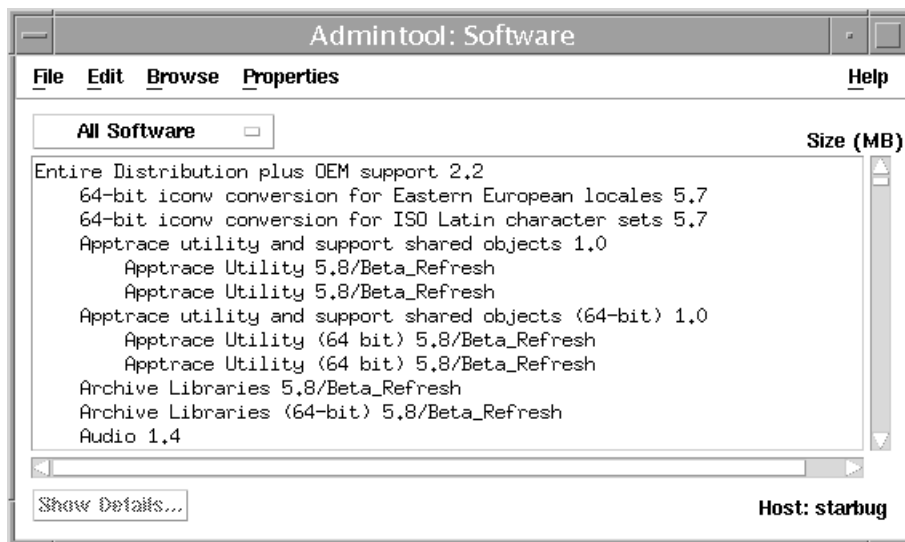
```
$ su
```

UNIX の `sysadmin` グループ (グループ 14) のメンバーでないかぎり、`Admintool` でソフトウェアパッケージを追加または削除するためには、スーパーユーザーにならなければなりません。

2. `Admintool` を起動します。

```
# admintool &
```

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したい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、ソフトウェアを削除したいことを確認します。

削除するパッケージ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。

パッチの管理 (概要)

この章では、実行中の Solaris システムでの Solaris パッチのインストールまたは削除について説明します。また、不要なパッチまたは障害の発生したパッチを削除 (バックアウトと呼びます) する方法についても説明します。

この章の内容は次のとおりです。

- 293ページの「パッチについて」
- 294ページの「パッチ管理用ツール」
- 295ページの「パッチの配布」
- 297ページの「パッチの番号付け」
- 297ページの「パッチの追加に伴う動作」
- 298ページの「パッチの削除に伴う動作」

パッチについて

パッチの最も単純な形式としては、ソフトウェアを実行するのに障害となる既存のファイルとディレクトリを置換または更新する、ファイルとディレクトリの集合と考えることができます。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。(パッケージについての詳細は、第 20 章を参照してください。)

パッチ管理用ツール

パッチ管理用ツールには、次の2つがあります。

- `patchadd` – ディレクトリ形式のパッチを Solaris システムにインストールするために使用する。
- `patchrm` – Solaris システムにインストールされたパッチを削除するために使用する。このコマンドは、ファイルシステムを、パッチが適用される以前の状態に復元する。

パッチのインストールとバックアウトの詳細は、`patchadd(1M)` および `patchrm(1M)` のマニュアルページに示されています。各パッチには、パッチについての特定の情報が入っている `README` ファイルも含まれています。

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。表 22-1 に、すでにシステムにインストールされている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 22-1 パッチ管理に役立つコマンド

コマンド	機能
<code>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> によって指定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。
<code>pkgparam pkgidPATCH_INFO_patch-number</code>	パッチが適用されたホストのインストール日と名前を表示する。 <code>pkgid</code> は、パッケージの名前を表示する。たとえば、 <code>SUNWadmap</code> など。
<code>patchadd -R client_root_path -p</code>	サーバーのコンソールから、クライアントに適用されたすべてのパッチを表示する。
<code>patchadd -p</code>	システムに適用されたすべてのパッチを表示する。

パッチの配布

Sun のユーザーはすべて、セキュリティパッチなどの推奨されるパッチに、World Wide Web または anonymous ftp を介してアクセスできます。サービス契約を購入した Sun ユーザーは、パッチの拡張セットと、パッチ情報に関する完全なデータベースにアクセスできます。この情報は、World-Wide Web、anonymous ftp を介して入手可能であり、通常は CD によって定期的に配布されます (表 22-2 を参照)。

表 22-2 ユーザーパッチアクセス情報

ユーザーの種類	アクセスできる情報
Sun Service 契約ユーザー	パッチに関する SunSolve データベースとパッチ情報にアクセスすることができる。これらの情報は、296ページの「World-Wide Web を介したパッチアクセス」と 296ページの「ftp を介したパッチアクセス」に説明されているように、World-Wide Web または anonymous ftp を介して入手できる。 これらのパッチは、夜間に更新される。ユーザーは、パッチ CD を 6~8 週間ごとに受け取る。
Sun Service 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。これらの情報は、296ページの「World-Wide Web を介したパッチアクセス」と 296ページの「ftp を介したパッチアクセス」に説明されているように、World-Wide Web または anonymous ftp を介して入手できる。

Sun パッチにアクセスするために必要な条件

Sun パッチには、World-Wide Web または anonymous ftp を介してアクセスできます。Sun Service 契約を購入した場合は、定期的に配布されるパッチ CD からパッチを入手できます。

World-Wide Web 上のパッチにアクセスするには、コンピュータが次の条件を満たしていなければなりません。

- インターネットに接続されている
- Netscape などの Web ブラウズソフトウェアを実行できる

anonymous ftp を介してパッチにアクセスするには、コンピュータが次の条件を備えていなければなりません。

- 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

World-Wide Web を介したパッチアクセス

World-Wide Web を介してパッチにアクセスするには、次のユニフォームリソースロケータ (URL) を使用してください。

<http://www.sun.com/>

Sun ホームページが表示されたら、「Sales and Service」ボタンをクリックして、SunSolve パッチデータベースに移動してください。

公開されているパッチのパッチデータベースには、「Public patch access」というラベルが付いています。契約ユーザーが使用できるパッチとパッチ上の包括的なセットに関するパッチデータベースには、「Contract customer patch access」というラベルが付いています。この契約ユーザーデータベースにアクセスするためのパスワードを入力するよう求められます。

URL を使用して、公開されているパッチにアクセスすることもできます。

<http://sunsite.unc.edu/> (日本では <http://sunsite.sut.ac.jp/>)

ftp を介したパッチアクセス

ftp を介してパッチにアクセスするには、ftp を使用して sunsolve1.sun.com (Sun サービス提供) または sunsite.unc.edu (the University of North Carolina 所有) (日本では sunsite.sut.ac.jp/ 東京理科大学所有) のいずれかに接続します。ftp にログインを要求されたら、ログイン名として anonymous と入力します。パスワードの入力を求められたら、完全な電子メールアドレスを使用してください。接続されれば、[/pubs/patches](http://pubs/patches) ディレクトリから、公開されているパッチを検索することができます。

注 - パッチを転送するには、ftp 転送モードをバイナリに変更する必要があります。これには、ftp プロンプトで bin と入力してください。

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 106925-02 は、glm デバイスドライバの問題を修正するための SunOS 5.7 パッチです。

パッチの追加に伴う動作

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。`patchadd` は、さらに次の処理も行います。

- 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、パッチの `pkginfo` ファイルを更新する。

Solaris 2.4 およびその互換バージョンでは、`patchadd` はパッチのインストール時に、パッチ情報の記録を `/var/sadm/patch/patch-number/log` 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。

パッチの削除に伴う動作

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` を使用してインストールされた (`-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た。
- パッチが他のパッチによって要求された。

`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。

デバイスの管理

ここでは、Solaris オペレーティング環境でデバイスを管理する方法について説明します。次の章で構成されています。

第 24 章	デバイス設定の概要を説明します。
第 25 章	デバイスを設定するための手順を説明します。
第 26 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を説明します。

デバイスの管理 (概要)

この章では、Solaris 環境の周辺デバイスを管理する方法の概要を示します。

この章の概要は次のとおりです。

- 303ページの「デバイス管理作業についての参照先」
- 303ページの「デバイスドライバについて」
- 304ページの「デバイスの自動構成」
- 306ページの「デバイス構成情報の表示」

デバイスへのアクセスについては、第 26 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サードパーティ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。

デバイス管理における新機能

この節では、Solaris 8 のデバイス管理における新機能について説明します。

SCSI と PCI のホットプラグ機能

cfgadm コマンドは今回のリリースで更新され、SPARC システムと IA システムの両方において、サポートされる SCSI と PCI のコントローラに SCSI と PCI のホットプラグ機能を提供するようになりました。

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 (特定の SPARC サーバー上で利用可能) を使用すると、システムが動作しているときでも、サービスプロバイダはホットプラグ可能なシステム入出力ボードを取り外したり交換したりできます。したがって、システムをリブートする手間が省けます。また、交換するボードがすぐに入手できない場合、システム処理を継続したまま、システム管理者は動的再構成を使用して障害の発生したボードをシャットダウンできます。

動的再構成が自分のシステムでサポートされているかどうかについては、SPARC ハードウェアメーカーのマニュアルを参照してください。PCI コントローラがホットプラグ機能をサポートしているかどうかを確認するには、『Solaris 8 ハードウェア互換リスト (Intel 版)』を参照してください。

第 25 章では、`cfgadm` コマンドで SCSI または PCI のコントローラをホットプラグする方法について説明しています。

改良されたデバイス構成 (`devfsadm`)

`devfsadm` コマンドによって、ディレクトリ `/dev` と `/devices` 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` 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。そして、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。`/dev` と `/devices` のディレクトリの管理に加えて、`devfsadm` は `path_to_inst(4)` インスタンスデータベースも保守します。

以前の Solaris リリースでは、デバイス構成は `drvconfig` と 5 つのリンクジェネレータが処理していました。つまり、`drvconfig` は `/devices` ディレクトリ内にある物理デバイスエントリを管理し、5 つのリンクジェネレータ (`devlinks`、`disks`、`tapes`、`ports`、`audlinks`) は `/dev` ディレクトリ内にある論理デバイスエントリを管理していました。

これらのユーティリティはホットプラグ可能なデバイスを認識せず、複数のインスタンスを持つデバイスにも十分に対応できません。互換性を保つために、`drvconfig` と 5 つのリンクジェネレータは `devfsadm` ユーティリティへのシンボリックリンクになっています。ホットプラグ可能なデバイスについては、第 25 章を参照してください。

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*` スクリプトから実行されます。

`devfsadm` 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。

デバイス管理作業についての参照先

表 24-1 に、プリンタやモデムなどのシリアルデバイスと、ディスク、CD-ROM、テープドライブなどの周辺デバイスをシステムに追加するための手順を説明している参照先を示します。

表 24-1 デバイスを追加する場合の参照先

作業内容	参照先
ディスクの追加	第 30 章または第 31 章
CD-ROM またはテープデバイスの追加	312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 (第 2 巻)』の「端末とモデム管理の概要」
プリンタの追加	『Solaris のシステム管理 (第 2 巻)』の「印刷管理の概要」

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、磁気バックアップ媒体があるはずです。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 24-2 で説明します。

表 24-2 カーネルモジュール

ディレクトリの位置	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。

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に使用されます。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。

<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 (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 (1M)</code>	最後のリポ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。

システム上のデバイスを識別するために使用されるデバイス名については、336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、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` コマンドと `sysdef` コマンドを使用すると、デバイスインスタンスの次に示される「driver not attached」メッセージを確認することによって、システムに接続されたディスク、テープ、CD-ROM デバイスを識別できます。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
.
.
.
```

同じデバイス情報は、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,p11 (driver not attached)
        se, instance #0
        su, instance #0
        su, instance #1
        ecpp (driver not attached)
        fdthree, instance #0
.
.
.

```

IA 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)

```

(続く)

```

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
date starbug genunix: [ID 540533 kern.notice] SunOS Release 5.8 Generic 64-bit
date starbug genunix: [ID 223299 kern.notice] Copyright (c) 1983-2000 by Sun Microsystems, Inc.
date starbug genunix: [ID 678236 kern.info] Ethernet address = 8:0:20:a6:d4:5b
date starbug genunix: [ID 897550 kern.info] Using default device instance
data
date starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
date starbug unix: [ID 930857 kern.info] avail mem = 121724928
date starbug rootnex: [ID 466748 kern.info] root nexus = Sun Ultra 5/10 UP

```

(続く)

```
A/PCI (UltraSPARC-III 333MHz)
.
.
.
#
```

IA システムからは、次の dmesg 出力が表示されます。

```
# dmesg
date naboo genunix: [ID 540533 kern.notice] SunOS Release 5.8 Version Generic 32-bit
date naboo genunix: [ID 223299 kern.notice] Copyright (c) 1983-
2000 by Sun Microsystems, Inc.
date naboo genunix: [ID 897550 kern.info] Using default device instance data
date naboo unix: [ID 168242 kern.info] mem = 32380K (0x1f9f000)
date naboo unix: [ID 930857 kern.info] avail mem = 19390464
date naboo rootnex: [ID 466748 kern.info] root nexus = i86pc
date naboo rootnex: [ID 349649 kern.info] pci0 at root: space 0 offset 0
date naboo genunix: [ID 936769 kern.info] pci0 is /pci@0,0
date naboo genunix: [ID 678236 kern.info] Ethernet address = 00:a0:24:89:b0:72
date naboo gld: [ID 944156 kern.info] elx0: 3COM EtherLink III:
type "ether" mac address 00:a0:24:89:b0:72
date naboo pci: [ID 370704 kern.info] PCI-device: pci10b7,5950@c, elx0
date naboo genunix: [ID 936769 kern.info] elx0 is /pci@0,0/pci10b7,5950@c
.
.
.
```

デバイスの構成

この章では、Solaris 環境でデバイスを構成する手順について説明します。

この章の内容は次のとおりです。

- 312ページの「周辺デバイスを追加する方法」
- 314ページの「デバイスドライバを追加する方法」
- 319ページの「すべてのデバイスの構成情報を表示する方法」
- 320ページの「SCSI コントローラの構成を解除する方法」
- 321ページの「SCSI コントローラを構成する方法」
- 321ページの「SCSI デバイスを構成する方法」
- 322ページの「SCSI コントローラを切り離す方法」
- 324ページの「SPARC: SCSI デバイスを SCSI バスに取り付ける方法」
- 326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」
- 327ページの「SPARC: SCSI デバイスを取り外す方法」
- 329ページの「IA: PCI スロット構成情報を表示する方法」
- 330ページの「IA: PCI アダプタカードを取り外す方法」
- 331ページの「IA: PCI アダプタカードを取り付ける方法」

デバイスにアクセスする方法については、第 26 章を参照してください。

システム構成要素がホットプラグ機能をサポートしている場合、Solaris 環境では、システムが動作しているときでも、デバイスを取り付け、取り外し、または交換できます。システム構成要素がホットプラグ機能をサポートしていない場合、システムをリブートしてデバイスを再構成しなければなりません。

システムへ周辺デバイスを追加する

新しい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

次のデバイスをシステムに追加するには、以下で説明する手順に従ってください。

- CD-ROM
- 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、サードパーティのデバイスドライバを追加しなければなりません。

▼ 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、314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 <code>init 0</code> 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されたら、電源を落とします。

SPARC システムの場合

IA システムの場合

ok または > プロンプトが表示されたら、電源を落としても安全。

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。

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。

ディスクの裏側に小さいスイッチがあるはずです。

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 26 章を参照してください。


```
.  
. Installation of <XYZdrv> was successful.  
# pkgchk XYZdrv  
#
```

動的再構成とホットプラグ機能

ホットプラグ機能とは、システムが動作しているときでも、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であり、さらに、システム資源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) システム資源 (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、無効にしたりできる機能のことです。

今回のリリースの Solaris では、`cfgadm` コマンドを使用すると、SPARC と IA の両システムでは SCSI デバイスをホットプラグでき、IA のシステムでは PCI アダプタカードをホットプラグできます。`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 319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。IA 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、329ページの「IA: `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。ホットプラグ機能をサポートしている PCI ハードウェアのリストについては、『Solaris 8 ハードウェア互換リスト (Intel 版)』を参照してください。

Sun が提供する高可用性の一部として、ホットプラグ機能は他の階層化製品 (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。デバイス障害に対する耐性が高くなります。

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。

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資源のことです。
- 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` についての詳細は、`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、上記テープデバイスの固有な識別子は次のようになります。

c0::st4

SCSI (デバイスの) ap_id についての詳細は、`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資源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。一般的な状態と操作については、`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。
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	デバイスは構成されていない。	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 接続点の条件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、以降の節を参照してください。

IA: 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資源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。重要なシステム資源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。
- 重要な資源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交換するには、別の階層化ソフトウェアサポートが必要です。

IA: 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、329ページの「IA: 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

この節では、cfgadm コマンドによるさまざまな SCSI ホットプラグ作業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。

▼ すべてのデバイスの構成情報を表示する方法

SCSI コントローラ c0 と c1、およびこれらに接続されているデバイスを使用して、cfgadm コマンドで表示できるデバイス構成情報の例を示します。

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::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
```

(続く)

```
c1::dsk/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。

次の例では、SCSI 接続点だけが表示されます。表示される接続点はシステム構成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

SCSI コントローラ `c1` を使用して、SCSI コントローラ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
```

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。

▼ SCSI コントローラを構成する方法

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

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
```

(続く)

```
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

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。このコマンドを使用するときにも十分に注意してください。

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い可能性があります。

4. SCSI 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。

▼ SCSI コントローラを接続する方法

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. 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::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
```

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::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 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。
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
```

3. **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)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

- 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。
 - b. デバイスを接続して、電源を入れます。
 - c. 新しいデバイスを挿入した後、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。
4. 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

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。

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** 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
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

```
Enter y if operation is complete or n to abort (yes/no)? y
```

- 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。
- 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。交換用のデバイスを取り付けます。こ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 (ターゲットと論理デバイス番号) でなければなりません。そして、取り付けたデバイスの電源を入れます。
- c. デバイスを交換した後、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。

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::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
```

3. SCSI デバイスをシステムから取り外します。

```
# cfgadm -x remove_device c1::disk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

- 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。
- 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. デバイスを取り外した後、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。

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::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
```


SPARC: SCSI 構成の障害対処

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。

解決方法

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。

IA: `cfgadm` コマンドによる PCI ホットプラグ

この節では、別のホットプラグ処理について説明し、IA 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。

▼ IA: PCI スロット構成情報を表示する方法

`cfgadm(1M)` 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。

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
# 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 (このスロットの物理的な識別子) 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。

▼ IA: 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. 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(1M)` 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、ボードを取り外します。

▼ IA: 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。
3. ホットプラグ可能なスロットにアダプタカードを挿入します。

- アダプタカードを挿入してラッチを閉じた後に、どのスロットにアダプタカードが入っているかを確認します。

```
# 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
```

- 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 ホットプラグ可能な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
```

9. 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、イーサネットカードの場合、`ifconfig(1M)` でインタフェースを設定します。

IA: PCI 構成の障害対処

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。

解決方法

`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` 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。

解決方法

接続点が正しいかどうかを確認します。`cfgadm` 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 (1M)` 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。

デバイスへのアクセス (概要)

この章では、システム管理者が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。

- 335ページの「デバイスへのアクセス」
- 337ページの「論理ディスクデバイス名」
- 341ページの「論理テープデバイス名」
- 342ページの「論理 CD-ROM デバイス名」

デバイスの構成についての概要は、第 24 章を参照してください。

デバイスへのアクセス

システム管理者は、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。通常システム管理者は、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現します。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。デバイス階層についての詳細は、『*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 – システム管理者が、デバイスを参照するために使用します。論理デバイス名を使用するファイルコマンドは、表 26-1 に記載されています。`/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。

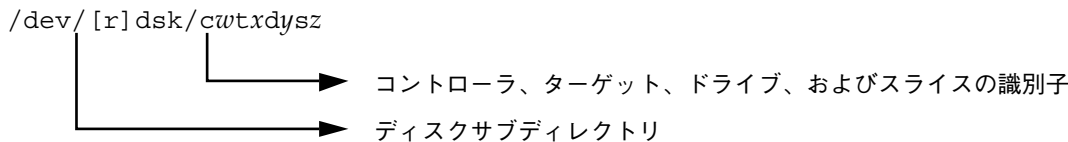
論理ディスクデバイス名

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。
- あるシステムから別のシステムへ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` のいずれか) 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。



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。

表 26-1 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 26-1 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。表 26-2 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 26-2 コントローラのタイプ

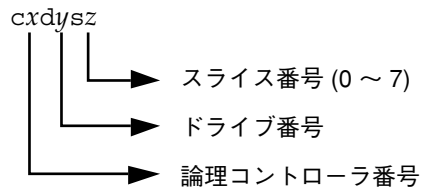
直接コントローラ	バス指向コントローラ
Xylogics (SPARC)	SCSI (SPARC/IA)
IDE (IA)	IPI (SPARC)

両方のタイプのコントローラについて、次の項で説明します。

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。

SPARC: 直接コントローラでアクセスされるディスク

SPARC システムにおいて直接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。

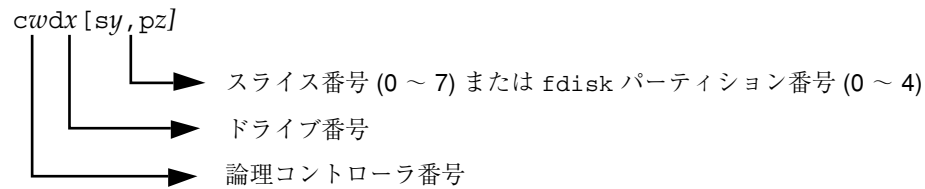


ディスク全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、x は常に 0 になります。

IA: 直接コントローラでアクセスされるディスク

IA 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。

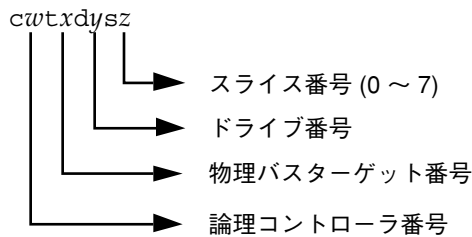


Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、*w* は常に 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。



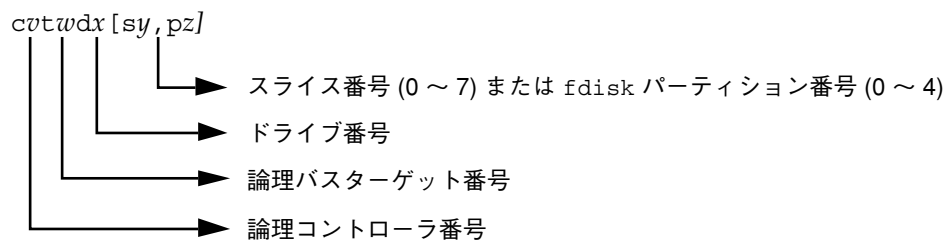
システムにコントローラが 1 つしかない場合、*w* は常に 0 になります。

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。

IA: SCSI コントローラでアクセスされるディスク

IA 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。



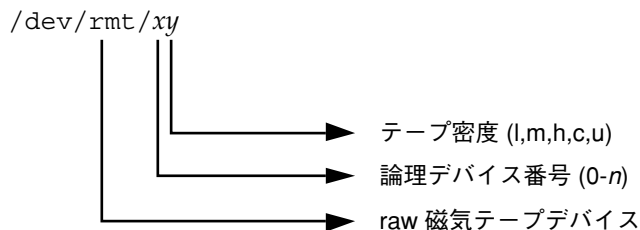
システムにコントローラが1つしかない場合、*v* は常に 0 になります。

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。



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`) であり、これは、QIC-11、QIC-24、QIC-150、または Exabyte のいずれかのタイプになります。テープ密度の値 (l、m、h、c、u) については、第 47 章を参照してください。

論理 CD-ROM デバイス名

システム上の最初の CD-ROM デバイスを表す論理デバイス名は、`/dev/dsk/c0t6d0s0` です。

CD-ROM はボリューム管理によって管理されるため、論理 CD-ROM デバイス名は、CD を手作業でマウントする必要がある場合以外は使用しません。CD-ROM デバイスへのアクセスについては、第 14 章を参照してください。

ディスクの管理

ここでは、Solaris オペレーティング環境でディスクを管理する方法について説明します。次の章で構成されています。

第 28 章	Solaris のディスクスライスの概要と、format ユーティリティを紹介します。
第 29 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 30 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 31 章	IA 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 32 章	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。また、format.dat ファイル、format 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。

ディスクの管理 (概要)

この章では、Solaris のディスクスライスの概念と `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。

- 346ページの「ディスク関連の用語」
- 347ページの「ディスクスライスについて」
- 348ページの「SPARC: ディスクスライス」
- 349ページの「IA: ディスクスライス」
- 352ページの「使用するスライスの決定」
- 353ページの「`format` ユーティリティ」
- 355ページの「`format` ユーティリティ使用上のガイドライン」
- 356ページの「ディスクのフォーマット」
- 357ページの「ディスクラベルについて」
- 357ページの「パーティションテーブル」

システムにディスクドライブを追加する方法については、第 30 章または第 31 章を参照してください。

ディスク管理における新機能

この節では、Solaris 8 リリースで導入された新機能について説明します。

大容量ディスクのサポート

この Solaris リリースでは、BIOS インタフェースが改良されたため、8G バイトを超える大容量ディスクを完全に使用できます。以前は、Intel 版 Solaris を実行している IA ベースのシステムにおいて、次のような制限がありました。

- IDE ディスクの場合、システムが使用できるのは最初の 8G バイトだけでした。
- SCSI または IDE ディスクのどちらの場合でも、ルート (/) スライス用に使用できるのは最初の 8G バイトだけでした。

BIOS が改良されたシステムでは、上記 2 つの制限はなくなりました。

詳細は、Solaris 8 インストールガイド (Intel 版) 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。

- 第 30 章
- 第 31 章

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。

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。

- トラック

- シリンダ
- セクター
- ディスクコントローラ
- ディスクラベル
- デバイスドライバ

上記の用語をよく理解していない場合は、用語集や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、そのファイルシステム用に確保されたシリンダ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。

ファイルシステムについての詳細は、第 34 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 IA 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。表 28-1 に、両者の違いを示します。

表 28-1 プラットフォームによるスライスの違い

SPARC	IA
ディスク全体が Solaris オペレーティング環境になる。	ディスクはオペレーティングシステムごとに 1 つの fdisk パーティションに分割される。
ディスクは 0 から 7 までの番号が付いた 8 つのスライスに分割される。	Solaris の fdisk パーティションは 0 から 9 までの番号が付いた 10 個のスライスに分割される。

SPARC: ディスクスライス

SPARC システム上では、Solaris は 8 つのディスクスライスを定義して、それぞれにある程度決まった役割を割り当てます。これらのスライスには、0 から 7 までの番号が付いています。表 28-2 に、SPARC システム上の 8 つの Solaris スライスの内容を示します。

表 28-2 SPARC: ディスクスライス

スライス	ファイルシステム	通常クライアントまたはサーバーのどちらにあるか	用途
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。
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。スワップ空間は、実行中のプログラムが大きすぎてコンピュータのメモリーに入りきらないときに使用される。その場合、Solaris 環境では、プログラムがメモリーからディスクに「スワップ」され、必要に応じて戻される。
2	—	両方	慣例的に、このスライスはディスク全体を表す。このスライスは、format と Solaris インストールプログラムによって自動的に定義される。このスライスのサイズは変更しないこと。

表 28-2 SPARC: ディスクスライス 続く

スライス	ファイルシステム	通常クライアントまたはサーバーのどちらにあるか	用途
3	/export	サーバーのみ	オペレーティングシステムの代替バージョンを含む。これらの代替バージョンは、サーバーとはアーキテクチャが異なるクライアントシステムに必要である。アーキテクチャのタイプがサーバーと同じクライアントは、/usr ファイルシステム (通常はスライス 6) にある実行可能プログラムを利用する。
4	/export/swap	サーバーのみ	クライアントシステムに仮想メモリー領域を提供する。
5	/opt	両方	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含む。インストール時に、この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 ディレクトリが入る。
6	/usr	両方	ユーザーが実行するオペレーティングシステムコマンド (「実行可能」コマンドとも呼ぶ) を含む。また、このスライスには、オンラインマニュアル、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。

IA: ディスクスライス

IA 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。

表 28-3 のように、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 のスライスを配置します。

表 28-3 IA: ディスクスライス

スライス	ファイルシステム	通常クライアントまたはサーバーのどちらにあるか	用途
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。
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。スワップ空間は、実行中のプログラムが大きすぎてコンピュータのメモリーに入りきらないときに使用される。その場合、Solaris 環境では、プログラムがメモリーからディスクに「スワップ」され、必要に応じて戻される。
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。このスライスは、Sun の format ユーティリティと Solaris インストールプログラムによって自動的に定義される。このスライスのサイズは変更しないこと。
3	/export	サーバーのみ	オペレーティングシステムの代替バージョンを含む。これらの代替バージョンは、サーバーとはアーキテクチャが異なるクライアントシステムに必要である。
4	/export/swap	サーバーのみ	クライアントシステムに仮想メモリーを提供する。
5	/opt	両方	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含む。インストール時に、この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 ディレクトリが入る。
6	/usr	両方	ユーザーが実行するオペレーティングシステムコマンド（「実行可能」コマンドとも呼ぶ）を含む。また、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。

表 28-3 IA: ディスクスライス 続く

スライス	ファイルシステム	通常クライアントまたはサーバーのどちらにあるか	用途
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。スライス番号は 8 であるが、この情報は、Solaris パーティションの先頭にあり、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であり、代替セクター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、シリンダ 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、シリンダ 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。

1. ブロック 0、シリンダ 0 (ディスクラベルが格納される領域)
2. シリンダ 0 全体 (パフォーマンスの向上のため)
3. スライス 2 (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が入っている /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、または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したい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

システム構成ごとに、使用すべきスライスが異なります。表 28-4 に、これらの要件を示します。

表 28-4 システム構成とスライスの要件

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	—	—

表 28-4 システム構成とスライスの要件 続く

スライス	サーバー	スタンドアロンシステム
3	/export	—
4	/export/swap	—
5	/opt	/opt
6	/usr	/usr
7	/export/home	/home

システム構成についての詳細は、102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推奨スライスサイズを表示します。

format ユーティリティ

format ユーティリティの使用方法や参照情報を読む前に概要を知りたい場合は、以下を読んでください。

定義

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。format ユーティリティは、フロッピーディスクドライブ、CD-ROM ドライブ、テープドライブに対しては使用できません。

機能と利点

表 28-5 に、format ユーティリティの機能とその利点を示します。

表 28-5 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	次の情報を表示する。 <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 ユーティリティの全オプションについての詳細は、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場合

Solaris のインストール処理の一部として、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。次のような場合には、format ユーティリティを使用する必要があります。

- 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を追加する。
- ディスクをフォーマットする。

- ディスクを修復する。

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 30 章と第 31 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 28-6 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 28-6 format ユーティリティのガイドライン

用途	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、<code>format</code> 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。 	368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、バックアップ媒体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。 	第 30 章または第 31 章、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 8 のインストール (上級編)』
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をパーティションに分割し直してラベルを付け直すと、既存のデータが失われる。 ■ ディスクにラベルを付け直し、ディスクをパーティションに分割し直す前に、復元するために既存のデータをバックアップ媒体にコピーしなければならない。 	第 30 章または第 31 章

表 28-6 format ユーティリティのガイドライン 続く

用途	注意事項	参照先
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媒体から復元しなければならない。 	第 30 章または第 31 章
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る。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。 	第 32 章

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るので、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。詳細は、368ページの「ディスクがフォーマット済みかどうかを調べる方法」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つのことが行われます。

- ディスク媒体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します。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。詳細は、368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。また、ディスクラベルを表すのに VTOC (Volume Table of Contents) という用語を使用することもあります。「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。表 28-7 にパーティションテーブル関連の用語を示します。

表 28-7 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	パーティション (またはスライス番号)。有効な番号は 0 から 7 まで。
タグ	0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME	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	パーティションは書き込み可能でマウント可能である。
	wu rm	パーティションは書き込み可能でマウント不可である。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。
	rm	パーティションは読み取り専用でマウント可能である。

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、370ページの「ディスクスライス情報を表示する方法」または 375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。

例 — パーティションテーブル

次のパーティションテーブルの例は、1.05G バイトのディスクについて format ユーティリティを使用して表示したものです。

```
Total disk cylinders available: 2036 + 2 (reserved cylinders)
```

(続く)

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

このパーティションテーブルには、次の情報が入っています。

カラム名	説明
Part	パーティション (またはスライス番号)。このカラムについての説明は、表 28-7 を参照。
Tag	パーティションのタグ。このカラムについての説明は、表 28-7 を参照。
Flags	パーティションのフラグ。このカラムについての説明は、表 28-7 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。

次の例では、prtvtoc コマンドを使用してディスクラベルを表示します。

```
# prtvtoc /dev/rdisk/c0t1d0s0
* /dev/rdisk/c0t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   72 sectors/track
*   14 tracks/cylinder
*  1008 sectors/cylinder
*   2038 cylinders
*   2036 accessible cylinders
*
```

(続く)

```

* Flags:
* 1: unmountable
* 10: read-only
*
*
* Partition  Tag  Flags      First      Sector      Last
* Partition  Tag  Flags      Sector     Count       Sector  Mount Directory
      0      2    00         0      303408     303407  /
      1      3    01     303408     225792     529199
      2      5    00         0     2052288     2052287
      6      4    00     529200     1523088     2052287  /usr
    
```

ディスクラベルには、次の情報が入っています。

Dimensions – このセクションには、ディスクドライブの物理的な構成が示されます。

Flags – このセクションには、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が記述されます。パーティションフラグについての説明は、表 28-7 を参照してください。

パーティション (またはスライス) テーブル – このセクションには次の情報が入っています。

カラム名	説明
Partition	パーティション (またはスライス番号)。このカラムについての説明は、表 28-7 を参照。
Tag	パーティションのタグ。このカラムについての説明は、表 28-7 を参照。
Flags	パーティションのフラグ。このカラムについての説明は、表 28-7 を参照。
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	スライス内の最後のセクター番号を示す。
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。その場合の手順は次のとおりです。

- 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、`partition` メニューから `modify` コマンドを使用するのが最も簡単な方法です。`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 M バイト単位で指定してスライスを作成できます。また、「free hog」スライス内の残りのディスク領域を追跡します。

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。

提供側のスライスは、インストール時または `format` ユーティリティの実行時のみ存在します。通常の日常操作中に提供側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の詳細は、396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 416ページの「IA: 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。

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。

この章で説明する手順は次のとおりです。

- 365ページの「システム上のディスクを確認する方法」
- 368ページの「ディスクがフォーマット済みかどうかを調べる方法」
- 368ページの「ディスクをフォーマットする方法」
- 370ページの「ディスクスライス情報を表示する方法」
- 373ページの「ディスクラベルを作成する方法」
- 375ページの「ディスクラベルを検査する方法」
- 376ページの「破損したディスクラベルを復元する方法」
- 381ページの「format.dat のエントリを作成する方法」
- 382ページの「SCSI ドライブを自動構成する方法」
- 385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」
- 386ページの「欠陥セクターを修復する方法」

ディスク管理の概要については、第 28 章を参照してください。

ディスクの管理

表 29-1 作業マップ: ディスクの管理

作業	説明	手順の説明
1. 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。	365ページの「システム上のディスクを確認する方法」
2. 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。	368ページの「ディスクがフォーマット済みかどうかを調べる方法」 368ページの「ディスクをフォーマットする方法」
3. スライス情報の表示	<code>format</code> ユーティリティを使用してスライス情報を表示する。	370ページの「ディスクスライス情報を表示する方法」
4. ディスクラベルの作成	<code>format</code> ユーティリティを使用してディスクラベルを作成する。	373ページの「ディスクラベルを作成する方法」
5. ディスクラベルの検査	<code>prtvtoc</code> コマンドを使用してディスクラベルを検査する。	375ページの「ディスクラベルを検査する方法」
6. <code>format.dat</code> 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 <code>format.dat</code> のエントリを作成する。	381ページの「 <code>format.dat</code> のエントリを作成する方法」
7. 欠陥ディスクセクターの検出	<code>format</code> 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。	385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」
8. 欠陥ディスクセクターの修復 (必要な場合)	<code>format</code> 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。	386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 32 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ります。
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、2 つのディスクを持つシステムのものであります。

```
# 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):
```

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。このため、どの論理名がシステムに接続されたディスクを表しているかを一目で識別できます。論理デバイス名と物理デバイス名については、第 26 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/c2t0d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@0,0
 1. /dev/rdisk/c2t1d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@1,0
 2. /dev/rdisk/c2t2d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@2,0
 3. /dev/rdisk/c2t3d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@3,0
 4. /dev/rdisk/c2t5d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@5,0
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
AVAILABLE DISK SELECTIONS:
 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、IA システム上のディスクを表示します。

```

# 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。IA 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
新しく追加したが、再構成ブートを実行しなかった	第 30 章または 第 31 章
サードパーティのディスク	380ページの「format.dat のエントリの作成」
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	373ページの「ディスクラベルを作成する方法」
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るので、通常はフォーマットしなくてもドライブをインストールできます。

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します。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どうかを調べる方法

1. スーパーユーザーになります。
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、チェックしたい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どうかを調べます。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どうかを調べる

次の例は、ディスク `c0t3d0` がフォーマット済みであることを示しています。

```
# format
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 c0t1d0
[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ります。
2. `format` ユーティリティを起動します。

```
# format
```


- 画面に表示されるリストから、フォーマットしたいディスクの番号を入力します。

```
Specify disk (enter its number): 0
```



警告 - 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、システムディスクに入っているすべてのデータが削除されます。

- 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
```

- 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

次の例では、ディスク `c0t3d0` をフォーマットします。

```
# 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 Wed Jul 14 10:03:34 1999
Formatting ...
done
Verifying media...
    pass 0 - pattern = 0xc6dec6de
    2035/12/18

    pass 1 - pattern = 0x6db6db6d
    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、396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416ページの「IA: 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ります。
2. format ユーティリティを起動します。

```
# format
```

3. AVAILABLE DISK SELECTIONS という見出しの下に表示されたディスクを選択して、スライス情報を表示したいディスクを指定します。

```
Specify disk (enter its number):1
```

4. format> プロンプトで partition と入力して、パーティションメニューに入ります。

```
format> partition
```

5. partition> プロンプトで print と入力して、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
```

6. partition> プロンプトで q と入力し、format プロンプトで q と入力して、format> ユーティリティを終了します。

```
partition> q
format> q
#
```

7. 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

次の例では、ディスク /dev/rdisk/c0t3d0 のスライス情報を表示します。

```

# format
Searching for disks...done
Specify disk (enter its number):1
Selecting c0t3d0
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 2036 + 2 (reserved cylinders)

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
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 28 章を参照してください。

次の例では、ディスク /dev/rdisk/c0t0d0 のスライス情報を表示します。

```

# format
Searching for disks...done
Specify disk (enter its number): 0
selecting c0t0d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 817 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
 0 unassigned wm        3 - 816      6.26GB  (814/0/0) 13128192
 1 unassigned wm         0              0          (0/0/0)    0
 2      backup    wm        0 - 816      6.28GB  (817/0/0) 13176576
 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 unassigned wm         0              0          (0/0/0)    0
 8      boot      wu         0 - 0         7.88MB  (1/0/0)   16128
 9 alternates wu         1 - 2         15.75MB  (2/0/0)   32256
partition> q
format> q

```

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ています。ラベルが付いていないディスクを自動構成できる場合は、次のようなメッセージが表示されます。

```
clt0d0: configured with capacity of 404.65MB
```

▼ 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ります。
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. ディスクラベルの作成方法を決定します。

ディスクにラベルが付いておらず、正常に自動構成された場合

ディスクにラベルが付いており、そのタイプを変更したいか、`format` でディスクを自動構成できなかった場合

ディスクラベルを作成したいかどうかを尋ねるプロンプトが表示される。手順5に進んでラベルを作成する。

ディスクタイプを指定しなければならない。手順6と7に進んでディスクのタイプを設定し、ラベルを付ける。

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9に進んで format ユーティリティを終了します。

- 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。

- 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
```

- 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
```

- format のメインメニューから verify コマンドを使用してディスクラベルを検査します。

```
format> verify
```

- format> プロンプトで q と入力して 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
#
```

▼ ディスクラベルを検査する方法

`prtvtoc(1M)` コマンドを使用して、ディスクラベル情報を検査します。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 28 章を参照してください。

1. スーパーユーザーになります。
2. `prtvtoc` コマンドを使用してディスクラベル情報を表示します。

```
# prtvtoc /dev/rdisk/device-name
```

device-name 検査したい raw ディスクデバイス

例 — ディスクラベルを検査する

次の例は、ディスク `/dev/rdisk/c0t0d0s0` のディスクラベル情報を示しています。

```
# prtvtoc /dev/rdisk/c0t1d0s0
* /dev/rdisk/c0t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   72 sectors/track
*   14 tracks/cylinder
```

```

*   1008 sectors/cylinder
*   2038 cylinders
*   2036 accessible cylinders
*
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition  Tag  Flags      First      Sector      Last
*           Count  Sector      Count      Sector      Mount Directory
*   0         2    00          0      303408      303407      /
*   1         3    01     303408      225792      529199
*   2         5    00          0     2052288     2052287
*   6         4    00     529200     1523088     2052287      /usr
#

```

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、かなら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。この操作は、通常のディスクラベル作成方法を使用して実行できます。つまり、自動構成するか、またはディスクタイプを手作業で指定します。

`format` 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 10 章または第 11 章を参照してください。

2. `format` ユーティリティを使用して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、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 xyzMB
```

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. ディスクラベルの作成方法を決定します。

ディスクが正常に自動構成された場合

ディスクが正常に自動構成されなかった場合

手順 5 と 6 を実行してから、手順 12 に進む。

手順 7 から 11 までを実行してから手順 12 に進む。

5. `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.
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
```

(続く)

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` でバックアップラベルが見つかり、バックアップラベルの内容が適切に表示された場合は、`backup` コマンドを実行し、バックアップラベルを使用してディスクにラベルをつけます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

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 Devices 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` でバックアップラベルが見つかり、その内容が適切な場合は、`backup` コマンドを実行し、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ます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

ディスクラベルが復元されました。

12. `q` と入力して `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 39 章を参照してください。

サードパーティのディスクの追加

Solaris 環境では、サードパーティの多数のディスクがサポートされます。ただし、デバイスドライバ、`format.dat`、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。

サードパーティのディスクが標準の SunOS オペレーティングシステム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サードパーティ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` ユーティリティがサードパーティ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 30 章または第 31 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。Solaris 2.3 からは、再構成ブート時にドライブの電源が入っていれば、`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 ドライブを自動的に構成する手順については、382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いっしょ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。`/etc/format.dat` ファイルにエントリを追加する方法については、第 32 章を参照してください。

▼ 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ります。
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
```

3. 第 32 章で説明する format.dat 情報を使用して、/etc/format.dat ファイルに、サードパーティ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

Solaris 2.3 およびその互換バージョンでは、/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、スライスへの分割、ラベルの作成を行うことができます。

次の手順にしたがって、自動構成によって SCSI ドライブを構成します。

- 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、382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。

次の表に、format がパーティションテーブルの作成に使用する規則を示します。

表 29-2 SCSI ディスクスライスの規則

ディスクサイズ	ルートファイルシステム	スワップスライス
0 ~ 180M バイト	16M バイト	16M バイト
180 ~ 280M バイト	16M バイト	32M バイト
280 ~ 380M バイト	24M バイト	32M バイト
380 ~ 600M バイト	32M バイト	32M バイト
600M ~ 1.0G バイト	32M バイト	64M バイト
1.0 ~ 2.0G バイト	64M バイト	128M バイト
2.0G バイト ~	128M バイト	128M バイト

いずれの場合も、スライス 6 (/usr ファイルシステム) にディスク上の残りの領域が割り当てられます。

1.3G バイトの SCSI ディスクドライブに関して `format` で生成されるパーティションテーブルを示しています。

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の詳細は、第 32 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ります。
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 -g30 -y
```

- i0 システムを init 状態 0 (電源切断) にする。
- g30 ログインしているユーザーに、30 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がシャットダウンされると、ok または > 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。
インストールについての詳細は、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。
システムがブートされ、ログインプロンプトが表示されます。
9. スーパーユーザーとしてログインし、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
```

(続く)

```
Specify disk (enter its number): 1
```

10. 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、自動構成機能によってディスクラベルが生成され、ディスクに書き込まれます。

```
Disk not labeled. Label it now? y
```

11. `verify` コマンドを使用してディスクラベルを確認します。

```
format> verify
```

12. `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、426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、385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 が不良であり、format ユーティリティの repair コマンドを使用して配置し直すか、または修復機能を使用可能にして表面解析を実行する必要があることを示しています。

▼ 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ります。
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。
詳細は、mount (1M) のマニュアルページを参照してください。

```
# umount /dev/dsk/device-name
```

3. format と入力して 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.
```

- format> プロンプトで analyze と入力して、**analyze** メニューに入ります。

```
format> analyze
```

- 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
```

- 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ります。

- format ユーティリティに入り、欠陥セクターの存在するディスクを選択します。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. c0t2d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
    /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@2,0
 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> プロンプトで repair コマンドを入力します。

```
format> repair
```

- 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、385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、SCSI デバイスに対してのみ有効な拡張メッセージと診断メッセージを出力するようにします。

次の例では、Inquiry: の下の一連の数字は、その右側に表示されている inquiry データの 16 進値を表わします。

```
# 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 x y z
> 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.
#
```


SPARC: ディスクの追加 (手順)

この章では、SPARC システム上にディスクを追加する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。

- 393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」
- 394ページの「SPARC: 二次ディスクを接続してブートする方法」
- 396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」
- 400ページの「SPARC: ファイルシステムを作成する方法」
- 401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」

ディスク管理の概要については、第 28 章を参照してください。IA システムにディスクを追加する手順については、第 31 章を参照してください。

SPARC: システムディスクと二次ディスクについて

システムディスクには、ルート (/) ファイルシステム、`/usr` ファイルシステム、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、復元方法には次の 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媒体から復元する。

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で復元できます。

SPARC: システムディスクまたは二次ディスクの追加

表 30-1 SPARC: 作業マップ: システムディスクまたは二次ディスクの追加

作業	説明	手順の説明
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 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>393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」</p> <p>394ページの「SPARC: 二次ディスクを接続してブートする方法」</p>
2. スライスとディスクラベルの作成	<p>まだディスクメーカーによって実行されていない場合は、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。</p>	<p>396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」</p>
3. ファイルシステムの作成	<p>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 (/) と /usr ファイルシステムを作成しなければならない。</p>	<p>400ページの「SPARC: ファイルシステムを作成する方法」</p>

表 30-1 SPARC: 作業マップ: システムディスクまたは二次ディスクの追加 続く

作業	説明	手順の説明
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 (/) と /usr ファイルシステムを復元する。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 44 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。	401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、ディスクの裏側にそのための小型のスイッチが付いています。
3. 交換用の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。
インストールについての詳細は、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** からブートするかに応じて、次の表の手順で操作します。

ブート元	操作
ローカル CD-ROM ドライブ内の Solaris CD から	1. CD が CD-ROM ドライブに入っているかどうかを確認する。 2. CD からシングルユーザーモードでブートする。 ok boot cdrom -s
ネットワーク上の CD-ROM ドライブ内の Solaris CD から	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。 ok boot net -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。

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。396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ります。
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。

このようなディスク用の `format.dat` のエントリの作成方法については、381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 -g30 -y
```

- i0 システムを init 状態 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。

オペレーティング環境がシャットダウンされると、ok または > 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、ディスクの裏側にそのための小型の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。
インストールについての詳細は、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を表示します。

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。396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ります。
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。

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 パーティション分割をし直したいディスクの番号

4. `partition` メニューにアクセスします (スライスを設定できます)。

```
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` スライスについての詳細は、361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、次のどちらか、または両方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)
- `/usr` (スライス 6)

スライスを設定すると、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 `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396ページの手順 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`

新しいパーティションテーブルの名前

12. 新しいディスク上の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. verify コマンドを使用して、ディスクラベルを確認します。

```
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 1G 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
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 c0t1d0
[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 - 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       0             0    (0/0/0)      0
```

(続く)

```

7 unassigned  wm      0          0          (0/0/0)      0
Do you wish to continue creating a new partition
table based on above table[yes]? yes
Free Hog partition[6]? 6
Enter size of partition `0' [0b, 0c, 0.00mb]: 200mb
Enter size of partition `1' [0b, 0c, 0.00mb]: 200mb
Enter size of partition `3' [0b, 0c, 0.00mb]:
Enter size of partition `4' [0b, 0c, 0.00mb]:
Enter size of partition `6' [0b, 0c, 0.00mb]:
Enter size of partition `7' [0b, 0c, 0.00mb]:

Part      Tag      Flag      Cylinders      Size      Blocks
0         root     wm        0 - 406        200.32MB  (407/0/0)  410256
1         swap     wu        407 - 813      200.32MB  (407/0/0)  410256
2         backup  wu        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        814 - 2035     601.45MB  (1222/0/0) 1231776
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 1G 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
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 c0t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:

```

(続く)

```

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 - 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         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 - 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         0                0      (0/0/0)      0
7         unassigned wm         0 - 2035      1002.09MB (2036/0/0) 2052288

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
#

```

次に進む手順

ディスクスライスを作成してディスクのラベル付けが終わったら、ディスク上にファイルシステムを作成します。400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ります。

2. `newfs` コマンドを使用して、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` 作成するファイルシステムの raw デバイス

`newfs` コマンドについての詳細は、第 35 章を参照してください。

3. 未使用のマウントポイントにマウントすることによって、新しいファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz/mnt
# ls
lost+found
```

参照先

追加するディスク	参照先
システムディスク	ディスク上にルート (/) と <code>/usr</code> のファイルシステムを復元する必要がある。第 44 章を参照。 ルート (/) と <code>/usr</code> 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。401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 44 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 36 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ります。

2. `installboot` コマンドを使用して、システムディスクにブートブロックを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

3. システムを実行レベル **3** にして、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、SPARCstation 10 にブートブロックをインストールします。

```
# installboot /usr/platform/sun4m/lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0
```

IA: ディスクの追加 (手順)

この章では、IA 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。この章で説明する手順は次のとおりです。

- 406ページの「IA: システムディスクを接続してブートする方法」
- 407ページの「IA: 二次ディスクを接続してブートする方法」
- 408ページの「IA: Solaris fdisk パーティションを作成する方法」
- 416ページの「IA: ディスクスライスとディスクラベルを作成する方法」
- 418ページの「IA: ファイルシステムを作成する方法」
- 419ページの「IA: 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」

ディスク管理の概要については、第 28 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 30 章を参照してください。

IA: システムディスクと二次ディスクについて

システムディスクには、ルート (/) ファイルシステム、/usr ファイルシステム、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媒体から復元する。

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。

IA: システムディスクまたは二次ディスクの追加

表 31-1 IA: 作業マップ: システムディスクまたは二次ディスクの追加

作業	説明	手順の説明
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 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>406ページの「IA: システムディスクを接続してブートする方法」</p> <p>407ページの「IA: 二次ディスクを接続してブートする方法」</p>
2. スライスとディスクラベルの作成	<p>まだディスクメーカーによって実行されていない場合は、ディスクスライスとディスクラベルを作成する。</p>	<p>408ページの「IA: Solaris fdisk パーティションを作成する方法」</p>
3. ファイルシステムの作成	<p>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。ルート (/) および /usr ファイルシステムは、システムディスク上に作成しなければならない。</p>	<p>418ページの「IA: ファイルシステムを作成する方法」</p>

表 31-1 IA: 作業マップ: システムディスクまたは二次ディスクの追加 続く

作業	説明	手順の説明
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 (/) および /usr ファイルシステムを復元する。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 44 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。	419ページの「IA: 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」

IA: 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 - Solaris スライス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 (Intel 版 Solaris でのみサポートされます) 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するようにしています。

▼ IA: 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。
インストールについての詳細は、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。また、ディスクに固有のハードウェア構成要件については、『Solaris 8 デバイスの構成 (Intel 版)』を参照してください。
4. ローカルまたはリモートの **Solaris CD** からブートする場合は、手順 **a** から **e** までを実行します。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. **Solaris** のインストール **CD** を **CD-ROM** ドライブに挿入します。
 - 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、ブートデバイスとして **CD** または **NET** 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。

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。416ページの「IA: 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。

▼ IA: 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ります。
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 -g30 -y
```

`-i0` システムを `init` 状態 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が始まることを通知する。

`-y` コマンドがユーザーの介入なしで実行されるように指定する。

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. ディスクをシステムに正しく接続します。
インストールについての詳細は、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。また、ディスクに固有のハードウェア構成要件については、『Solaris 8 デバイスの構成 (Intel 版)』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。
システムがブートされ、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。416ページの「IA: 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。

▼ IA: Solaris fdisk パーティションを作成する方法

1. 405ページの「IA: fdisk パーティションの作成上のガイドライン」を必ず読んでください。
2. スーパーユーザーになります。
3. `format` ユーティリティを起動します。

```
# format
```

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

Solaris fdisk パーティションを作成するディスクの番号

5. fdisk メニューに移動します。

```
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。

目的の作業	次の手順	参照
IA: 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	412ページの「IA: 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	413ページの「IA: 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	414ページの「IA: 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、412ページの手順 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 "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:

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

(続く)

```
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。

- 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、411ページの手順9から411ページの手順11を参照してください。

- 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
```

- 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>
```

- format メニューを終了します。

```
format> quit
```

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。416ページの「IA: 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。

IA: 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
```

IA: 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
```

(続く)

```

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

```

IA: 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

```

(続く)

```

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)

```

(続く)

```
Enter Selection: 4
format> q
```

▼ IA: 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ります。
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、パーティションを変更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 パーティションを変更したいディスクの番号

4. `partition` メニューに移動します (スライスを設定できます)。

```
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 スライスについての詳細は、361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) (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。

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 416ページの手順 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

新しいパーティションテーブルの名前

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. `verify` コマンドで、新しいディスクラベルを確認します。

```
format> verify
```

15. `format` メニューを終了します。

```
format> quit
```

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。418ページの「IA: ファイルシステムを作成する方法」を参照してください。

▼ IA: 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ります。
2. `newfs` コマンドを使用して、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
```

newfs コマンドについての詳細は、第 36 章を参照してください。

3. 未使用のマウントポイントにマウントすることによって、新しいファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls /mnt
lost+found
```

次に進む手順

追加するディスク	参照先
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 44 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。419ページの「IA: 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 44 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完了することになる。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 36 章を参照。

▼ IA: 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ります。
2. 次のコマンドを入力します。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i`  
/lib/fs/ufs/bootblk /dev/rdisk/cwtxdys2
```

/usr/platform/`uname -i`/lib/fs/ ufs/pboot	パーティションのブートファイル
/usr/platform/`uname -i`/lib/fs/ ufs/bootblk	ブートブロックコード
/dev/rdisk/cwtxdys2	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
```

IA: 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
```

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。

- 421ページの「format ユーティリティを使用するための要件または制限」
- 422ページの「format のメニューとコマンドの説明」
- 429ページの「format で使用されるファイル — format.dat」
- 437ページの「format に関連するマニュアルページ」
- 435ページの「format コマンドへの入力規則」

format ユーティリティの概要については、第 28 章を参照してください。

format ユーティリティを使用するための要件または制限

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らなければなりません。スーパーユーザーでない場合は、format を使用しようとするときのエラーメッセージが表示されます。

```
% format Searching for disk...done
No permission (or no disk found)!
```

format を使用する場合に情報を保存するための推奨事項

- 他の作業を実行する前に、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とります。
- format の dump コマンドを使用して、欠陥の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、欠陥の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>
```

表 32-1 に、format のメインメニュー項目を示します。

表 32-1 format のメインメニュー項目の説明

項目	コマンド/メニュー	機能
disk	コマンド	後の操作に使用するディスクを選択する (現在のディスクと呼ぶ)。システムのすべてのドライブが表示される。
type	コマンド	現在のディスクのメーカーとモデルを選択する。認識されているドライブタイプのリストが表示される。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	スライスを作成したり変更したりする。詳細は、424 ページの「partition メニュー」を参照。
current	コマンド	現在のディスクに関して次の情報が表示される。 <ul style="list-style-type: none"> ■ デバイス名とタイプ ■ 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にプロンプトに回答した情報
fdisk	メニュー	fdisk プログラムが実行され、Solaris fdisk パーティションを作成する。
repair	コマンド	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、426 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、428 ページの「defect メニュー」を参照。
backup	コマンド	バックアップラベルを検索する。

表 32-1 format のメインメニュー項目の説明 続く

項目	コマンド/メニュー	機能
verify	コマンド	ディスクに関して次の情報を出力する。 <ul style="list-style-type: none"> ■ デバイス名とタイプ ■ 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	新しいディスクとパーティションの情報を保存する。
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。
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>
```

表 32-2 に、partition メニューの項目を示します。

表 32-2 partition メニューの項目の説明

コマンド	機能
change 'x' partition	以下の値を設定し、新しいスライスを指定する。 <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	あらかじめ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が表示される。
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。
quit	partition メニューを終了する

IA: fdisk メニュー

IA 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)
```

(続く)

```
Enter Selection:
```

表 32-3 に、fdisk メニューの項目を示します。

表 32-3 fdisk メニューの項目の説明

コマンド	機能
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	どのパーティションをブートに使用するかを指定する。これによって、第 1 段階のブートプログラムが実行する第 2 段階のブートプログラムの存在するパーティションを指定する。
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。
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)
```

(続く)

```

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>

```

表 32-4 に、analyze メニューの項目を示します。

表 32-4 analyze メニューの項目の説明

コマンド	機能
read	こ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ディスク上で読み込んで書き込む。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、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。
purge	ディスク上のデータを消去する。いかなる手段でも取り出せないように、ディスクに複数のパターンを書き込むことによって、すべてのデータを削除する。ディスク全体(またはディスクのセクション)に3種類のパターンが書き込まれ、検査に合格すると16進のビットパターンが書き込まれて、データが削除される。 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み、2度目にそのデータを検査する。ディスク上の既存のデータは失われる。デフォルトで欠陥ブロックを修復する。

表 32-4 analyze メニューの項目の説明 続く

コマンド	機能
print	読み込み/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/m) 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
```

(続く)

```

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>

```

表 32-5 に defect メニューの項目を示します。

表 32-5 defect メニューの項目の説明

コマンド	機能
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んで、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 (analyze メニューから検出された欠陥) 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 で使用されるファイル — format.dat

format で使用されるデータ、/etc/format.dat の内容は次のとおりです。

- ディスクタイプ
- デフォルトのスライステーブル

Solaris オペレーティングシステムといっしょに出荷される 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、そのエントリを追加するか、format で処理を実行中に必要な情報の入力を促すプロンプトを表示させることができます。

ディスクドライブをサイト全体で使用する場合は、format.dat ファイルにエントリ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。

次の場合には、システムのデータ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 - デフォルトエントリは変更しないでください。デフォルトエントリを変更したい場合は、混乱を避けるために、そのエントリをコピーし、別の名前を付けて変更します。

format.dat ファイルの構造

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

データファイルには、次の構文規則が適用されます。

- シャープ記号(#)はコメント文字です。シャープ記号に続く1行のテキストは、format で解釈されません。
- 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slash (\) 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。
- 空白は、format に無視されます。代入値に空白を含めたい場合は、値全体を二重引用符 (") で囲みます。これによって、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。値はカンマ (,) で区切ります。

format.dat ファイル中のキーワード

データファイルには、起動時に format に読み込まれるディスク定義が入っています。各定義は、キーワード search_path、disk_type、または partition で始まります。表 32-6 を参照してください。

表 32-6 format.dat のキーワードの説明

キーワード	用途
search_path	このキーワードは format.dat ファイルでは使用しない。Solaris 2.0 リリース以降は、format ユーティリティは論理デバイス階層 (/dev) を検索するので、このキーワードを設定してシステムのディスクを検索する必要がなくなった。
disk_type	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。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。
partition	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 内で参照できる名前が入っている。デフォルトのデータファイルには、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入っている。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。

ディスクタイプ (format.dat)

disk_type 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入っています。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ます。サポートされないディスクを使用する場合に限り、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 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。表 32-7 に、すべての disk_type 定義でキーワードの他に割り当てなければならない識別子を示します。

表 32-7 必須の disk_type 識別子

識別子	説明
ctrlr	ディスクタイプに有効なコントローラのタイプ。現在、この代入式に有効な値は SCSI と ISP-80 (IPI コントローラ) 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 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 つはシリンダを残しておく必要がある。
pcyl	ディスクタイプ内の物理シリンダ数。この数は、ディスク媒体の境界を計算するために使用される。通常、この数値は ncyl と acyl の合計に等しくなるが、そうでない場合もある。
nhead	ディスクタイプ内のヘッド数。この数値は、ディスク媒体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、ディスク媒体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、スペアは割り当てには含まれない。
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、他の代入式が必要な場合があります。表 32-8に、SCSI コントローラに必要な代入式を示します。

表 32-8 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 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	このパラメータに代入した数値は、所定の欠陥領域内で代替マッピングに利用できる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" \  
: ctlr = SCSI : fmt_time = 4 \  
: trks_zone = 17 : asect = 6 : atrks = 17 \  
: ncyl = 1965 : acyl = 2 : pcyl = 3500 : nhead = 17 : nsect = 80 \  
: rpm = 5400 : bpt = 44823  
  
disk_type = "SUN2.1G" \  
: ctlr = SCSI : fmt_time = 4 \  
: ncyl = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \  
: rpm = 5400 : bpt = 44823  
  
disk_type = "SUN2.9G" \  
: ctlr = SCSI : fmt_time = 4 \  
: ncyl = 2734 : acyl = 2 : pcyl = 3500 : nhead = 21 : nsect = 99 \  
: rpm = 5400
```

パーティションまたはスライステーブル (format.dat)

partition 定義のキーワードが、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。表 32-9 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 32-9 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 <code>disk_type</code> の名前。この名前は <code>disk_type</code> 内で使用されるとおりに指定しなければならない。
ctrlr	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、この代入式に有効な値は IPI コントローラを表す <code>ISP-80</code> と SCSI コントローラを表す <code>SCSI</code> である。ここで指定したコントローラタイプは、上記で選択した <code>disk_type</code> にも定義しなければならない。

スライス定義内の他の代入式では、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。これらの代入式は省略可能です。明示的に代入されていないスライスは、長さ 0 に設定されます。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" : ctrlr = SCSI \
: 0 = 0, 34000 : 1 = 25, 133280 : 2 = 0, 2672400 : 6 = 123, 2505120

partition = "SUN2.1G" \
: disk = "SUN2.1G" : ctrlr = SCSI \
: 0 = 0, 62320 : 1 = 41, 197600 : 2 = 0, 4154160 : 6 = 171, 3894240

partition = "SUN2.9G" \
: disk = "SUN2.9G" : ctrlr = SCSI \
: 0 = 0, 195426 : 1 = 94, 390852 : 2 = 0, 5683986 : 6 = 282, 5097708
```

format データファイルの位置を指定する

format ユーティリティは、次の方法でデータファイルの位置を認識します。

1. `-x` コマンド行オプションでパス名を指定した場合は、その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は、format 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ければ、format 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、必ず存在するはずです。

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、437ページの「format のヘルプを使用する」を参照してください。

format コマンドへ数値を入力する

format では、整数を入力しなければならない場所が数カ所あります。データを指定するか、選択肢のリストから選択しなければなりません。どちらの場合も、help 機能を使用すると、format は期待する整数の上限と下限を表示し、目的の数値を入力するだけですみます。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。任意の底の整数を指定できますが、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 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す。

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、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、媒体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ければ、該当する値は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 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 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 コマンドヘディスク名を指定する

format 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 (") で囲まなければなりません。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。

format のヘルプを使用する

format ユーティリティにはヘルプ機能が組み込まれており、入力が必要なときに使用できます。疑問符 (?) を入力するだけで必要な情報に関するヘルプが表示され、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 に関連するマニュアルページ

format ユーティリティに関連するマニュアルページは、format (1M) と format.dat (4) です。format (1M) では、format ユーティリティの基本機能とコマンド行で使用できるすべてのオプションについて説明されています。

format.dat (4) では、format ユーティリティで使用するディスクドライブ構成情報について説明されています。

ファイルシステムの管理

ここでは、Solaris オペレーティング環境でファイルシステムを管理する方法について説明します。次の章で構成されています。

第 34 章	ファイルシステムの種類、一般に使用する管理コマンド、ファイルシステムをマウントし、マウント解除する基本的な作業など、ファイルシステム全般について説明します。
第 35 章	UFS ファイルシステムの作成、一時ファイルシステム (TMPFS) の作成と管理、ループバックファイルシステム (LOFS) の作成を行う手順について説明します。
第 36 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 37 章	キャッシュファイルシステム (CacheFS™) の概要と使用手順について説明します。
第 38 章	スワップ空間を追加する場合の構成、スワップ資源の監視、スワップファイルの作成と使用可能にする方法、余分なスワップ空間を削除する手順などについて説明します。

第 39 章

ファイルシステムの状態を記録する方法、`fsck` プログラムでチェックされる内容、自動ブートチェック機能を変更する方法、`fsck` プログラムの使用方法について説明します。

第 40 章

ルート(/)と `/usr` のファイルシステムのデフォルトディレクトリ、`/kernel` ディレクトリにあり、`mkfs` コマンドと `newfs` コマンドに固有のディレクトリなど、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)

この章の内容は次のとおりです

- 441ページの「ファイルシステムにおける新機能」
- 448ページの「ファイルシステムのタイプ」
- 453ページの「ファイルシステム管理コマンド」
- 455ページの「デフォルトの Solaris ファイルシステム」
- 456ページの「スワップ空間」
- 457ページの「UFS ファイルシステム」
- 459ページの「ファイルシステムのマウントとマウント解除」
- 465ページの「ファイルシステムのタイプを調べる」

ファイルシステムにおける新機能

この節では、ファイルシステムの新機能について説明します。

`/var/run` ファイルシステム

新しい TMPFS マウントのファイルシステム `/var/run` は、一時的な (つまり、システムをリブートした場合に残る必要がない) ファイルシステム用のリポジトリです。今回のリリースから Solaris で採用されています。従来の `/tmp` ディレクトリもシステムファイル以外の一時的なファイル用のリポジトリとして存続します。

`/var/run` は、ディスクベースのファイルシステムではなく、メモリーベースのファイルシステムとしてマウントされます。このディレクトリが更新されても不必要なディスクトラフィックが発生しないため、システムは干渉されずに電源管理ソフトウェアを実行し続けることができます。

`/var/run` ディレクトリは管理が不要です。`umount -a` コマンドや `umountall` コマンドでもマウントは解除されません。

セキュリティ上の理由から、`/var/run` の所有者は `root` です。

マウントテーブル (`/etc/mnttab`) の変更

以前の Solaris リリースでは、`/etc/mnttab` はテキストベースのファイルであり、マウントされているファイルシステムについての情報を格納していました。このようにテキストファイルにすると、マウントされているファイルシステムの実際の状態が反映されないという欠点がありました。

今回のリリースの Solaris では、`/etc/mnttab` は MNTFS ファイルシステムであり、マウントされているファイルシステムについての (読み取り専用の) 情報をカーネルからローカルシステムに直接提供します。

次に、`mnttab` の動作の変更点を示します。

- `/etc/mnttab` に書き込もうとするプログラムまたはスクリプトは失敗します。
- `mnttab` エントリに似せている `mount -m` オプションは機能しません。

`/etc/mnttab` マウントテーブルは管理が不要です。

詳細は、`mnttab(4)` のマニュアルページを参照してください。

UDF (Universal Disk Format) ファイルシステムの使用

今回のリリースの Solaris には、DVD (Digital Versatile Disc または Digital Video Disc) 光学式媒体に情報を格納するための業界標準形式である UDF ファイルシステムが含まれています。

UDF ファイルシステムは、SPARC と IA 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 ファイルシステムのサポートは、次の新しいパッケージで提供されます。

- SUNWudfr — 32 ビットのカーネル構成要素
- SUNWudfrx — 64 ビットのカーネル構成要素
- SUNWudf — /usr の構成要素

UDF の特徴と利点

今回のリリースの Solaris では、UDF ファイルシステムには次のような特徴があります。

- 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 の媒体にアクセスできます。
- プラットフォームやオペレーティングシステムを超えて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。

次の特徴は、今回のリリースの UDF ファイルシステムには含まれていません。

- CD-RW (何度でも書き換えが可能な CD 媒体) と DVD-RAM は順次ディスク同時記録方式と増分記録方式の両方ともサポートされません。
- 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の UFS 構成要素は UDF 1.50 仕様の一部ではないため、サポートされません。

ハードウェアとソフトウェアの要件

次に、UDF ファイルシステムの要件を示します。

- Solaris 7 11/99 または Solaris 8 リリース
- サポートされている SPARC または Intel のプラットフォーム
- サポートされている CD-ROM または DVD-ROM のドライブ

UDF の互換性について

このリリースの Solaris で初めて実装された UDF ファイルシステムには、次のような互換性があります。

- 業界標準の読み書き 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

▼ DVD-ROM デバイスを接続する方法

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. システムをシャットダウンして、電源を切ります。

```
# init 0
```

4. DVD-ROM デバイスを接続します。
5. システムの電源を入れます。

▼ DVD-ROM デバイス上のファイルにアクセスする方法

1. DVD-ROM デバイスが自動的にマウントされていることを確認します。

```
$ ls /cdrom
```

注 - CD-ROM と DVD-ROM の両方のデバイスがシステムに接続されている場合、CD-ROM の名前が /cdrom/cdrom0 となり、DVD-ROM の名前が /cdrom/cdrom1 となります。DVD-ROM だけがシステムに接続されている場合は、/cdrom/cdrom0 を使用してください。

2. `ls` コマンドで **DVD-ROM** の内容を表示します。

```
$ ls /cdrom/cdrom1
Copyright filea fileb
```

現在のところ、CDE ファイルマネージャによる自動表示は実装されていません。CDE ファイルマネージャのその他の機能 (ドラッグ&ドロップによるコピーや `imagetool` の機能など) はすべて利用できます。

▼ UDF ファイルシステムのパラメータを表示する方法

UDF ファイルシステムのパラメータを表示するには、`mkfs` 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. **UDF** ファイルシステムのパラメータを表示します。

```
# mkfs -F udfs -m /dev/rdisk/device-name
```

▼ UDF ファイルシステムを作成する方法

UDF ファイルシステムを作成するには、`mkfs` 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. **UDF** ファイルシステムを作成します。

```
# mkfs -F udfs /dev/rdisk/device-name
```

3. **UDF** ファイルシステムをマウントして、**UDF** ファイルシステムが作成されていることを確認します。詳細は、446ページの「UDF ファイルシステムをマウントする方法」を参照してください。

詳細は、`mkfs_udfs(1M)` のマニュアルページを参照してください。

▼ UDF ファイルシステムのタイプを識別する方法

UDF ファイルシステムのタイプを識別するには、`fstyp` 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. ファイルシステムが **UDF** ファイルシステムであるかどうかを確認します。

```
# fstyp -v /rdev/dsk/device-name
```

▼ UDF ファイルシステムを検査する方法

UDF ファイルシステムの完全性を検査するには、`fsck` 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. **UDF** ファイルシステムを検査します。

```
# fsck -F udfs /dev/rdisk/device-name
```

詳細は、`fsck_udfs(1M)` のマニュアルページを参照してください。

▼ UDF ファイルシステムをマウントする方法

UDF ファイルシステムをマウントするには、次のようにします。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムをマウントします。

```
# mount -F udfs /dev/dsk/device-name /mount-point
```

3. **UDF** 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。

```
# ls /mount-point
```

詳細は、`mount_udfs(1M)` のマニュアルページを参照してください。

▼ UDF ファイルシステムのマウントを解除する方法

UDF ファイルシステムのマウントを解除するには、次のようにします。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムのマウントを解除します。

```
# umount /dev/dsk/device-name
```

▼ UDF ファイルシステムのデバイスにラベルとボリューム名を作成する方法

UDF ファイルシステム用のファイルシステム名 (ラベル) とボリューム名を作成するには、次のようにします。

1. スーパーユーザーになります。
2. **UDF** ファイルシステム用のファイルシステム名とボリューム名を作成します。

```
# labelit -F UDFS /dev/rdsk/device-name fsname volume
```

詳細は、`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」という用語には、さまざまな使用方法があります。

- 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下位への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媒体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、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 アーキテクチャによって、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ます。また、新しいファイルシステムの追加が容易になります。

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。この章を読んで概念と計画方法について理解してください。次の作業については、下記の該当箇所を参照してください。

作業内容	参照
新しい UFS と仮想ファイルシステムを設定する	第 35 章 と 第 37 章
ローカルファイルとリモートファイルを利用できるようにする	第 36 章
新しい記憶デバイスを接続して構成する	第 28 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 42 章
ファイルシステムの損傷をチェックして訂正する	第 39 章

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、465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媒体に格納されます。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	説明
UFS	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。 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ディスクのスライスへの分割についての詳細は、第 28 章を参照してください。
HSFS	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。
PCFS	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/書き込みのアクセスができます。
UDF	UDF ファイルシステム。DVD (Digital Versatile Disc または Digital Video Disc) と呼ばれる光学式媒体テクノロジーに情報を格納するための業界標準形式。

ディスクベースの各種ファイルシステムは、次のように特定の媒体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、ネットワーク上で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す。NFS は、ネットワークベースまたは分散コンピューティングで利用できる唯一のファイルシステムです。

NFS で分散資源(ファイルやディレクトリ)を管理するには、サーバーから分散資源をエクスポートして個々のクライアントシステムでそれらをマウントします。詳細は、463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、キャッシュファイルシステム(CacheFS)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。

キャッシュファイルシステム

キャッシュファイルシステム(CacheFS)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。CacheFS ファイルシステムの設定と管理については、第 37 章を参照してください。

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。

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上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 /tmp 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がいっぱいになるの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。

TMPFS ファイルシステムの作成方法については、第 35 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 38 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 上に複写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 35 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ます。PROCFS の /proc ディレクトリには、有効なプロセスのプロセス番号別リストが入っています。/proc ディレクトリ内の内容は、ps などのコマンドに使用されます。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、システム管理が不要です。

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
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。表 34-1 に、汎用ファイルシステム管理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 34-1 汎用ファイルシステム管理コマンド

コマンド	機能
<code>clri (1M)</code>	i ノードをクリアする。
<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 (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 (1M)</code>	ファイルシステムの完全性をチェックし、検出された損傷を修復する。
<code>fsdb (1M)</code>	ファイルシステムをデバッグする。
<code>fstyp (1M)</code>	ファイルシステムのタイプを調べる。
<code>labelit (1M)</code>	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 (<code>volcopy</code> コマンド専用)。
<code>mkfs (1M)</code>	新しいファイルシステムを作成する。
<code>mount (1M)</code>	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。
<code>mountall (1M)</code>	仮想ファイルシステムテーブ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。 (<code>/etc/vfstab</code>)
<code>ncheck (1M)</code>	パス名とその i 番号のリストを生成する。
<code>umount (1M)</code>	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。

表 34-1 汎用ファイルシステム管理コマンド 続く

コマンド	機能
umountall (1M)	仮想ファイルシステムテーブ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。(/etc/vfstab)
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. -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
2. 特殊デバイスを /etc/vfstab 内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 内で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 内で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(1M): System Administration Commands*』を参照してください。専用コマンドのマニュアルページは、汎用コマンドのマニュアルページに続いて入っています。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、**man mount_ufs(1M)** と入力します。

デフォルトの Solaris ファイルシステム

Solaris 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。表 34-2 に、デフォルトの Solaris ファイルシステムの概要と、各ファイルシステムのタイプを示します。

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。デフォルトディレクトリのリストについては、第 40 章を参照してください。

表 34-2 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 ディレクトリに入っている (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 に入っている。
/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 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上では、/home はローカルディスク上の UFS ファイルシステムの場合がある。

表 34-2 デフォルトの Solaris ファイルシステム 続く

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
/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。システムによっては、/opt が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たびにまたは /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

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ことは大切です。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 38 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になります。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	UFS ファイルシステムの最大サイズは 1T バイト (テラバイト) です。Solaris オペレーティング環境では、論理スライスの大きさを 1T バイトのファイルシステムに対応させるストライプ機能はサポートされていませんが、Sun 提供の Solstice DiskSuite™ ソフトウェアには、この機能が含まれています。
大規模ファイル	デフォルトでは、UFS ファイルシステムは 2G バイト (ギガバイト) を超える通常ファイルを持つことができます。2G バイトの最大ファイルサイズ制限を有効にするには、 <code>nolargefiles</code> マウントオプションを明示的に使用しなければなりません。この制限は Solaris 2.6 リリースでなくなりました。

UFS ファイルシステムの詳細は、第 40 章を参照してください。

UFS ファイルシステムの構成

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制御され、編成されます。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。

`newfs` コマンドまたは `mkfs` コマンドの引数を使用してファイルシステムをカスタマイズするには、これらのパラメータの変更方法について第 40 章を参照してください。

UFS ロギング

UFS ロギングは、トランザクション (完全な UFS 操作を構成する変更) 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す。保存されたトランザクションは、後でファイルシステムに適用できます。

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。まず、ファイルシステムの整合性が保持されるため、`fsck(1M)` を実行する必要がなくなります。また、`fsck` を省略できるため、システムがクラッシュしたり、異常停止 (クリーンではない停止) した場合でもシステムをリブートするのに要する時間を短縮できます。クリーンではない停止についての詳細は、554 ページの「`fsck` でチェックして修復される内容」を参照してください。UFS ロギングを使用すれば、特に、通常であれば `fsck` による読み込みと確認に時間がかかる大規模ファイルシステムを持つシステムのブート時間をかなり短縮できます。

UFS ロギングが作成するログは、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 -f` コマンドを実行したときにも、ログはすべてフラッシュされます。

デフォルトでは、UFS ロギングは無効です。UFS ロギングを有効にするには、ファイルシステムをマウントするときか `/etc/vfstab` ファイル内で `-o logging` オプションを指定して `mount(1M)` コマンドを実行しなければなりません。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、ルート (/) ファイルシステムを含む、任意の UFS 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オプションが追加されています。

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なければなりません。推奨事項は次の通りです。

- 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 を分散させたり、ディレクトリを入れ替えたりして、異なるディスクに負荷を均等に割り当てます。
- 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、/、/usr、スワップ空間の3つのファイルシステムがあります。他のディスク上では、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。多数の小型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 で複数のボリュームを処理できるようにしておけば、大型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ように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、新しいファイルシステムを作成するときの前提条件と手順については、第 35 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

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。ファイルシステムのマウントとは、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することができます。

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。

次の例では、ルート (/) ファイルシステムの下に、サブディレクトリ `sbin`、`etc`、および `opt` を持つローカルのファイルシステムを示します。

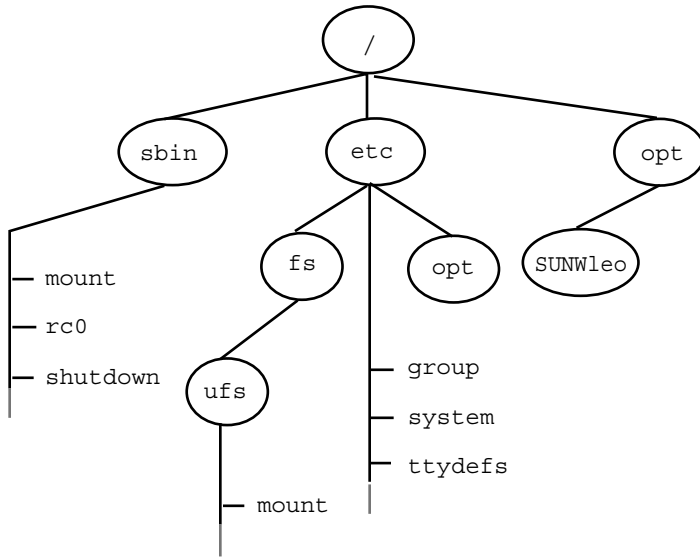


図 34-1 ルート (/) ファイルシステムの例

ここでは、標準添付されていない製品セットが含まれている `/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ものとします。

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 (図 34-2 を参照)。これらの作業を実行するための詳細な手順については、第 36 章を参照してください。

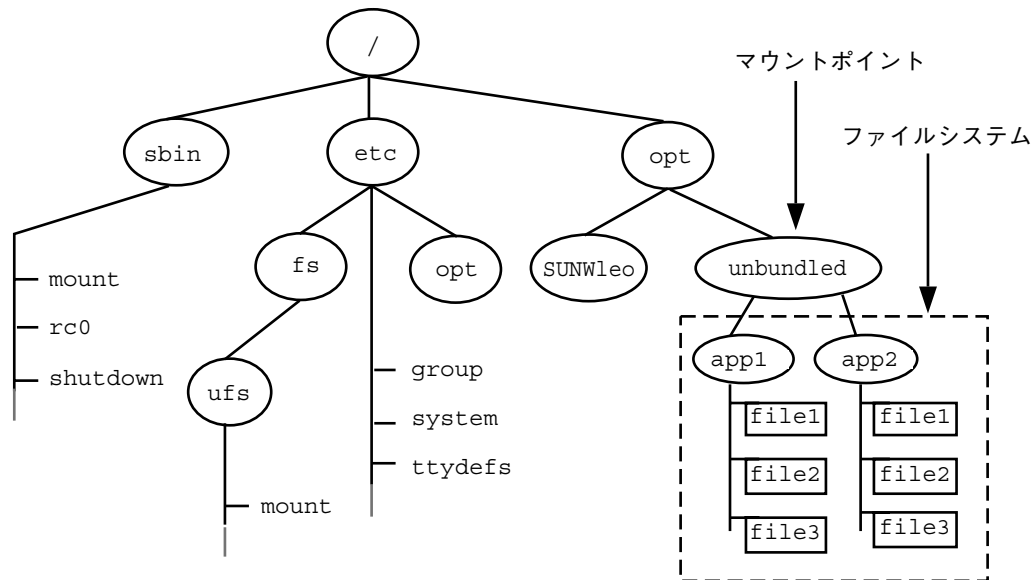


図 34-2 ファイルシステムのマウント

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、`/etc/vfstab` ファイルのように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 938557529
$

```

仮想ファイルシステム

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

システムテーブル (/etc/vfstab ファイル) が作成されました。このファイルに、マウントするファイルシステムとそのマウント方法が指定されます。/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より可能になります。

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 (1M) 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 ファイルです。

```

$ more /etc/vfstab
#device          device          mount          FS          fsck  mount  mount
#to mount        to fsck         point          type        pass   at boot options
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /
/proc            -               /proc         proc        -      no     -
/dev/dsk/c0t0d0s1 -               -             swap       -      no     -
swap            -               /tmp          tmpfs      -      yes    -
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /usr          ufs        2      no     -
/dev/dsk/c0t3d0s7 /dev/rdisk/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 ファイルシステムを自動的にマウントポイント /test にマウントすることを指定しています。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、このファイルの編集方法、使用方法については、第 36 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) の資源 (ファイルやディレクトリ) をネットワーク経由で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です。たとえば、サードパーティ (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、資源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。共通に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経由で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つかいません。

システムは、ネットワークに対して共有するファイルシステムを持っているときに、NFS サーバーになります。サーバーは、現在共有されている資源とアクセス制限 (読み取り/書き込み、読み取り専用など) のリストを管理します。

資源を共有する場合は、リモートシステムにマウントできるように、その資源を使用可能な状態にします。

資源を共有するには、次の方法があります。

- `share(1M)` コマンドまたは `shareall(1M)` 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、システムをリブートする。

資源を共有する方法については、第 36 章を参照してください。NFS についての詳細は、『*Solaris* のシステム管理 (第 3 巻)』を参照してください。

自動マウント (AutoFS)

NFS ファイルシステム資源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 資源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、ファイルシステム資源はマウントされたままになります。資源が一定の時間アクセスされなかった場合、資源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 資源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 資源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 資源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 (1M) によって初期化され、automount はシステムのブート時に自動的に実行されます。自動マウントデーモン automountd (1M) 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、Solaris オペレーティング環境は /home を自動マウントします。

AutoFS は、ネームサービスに指定されたファイルシステムに対して動作します。この情報は、NIS、NIS+、またはローカルの /etc 内のファイルで管理されます。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。各サーバーに優先度を示す数値を割り当てることによって、マップ内の各資源ごとにサーバーの優先順位を指定できます。

AutoFS の設定と管理方法については、『Solaris のシステム管理 (第 3 巻)』を参照してください。

キャッシュファイルシステム (CacheFS)

NFS または 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、キャッシュファイルシステム (CacheFS) を使用します。CacheFS は、サーバーとネットワークの負荷を軽減することによって、NFS サーバーのパフォーマンスとスケーラビリティを向上させる汎用ファイルシステムキャッシュ機構です。

CacheFS 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 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 と AutoFS サービスを併用すると、さらにパフォーマンスとスケーラビリティを向上することができます。

CacheFS についての詳細は、第 37 章を参照してください。

マウント方法の決定

表 34-3 に、ファイルシステムの用途に応じてファイルシステムをマウントするためのガイドラインを示します。

表 34-3 ファイルシステムのマウント方法の決定

マウントするファイルシステムとマウントの頻度	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。■ AutoFS により、ユーザーがディレクトリに移動したとき (マウント) あるいはディレクトリから移動したとき (マウント解除) 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。 パフォーマンスを向上させるには、CacheFS を使用してリモートのファイルシステムをキャッシュすることもできる。

ファイルシステムを含む CD-ROM は、単にドライブに挿入するだけでマウントできます (ボリューム管理が自動的にマウントします)。ファイルシステムを含むフロッピーディスクは、ドライブに挿入して `volcheck` コマンドを実行することによってマウントできます。詳細は、第 14 章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関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
```

<code>mount-point</code>	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、 <code>/var</code> ディレクトリ
<code>fs-table</code>	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指定する。ファイルシステムがマウントされている場合、 <code>fs-table</code> は <code>/etc/mnttab</code> 。マウントされていない場合、 <code>fs-table</code> は <code>/etc/vfstab</code> 。

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(1M)` 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)。

例 — ファイルシステムのタイプを調べる

次の例は、`/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/rdsk/c0t3d0s6 /export ufs 2 yes -
$
```

次の例は、`/etc/mnttab` ファイルを使用して、現在マウントしている (ボリューム管理でマウントした) フロッピーディスク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
$
```


ファイルシステムの作成 (手順)

この章では、UFS、TMPFS、および LOFS ファイルシステムを作成する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「作成」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。

- 471ページの「UFS ファイルシステムを作成する方法」
- 473ページの「TMPFS ファイルシステムを作成する方法」
- 475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取り外し可能な媒体上に作成する手順については、第 14 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に作成するためには、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についての詳細は、第 28 章を参照してください。

Solstice DiskSuite などの論理ボリューム管理製品は、より精練された(単一のスライスや単一のディスク境界を拡張する)メタデバイスを作成します。メタデバイスについての詳細は、『Solstice DiskSuite 4.2.1 ユーザーズガイド』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 (1M)` 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 (1M)` の使いやすいフロントエンドで、新しいファイルシステムを作成します。Solaris システムの場合、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のような `newfs (1M)` のパラメータは、新しいファイルシステムを作成するディスクのラベルから読み込まれ、ユーザーが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。

ファイルシステムパラメータ

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。表 35-1 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。

表 35-1 `newfs` コマンドで使用するデフォルトのパラメータ

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

表 35-1 newfs コマンドで使用するデフォルトのパラメータ 続く

パラメータ	デフォルト値
最小空き領域	((64M バイト/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。
回転遅延	0
最適化のタイプ	容量
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

- 次の前提条件を満たしているかどうかを確認します。
 - UFS ファイルシステムを作成する前に、ディスクをフォーマットして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の詳細は、第 28 章を参照してください。
 - 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 29 章を参照してください。
 - 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。
 - スーパーユーザーでなければならない。
- ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name
```

-N	<code>newfs</code> が <code>mkfs</code> 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。 <code>newfs</code> コマンドをテストするのに好ましい方法です。
-b <i>size</i>	ファイルシステムのブロックサイズを指定する。ブロックごとに 4096 バイトまたは 8192 バイト。デフォルトは 8192 バイトです。
-i <i>bytes</i>	i ノード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。詳細は <code>newfs(1M)</code> のマニュアルページを参照。
<i>device-name</i>	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 - 次の手順を実行する前に、スライスのデバイス名を正しく指定し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、`fsck(1M)` コマンドを使用して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 新しいファイルシステムを格納するディスクデバイス名を指定する。

`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、問題が検出された場合には、問題を修復する前にプロンプトを表示します。`fsck` についての詳細は、第 39 章を参照してください。

例 — UFS ファイルシステムを作成する

次の例では、`/dev/rdisk/c0t1d0s7` 上に UFS ファイルシステムを作成します。


```
# 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

ファイルシステムをマウントし、使用可能にするには、第 36 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のメモリーを使用します。そのため、一時ファイルシステムは、UFS ファイルシステムに比べはるかに高速です。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資源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 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ります。
2. 必要に応じて、**TMPFS** ファイルシステムをマウントしたいディレクトリを作成し、アクセス権と所有権を設定します。

3. TMPFSファイルシステムを作成します。

ブート時に一時ファイルシステムを自動的に作成するようにシステムを設定するには、474ページの「例 — ブート時に TMPFS ファイルシステムを作成する」を参照してください。

```
# mount -F tmpfs -o size=number swap mount-point
```

`-o size=number` TMPFS ファイルシステムのサイズを M バイト単位で指定する。

`mount-point` 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指定する。

4. mount コマンドからの出力を調べて、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、新しいディレクトリ `/export/reports` を作成し、そのマウントポイントに TMPFS ファイルシステムをマウントして、50M バイトに制限します。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50 swap /export/reports
```

例 — ブート時に TMPFS ファイルシステムを作成する

ブート時にシステムが自動的に TMPFS ファイルシステムを作成するようにするには、`/etc/vfstab` ファイルにエントリを追加します。次の例は、システムのブート時に TMPFS ファイルシステムを `/export/test` に作成する `/etc/vfstab` ファイルのエントリを示します。 `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資源によって制限されます。

```
swap - /export/test tmpfs - yes -
```

/etc/vfstab ファイルについての詳細は、482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 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ります。
2. **LOFS** ファイルシステムをマウントしたいディレクトリを作成し、適切なアクセス権と所有権を設定します。
3. **LOFS** ファイルシステムを作成します。
ブート時にループバックファイルシステムを自動的に作成するようにシステムを設定するには、476ページの「例 — ブート時に LOFS ファイルシステムを作成する」を参照してください。

```
# 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。

4. `mount` コマンドからの出力を調べて、**LOFS** 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 — LOFS ファイルシステムを作成する

次の例は、新しいソフトウェアを、実際にはインストールせずに、ループバックファイルシステムとして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/usr/local  
# chroot /tmp/newroot command
```

例 — ブート時に LOFS ファイルシステムを作成する

ブート時にシステムが自動的に LOFS ファイルシステムを作成するようにするには、`/etc/vfstab` ファイル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` に作成する `/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作成できません。

`/etc/vfstab` ファイルの詳細は、482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。

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。

- 482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」
- 484ページの「`/etc/vfstab` ファイルにエントリを追加する方法」
- 486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`)」
- 486ページの「`/etc/vfstab` ファイル内に列挙されているすべてのファイルシステムをマウントする方法」
- 488ページの「UFS ファイルシステムをマウントする方法」
- 491ページの「NFS ファイルシステムをマウントする方法」
- 492ページの「IA: System V (S5FS) ファイルシステムをマウントする方法」
- 493ページの「IA: ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法」
- 495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」
- 496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」
- 497ページの「すべてのファイルシステムをマウント解除する方法 (`/etc/vfstab`)」

ファイルシステムのマウント

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。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。

表 36-1 に、ファイルシステムの用途に応じてファイルシステムをマウントするためのガイドラインを示します。

表 36-1 ファイルシステムのマウント方法の決定

マウントするファイルシステムとマウントの頻度	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。■ <code>AutoFS</code> により、ユーザーがディレクトリに移動したとき (マウント) あるいはディレクトリから移動したとき (マウント解除) 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。 パフォーマンスを向上させるには、 <code>CacheFS</code> を使用してリモートのファイルシステムをキャッシュすることもできる。

ファイルシステムを含む CD-ROM は、単にドライブに挿入するだけでマウントできます (ボリューム管理が自動的にマウントします)。ファイルシステムを含むフロッピーディスクは、ドライブに挿入して `volcheck(1)` コマンドを実行することによってマウントできます。詳細は、第 14 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 36-2 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 36-2 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	説明
<code>mount (1M)</code>	ファイルシステムとリモート資源をマウントする。
<code>mountall (1M)</code>	<code>/etc/vfstab</code> ファイル内で指定されたすべてのファイルシステムをマウントする。マルチユーザー実行状態になる時に、 <code>mountall</code> コマンドが自動的に実行される。
<code>umount (1M)</code>	ファイルシステムとリモート資源をマウント解除する。
<code>umountall (1M)</code>	<code>/etc/vfstab</code> ファイル内で指定されたすべてのファイルシステムをマウント解除する。

`mount` コマンドは、不整合がある読み取り/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 39 章を参照してください。

`umount` 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。ユーザーがファイルシステム内のファイルにアクセスしているか、ディレクトリをカレントディレクトリにしているか、あるいは、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、そのファイルシステムは、使用中であると認識されます。

汎用マウントオプション

表 36-3 に、`mount` コマンドの `-o` オプションで指定できる汎用マウント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、`-o ro,nosuid` のようにコマンド (空白を入れない) で区切ります。

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 (たとえば、`mount_ufs(1M)`) を参照してください。

表 36-3 -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
<code>bg</code> <code>fg</code>	NFS	最初の試行に失敗すると、バックグラウンド (<code>bg</code>) またはフォアグラウンド (<code>fg</code>) で再試行する。このオプションは重要でない <code>vfstab</code> エントリには安全である。デフォルトは <code>fg</code> 。
<code>hard</code> <code>soft</code>	NFS	サーバーが応答しない場合の手続きを指定する。 <code>soft</code> は、エラーが返されることを示す。 <code>hard</code> 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 <code>hard</code> 。
<code>intr</code> <code>nointr</code>	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 <code>intr</code> (割り込み可能)。
<code>largefiles</code> <code>nolargefiles</code>	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。 <code>largefiles</code> 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)。デフォルトは <code>largefiles</code> 。 <code>largefiles</code> 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。

表 36-3 -o で指定する汎用マウントオプション 続く

オプション	ファイルシステム	説明
logging nologging	UFS	<p>ファイルシステムのロギングを有効にする。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きれいにシャットダウンできなかったとき、システムをリポ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>
noatime	UFS	<p>ファイルのアクセス時間更新を抑制する。ただし、ctime または mtime の更新が同時に行れると場合を除く。stat(2) 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 (たとえば、Usenet ニューススプール) 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。</p>
remount	All	<p>読み取り専用のファイルシステムを読み書き可能として (rw オプションで) 再マウントする。このオプションは、f、logging nologging、および m オプションと組み合わせる場合にだけ指定できる。このオプションは現在マウントされている読み取り専用のファイルシステムに対してのみ動作する。</p>
retry= <i>n</i>	NFS	<p>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。</p>
ro rw	CacheFS, NFS, PCFS, UFS, S5FS	<p>読み取り/書き込みまたは読み取り専用と指定する。このオプションを指定しない場合のデフォルトは読み取り/書き込み。HSFS のデフォルトのオプションは ro。</p>
suid nosuid	CacheFS, HSFS, NFS, S5FS, UFS	<p>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。</p>

▼ 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

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には、`mount (1M)` コマンドを使用します。

```
$ mount -v
```

`-v`

マウントされているファイルシステムのリストを冗長モードで表示する

例 — 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

```
$ mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic ...
$
```

ファイルシステムのマウント (/etc/vfstab)

/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、表 36-4 に示すように7つのフィールドがあります。

表 36-4 /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	<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資源名 (myserver:/export/home など)。NFS の詳細は、『Solaris のシステム管理 (第 3 巻)』を参照してください。 ■ 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) ■ /proc ディレクトリと proc ファイルシステムのタイプ
device to fsck	<p>「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/rdsk/c0t0d0s0 など)。これによって、fsck が使用する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。</p>
mount point	<p>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。</p>
FS type	<p>「device to mount」フィールドで指定したファイルシステムのタイプを指定する。</p>
fsck pass	<p>fsck 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、フィールドに 0 より大きい値が指定されている場合に UFS 以外のファイルシステムはチェックされる。</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 つずつ検査される。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 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。fsck pass フィールドでは、ファイルシステムをチェックする順序を明示的には指定できない。</p>

表 36-4 /etc/vfstab ファイルのフィールドの説明 続く

フィールド名	説明
mount at boot	システムのブート時にファイルシステムが <code>mountall</code> によって自動的にマウントされるかどうかを表す。yes または no に設定。このフィールドは <code>AutoFS</code> とは連動していないので注意してください。ルート (<code>/</code>)、 <code>/usr</code> 、 <code>/var</code> のファイルシステムは最初は <code>vfstab</code> 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 <code>/proc</code> や <code>/dev/fd</code> 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 (空白を空けずに) 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。汎用オプションについては、表 36-3 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。そうしなければ、システムが正常にブートしない可能性もあります。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。

▼ /etc/vfstab ファイルにエントリを追加する方法

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. /etc/vfstab ファイルを編集してエントリを追加します。

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。

/etc/vfstab のフィールドエントリの詳細は、表 36-4 を参照してください。次のことを確認します。

- 各フィールドを空白 (空白文字またはタブ) で区切る。
- フィールドで値を指定しない場合はダッシュ (-) を入力する。

3. 変更結果を保存します。

例 — /etc/vfstab ファイルにエントリを追加する

次の例では、デフォルトのマウントオプション (読み取り/書き込み) を使用して、ディスクスライス /dev/dsk/c0t3d0s7 を UFS ファイルとして、マウントポイントディレクトリ /files1 にマウントします。また、「device to fsck」として raw キャラクタ型デバイス /dev/rdsk/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/rdsk/c0t3d0s7	/files1	ufs	2	yes	-

次の例では、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 にマウントします。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。信頼性を高めるために、読み取り/書き込みの NFS ファイルシステムでは hard マウントオプションを指定します。

#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

次の例では、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 にマウントします。「mount at boot」には yes を指定し、「device to fsck」と「fsck pass」の番号は指定しません。LOFS ファイルシステムをマウントするときは、LOFS を構成するのに使用され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)

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

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

/etc/vfstab ファイル内の「mount point」または「device to mount」フィールド内のエントリ。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)

次の例では、/etc/vfstab ファイル内に列挙されているファイルシステム /usr/vfstab をマウントします。

```
# mount /usr/dist
```

▼ /etc/vfstab ファイル内に列挙されているすべてのファイルシステムをマウントする方法

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. /etc/vfstab ファイル内に列挙されたファイルシステムをマウントします。

```
# mountall [-l | -r] [-F fstype]
```

オプションを指定しなければ、/etc/vfstab ファイル内で「mount at boot」フィールドに yes を指定したすべてのファイルシステムがマウントされます。

-l	/etc/vfstab ファイル内で「mount at boot」フィールドに yes を指定したすべてのローカルファイルシステムがマウントされる
-r	/etc/vfstab ファイル内で「mount at boot」フィールドに yes を指定したすべてのリモートファイルシステムがマウントされる
-F <i>fstype</i>	/etc/vfstab ファイル内で「mount at boot」フィールドに yes を指定した <i>fstype</i> タイプのすべてのファイルシステムがマウントされる

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。

例 — /etc/vfstab ファイル内に列挙されているすべてのファイルシステムをマウントする

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/rdisk/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
```

次の例では、/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します。

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic on ...
```

次の例では、`/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します。

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/
onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/
onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/
onerror=panic on ...
/usr/dist on mars:/usr/dist remote/read/write/
setuid on Tue Sep 14 15:32:18 1999
```

ファイルシステムをマウントする (mount コマンド)

▼ UFS ファイルシステムをマウントする方法

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. `mount (1M)` コマンドで、**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
```

<code>-o mount-options</code>	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。汎用マウントオプションについては、表 36-3を参照。すべてのオプションについては、 <code>mount_ufs(1M)</code> のマニュアルページを参照。
<code>/dev/dsk/device-name</code>	ファイルシステムを保持す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)。ディスクデバイス名を調べる方法については、370ページの「ディスクスライス情報を表示する方法」を参照。
<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

例 — UFS ファイルシステムをマウントする

次の例では、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントします。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

例 — ログを有効にして、UFS ファイルシステムをマウントする

UFS ログインによってファイルシステムの整合性が保たれるために、システムの起動時間が大幅に短縮されます。次の例は、ログを有効にして、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントします。

```
# 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再マウントする方法

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て、`fsck` 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ります。
2. 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 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する。

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、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。

3. ファイルシステムのマウントを解除します。

```
# umount mount-point
```

4. ファイルシステムの状態をリセットします。

```
# fsck mount-point
```

5. `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

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 ファイルシステムをマウントする方法

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. 資源 (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(1M)` コマンドを使用し、サーバー上の資源を使用可能にしておかなければなりません。資源を共有する方法については、『*Solaris* のシステム管理 (第 3 巻)』を参照してください。

3. `mount(1M)` コマンドで、**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
```

<code>-o mount-options</code>	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 36-3を参照。オプションについては、 <code>mount_nfs(1M)</code> のマニュアルページを参照。
<code>server:/directory</code>	共有する資源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。
<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。

例 — NFS ファイルシステムをマウントする

次の例は、サーバー pluto の /export/packages ディレクトリを /mnt にマウントしています。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ IA: System V (S5FS) ファイルシステムをマウントする方法

1. スーパーユーザーになります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. mount (1M) コマンドで、**S5FS** ファイルシステムをマウントします。

```
# mount -F s5fs [-o mount-options] /dev/dsk/device_name mount-point
```

<i>-o mount-options</i>	S5FS 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については、表 36-3 を参照。オプションについては、mount_s5fs(1M) 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存在するディスクスライスのデバイス名 (/dev/dsk/c0t3d0s7 など)。ディスクデバイス名を調べる方法については、370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。

IA: 例 — S5FS ファイルシステムをマウントする

次の例では、/dev/dsk/c0t3d0s7 を /files1 ディレクトリにマウントします。

```
# mount -F s5fs /dev/dsk/c0t3d0s7 /files1
```

▼ IA: ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法

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ります。

ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。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。

2. `mount (1M)` コマンドで、**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/書き込みまたは読み取り専用でマウントできるように指定する。このオプションを指定しなければ、デフォルトは読み取り/書き込みになる
<code>/dev/dsk/device-name</code>	ディスク全体のデバイス名 (<code>/dev/dsk/c0t0d0p0</code> など)
<code>logical-drive</code>	DOS の論理ドライブ名 (c から z まで) または 1 から 24 までのドライブ番号を指定する。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

「`device-name`」と「`logical-drive`」とは、コロンの区切りが必要があります。

IA: 例 — PCFS (DOS) ファイルシステムをハードディスクからマウントする

次の例では、基本 DOS スライス内のドライブが `/pcfs/c` ディレクトリにマウントされます。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例では、ディスク上の拡張 DOS スライス内の最初の論理ドライブが、`/mnt` に読み取り専用としてマウントされます。

```
# 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、またはより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、修復する場合 (`fsck` コマンドについての詳細は、第 39 章を参照)

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についての詳細は、第 43 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続きの一部として自動的にマウント解除されます。

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。

- スーパーユーザーでなければならない
- ファイルシステムが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。詳細は、495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

マウント解除したファイルシステムを検査する

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。この手順については、482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」を参照してください。

▼ 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ります。
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ます
<code>mount-point</code>	プロセスを終了させたいファイルシステム名

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。

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。

```
# fuser -c -k mount-point
```

ファイルシステムを使用中のプロセスに SIGKILL が送信されます。

4. ファイルシステムを使用中の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

次の例では、`/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

次の手順で、ファイルシステム (`/`、`/usr` または `/var` を除く) をマウント解除します。

注 - ルート (`/`)、`/usr` および `/var` の UFS ファイルシステムは特殊な場合です。システムが機能するにはルートが必要なので、ルート (`/`) のファイルシステムは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. 494ページの「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

マウント解除したいファイルシステム名。ここでは、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資源、LOFS ファイルシステムのループバックディレクトリのいずれかを指定できる

例 — 1 つのファイルシステムをマウント解除する

次の例では、ローカルのホームディレクトリからファイルシステムがマウント解除されます。

```
# umount /export/home
```

次の例で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のマウントが解除されます。

```
# umount /dev/dsk/c0t0d0s7
```

▼ すべてのファイルシステムをマウント解除する方法 (/etc/vfstab)

次の手順で、/、/proc、/var、/usr のファイルシステムを除き、/etc/vfstab ファイルに列挙されているすべてのファイルシステムをマウント解除します。

1. 494ページの「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. /etc/vfstab ファイルに列挙されているすべてのファイルシステムをマウント解除します。

```
# umountall
```

可能なファイルシステムがすべてマウント解除されます。使用中のファイルシステムはマウント解除されません。

3. 使用中だったためにマウント解除されなかったファイルシステムについては、495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」で説明した手順に従って、マウント解除できるようにします。

4. 必要に応じて、すべてのファイルシステムがマウント解除されるまで手順 2 を繰り返します。

キャッシュファイルシステム (手順)

キャッシュファイルシステム (CacheFS) 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機構です。CacheFS は階層化ファイルシステムとして設計されていて、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 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) のような低速リンク上でクライアントのパフォーマンスを改善します。

この章の内容は次のとおりです。

- 500ページの「CacheFS の機能」
- 501ページの「キャッシュファイルシステムの設定」
- 502ページの「キャッシュを作成する方法」
- 504ページの「mount を使用してキャッシュにマウントするファイルシステムを指定する方法」
- 506ページの「/etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法」
- 507ページの「AutoFS 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方法」
- 510ページの「キャッシュ内のファイルシステムを変更する方法」
- 511ページの「キャッシュされたファイルシステムに関する情報を表示する方法」
- 512ページの「必要に応じて整合性チェックを指定する方法」
- 512ページの「キャッシュされたファイルシステムを削除する方法」

- 514ページの「キャッシュされたファイルシステムの完全性をチェックする方法」
- 529ページの「CacheFS の統計情報」
- 530ページの「CacheFS 統計情報を設定して表示するための前提条件」
- 531ページの「CacheFS 統計情報の設定」
- 531ページの「ロギングプロセスを設定する方法」
- 533ページの「キャッシュサイズの表示」
- 533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」
- 535ページの「統計情報の表示」
- 535ページの「キャッシュ統計情報を表示する方法」
- 536ページの「キャッシュの構造と動作」
- 537ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」
- 537ページの「必要に応じて行う整合性チェック」

CacheFS の機能

キャッシュにマウントするように指定したファイルシステムに、ユーザー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るように、`cfsadmin(1M)` コマンドを使用してキャッシュをクライアント上で作成します。図 37-1 に、CacheFS の使用に関連する構成要素の関係を示します。

バックファイルシステムとは、キャッシュにマウントするように指定する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。したがって、ユーザーがファイルにアクセスするまで、キャッシュは空になっています。ユーザーにとって、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、同じファイルを使用すると高速になります。

注 - 共有可能なファイルシステムしかマウントできません。`share(1M)` コマンドを参照してください。

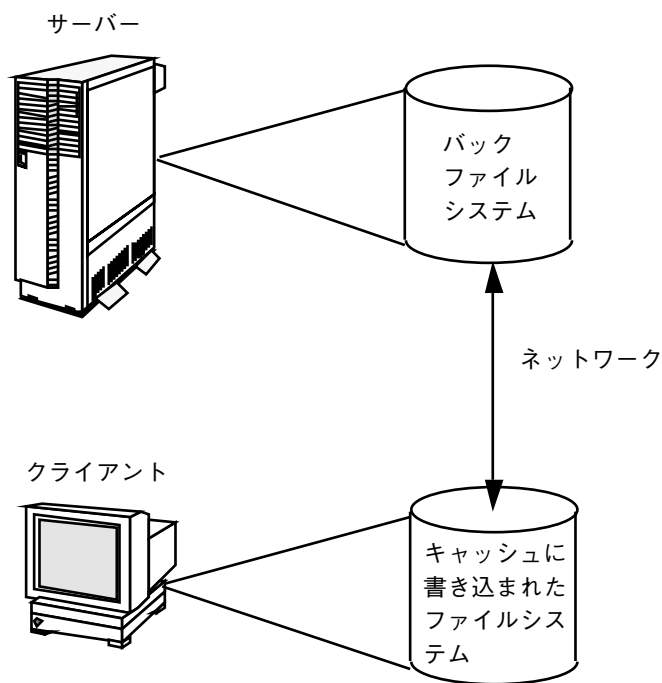


図 37-1 CacheFS の機能

キャッシュファイルシステムの設定

表 37-1 作業マップ：キャッシュファイルシステムの設定

作業	説明	手順の説明
1. キャッシュの作成	cfsadmin コマンドを使用してキャッシュを作成する。	502ページの「キャッシュを作成する方法」
2. ファイルシステムをキャッシュにマウントする	mount コマンド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。	504ページの「mount を使用してキャッシュにマウントするファイルシステムを指定する方法」

表 37-1 作業マップ: キャッシュファイルシステムの設定 続く

作業	説明	手順の説明
	<p>/etc/vfstab ファイルを編集してファイルシステムをキャッシュに書き込む。</p> <p>AutoFS を使用してファイルシステムをキャッシュに書き込む。</p>	<p>506ページの「/etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法」</p> <p>507ページの「AutoFS 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方法」</p>

キャッシュを作成する

次に、キャッシュディレクトリを作成する手順について説明します。

▼ キャッシュを作成する方法

1. スーパーユーザーになります。
2. `cfsadmin -c` コマンドを使用してキャッシュを作成します。

```
# cfsadmin -c cache-directory
```

cache-directory

キャッシュが作成されるディレクトリ名。詳細は、`cfsadmin(1M)` のマニュアルページを参照

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ます。

例 — キャッシュを作成する

次の例では、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します。

```
# mkdir /local
# cfsadmin -c /local/mycache
```

キャッシュにマウントするファイルシステムの指定

指定したファイルシステム内のファイルにユーザーがローカルにアクセスできるように、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。

キャッシュにマウントされるファイルシステムを指定するには、次の3つの方法があります。

キャッシュにマウントされるファイルシステムを指定する方法	説明
<code>mount (1M)</code> 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 <code>mount</code> コマンドを使用する
<code>/etc/vfstab</code> ファイルを編集する方法	一度だけですむ。 <code>/etc/vfstab</code> 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	一度だけですむ。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。

注 - CacheFS 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。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には、Solstice AutoClient 製品を購入しなければなりません。AutoClient 製品の詳細は、『Solstice AutoClient 2.1 管理者ガイド』を参照してください。

▼ mount を使用してキャッシュにマウントする ファイルシステムを指定する方法

1. スーパーユーザーになります。

2. マウントポイントを作成します。

マウントポイントを作成すると、ユーザーはそのマウントポイントで指定されたファイルシステムにアクセスできます。マウントポイントはどこからでも作成できます。次の手順のように、マウントコマンドで使用する CacheFS オプションによって、作成するマウントポイントを、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むように決めます。

3. mount コマンドを使用して、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タイプ (NFS または HSFs)
<i>cache-directory</i>	キャッシュがあるディレクトリ名。これは、502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を指定する。CacheFS マウントオプションの詳細は、mount_cachefs(1M) のマニュアルページを参照

back-filesystem

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れば、merlin: /usr/openwin など、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) を指定しなければならない

mount-point

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するには、次のように `cachefsstat (1M)` コマンドを使用します。

```
# cachefsstat mount-point
```

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
```

マウントポイントは、作成したキャッシュファイルシステムです。cachefsstat コマンドの詳細は、529ページの「CacheFS の統計情報」を参照してください。

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
```

例 — mount を使用してキャッシュにマウントするファイルシステムを指定する

次の例では、マウントポイント /docs を作成し、NFS ファイルシステム merlin:/docs を、キャッシュされたファイルシステム /docs としてキャッシュ /local/mycache にマウントします。

```
# mkdir /docs
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

次の例では、CD-ROM (HSFS ファイルシステム) をキャッシュされたファイルシステム /docs として使用できるようにします。CD-ROM には書き込めないため、引数 ro を指定してキャッシュされたファイルシステムを読み込み専用にします。CD-ROM は挿入時にボリューム管理によって自動的にマウントされるので、引数 backpath を指定しなければなりません。マウントポイントは /cdrom ディレクトリにあり、CD-ROM 名によって決まります。マウントする特殊デバイスは、backpath オプションに指定した値と同じです。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/local/mycache,ro backpath=/cdrom/cdrom_name /cdrom/cdrom_name /docs
```

次の例では、demandconst オプションを使用して、NFS キャッシュファイルシステム /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。詳細は、537ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」を参照してください。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ /etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法

1. 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。

```
#device      device          mount FS    fsck  mount  mount
#to mount    to fsck          point type  pass  at boot options
#
/dev/dsk/devicename /dev/rdsk/devicename /mount-point cachefs 2    yes  -
```

最終行は新しいエントリを表します。

3. 次のように `mount` コマンド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か、リブートします。

```
# mount /mount-point
```

例 — /etc/vfstab ファイルを編集してキャッシュにファイルシステムをマウントする

次の例は、キャッシュファイルシステム用の /etc/vfstab エントリを示します。

```
#device      device          mount FS    fsck  mount  mount
#to mount    to fsck          point type  pass  at boot options
#
/dev/dsk/c0t1d0s0 /dev/rdsk/c0t1d0s0 /usr/local cachefs 2    yes  -
```

次のように入力して、/usr/local ディレクトリをキャッシュディレクトリにマウントします。

```
# mount /usr/local
```

▼ AutoFS 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方法

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 によってキャッシュにファイルシステムをマウントします。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についての詳細は、`automount (1M)` のマニュアルページや、『Solaris のシステム管理 (第 3 巻)』も参照してください。

1. 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 とマップを編集する方法についての詳細は、『Solaris のシステム管理 (第 3 巻)』の AutoFS に関する章を参照してください。

例 — AutoFS 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

次の `auto_master` エントリは、キャッシュ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
```

キャッシュされたファイルシステムの管理

表 37-2 作業マップ：キャッシュされたファイルシステムの管理

作業	説明	手順の説明
1. キャッシュの変更	キャッシュの動作を変更する。	510ページの「キャッシュ内のファイルシステムを変更する方法」
2. キャッシュ情報の表示	<code>cfsadmin</code> コマンドを使用してキャッシュされたファイルシステムに関する情報を表示する。	511ページの「キャッシュされた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。	512ページの「必要に応じて整合性チェックを指定する方法」
4. キャッシュの削除	<code>umount</code> コマンドと <code>cfsadmin</code> コマンドを使用して、キャッシュされたファイルシステムを削除する。	512ページの「キャッシュされたファイルシステムを削除する方法」
5. ファイルシステムの完全性の検査	<code>fsck_cachefs</code> コマンドを使用して、キャッシュされたファイルシステムの完全性をチェックする。	514ページの「キャッシュされたファイルシステムの完全性をチェックする方法」

キャッシュの管理

キャッシュを設定し終わったら、次の管理作業を実行できます。

- キャッシュ内のファイルシステムを変更する (キャッシュをマウント解除し、削除し、作成し直し、再びマウントする)
- キャッシュ情報を表示する
- キャッシュの整合性をチェックする
- キャッシュからファイルシステムを削除する
- キャッシュされたファイルシステムの完全性をチェックする

注 - /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。

▼ キャッシュ内のファイルシステムを変更する方法

ファイルシステムの特定のオプションを変更する方法については、第 36 章を参照してください。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。

次の例は、この手順を示しています。

例 — キャッシュ内でファイルシステムを変更する

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。この例は、シングルユーザーモードでリブートするなどの手順を示しています。必要であれば他のコマンドを使用して、この例の作業を実行することもできます。

```
# shutdown -g30 -y
.
.
.
Type Cntrl-d to proceed with normal startup,
(or give root password for system maintenance):
# enter password:
.
.
.
Here is where you might be prompted from system to run fsck on the
file system where the cache is located.

# fsck /local
# mount /local
# cfsadmin -d all /local/mycache
# cfsadmin -c /local/mycache
# init 6
.
.
.
console login:
password:
```

(続く)

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1, demandconst merlin:/docs /docs
#
```

ファイルシステムをキャッシュに正常に戻せなかった場合は、次のようなエラーメッセージが表示されます。

```
cachefsstat: /doc: not a cachefs mount point
```

▼ キャッシュされたファイルシステムに関する情報を表示する方法

1. 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名です。

例 — キャッシュされたファイルシステムに関する情報を表示する

次の例は、キャッシュディレクトリ `/local/mycache` に関する情報を示しています。この例では、ファイルシステム `/docs` が `/local/mycache` にマウントされています。最終行にはキャッシュされたファイルシステムが表示されます。

```
# 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
```

(続く)

```
#
```

▼ 必要に応じて整合性チェックを指定する方法

1. スーパーユーザーになります。
2. 次のように `mount (1M)` コマンドの `demandconst` オプションを指定して、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=nfs,cachedir=/directory,demandconst  
server:/file-system /mount-point
```

3. キャッシュされた特定のファイルシステム上で整合性チェックを有効にするには、次のように `cfsadmin -s` コマンドを使用します。

```
# cfsadmin -s /mount-point
```

整合性チェックについての詳細は、537ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」を参照してください。

▼ キャッシュされたファイルシステムを削除する方法

1. スーパーユーザーになります。
2. キャッシュ内のすべてのファイルシステムをマウント解除します。

```
# umount mount-point
```

`mount-point` には、削除したい、キャッシュされたファイルシステムを指定します。

3. 次のように、`cfsadmin -l` の出力からキャッシュ 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. キャッシュされたファイルシステムを指定したキャッシュから削除します。

```
# cfsadmin -d cache-id cache-directory
```

<i>cache-id</i>	キャッシュされた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。詳細は、511ページの「キャッシュされた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すべてのファイルシステムを削除できる
<i>cache-directory</i>	キャッシュがあるディレクトリ

5. キャッシュされたファイルシステムが削除されたことを確認します。

削除したファイルシステムのキャッシュ ID は、次のコマンドの出力に含まれなくなります。コマンド出力で指定するフィールドの詳細は、`cfsadmin(1M)` のマニュアルページを参照してください。

```
# cfsadmin -l cache-directory
cfsadmin: list cache FS information
maxblocks      90%
minblocks      0%
threshblocks   85%
maxfiles       90%
minfiles       0%
threshfiles    85%
maxfilesize    3MB
```

(続く)

```
#
```

例 — キャッシュされたファイルシステムを削除する

次の例では、キャッシュされたファイルシステムをマウント解除して、キャッシュから削除します。

```
# umount /usr/openwin  
# cfsadmin -d merlin:docs /docs /local/mycache
```

次の例では、/local/mycache キャッシュ内のキャッシュされたファイルシステムをすべて削除します。

```
# cfsadmin -d all /local/mycache
```

▼ キャッシュされたファイルシステムの完全性をチェックする方法

`fsck(1M)` コマンドを使用して、キャッシュされたファイルシステムが完全かどうかを確認します。何も操作しなくても、`fsck` の `CacheFS` バージョンによって問題が自動的に解決されます。キャッシュされたファイルシステムの場合は、`fsck` を手作業で実行する必要はありません。`fsck` 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ます。完全性を手作業で確認したい場合は、次の手順を使用できます。

詳細は、`fsck_cachefs(1M)` を参照してください。

1. 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キャッシュされた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> にキャッシュされたファイルシステムをチェックさせるが、修復しない
<code>-o noclean</code>	キャッシュされたファイルシステム上でチェックのみを実行させるが、修復しない
<code>cache-directory</code>	キャッシュがあるディレクトリ名

例 — キャッシュされたファイルシステムの完全性をチェックする

次の例では、`/local/mycache` キャッシュの一部になっているキャッシュされたファイルシステムをチェックします。

```
# fsck -F cacheefs /local/mycache
#
```

cachefspack によるキャッシュファイルシステムの管理

一般的な使い方でも、CacheFS 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。パッキング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。

パッキング機能を使用すると、キャッシュにロードするファイルやディレクトリを指定できます。つまり、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。これによって、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を指定する手間が省けます。

cachefspack コマンドは、CacheFS ファイルシステムの制御にパッキング機能を追加します。

▼ 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
```

-p ファイルまたは複数のファイルをパックすることを指定する。これはデフォルト

filename キャッシュにパックするファイル名またはディレクトリ名を指定する。ディレクトリを指定すると、そのサブディレクトリもすべてパックされる。cachefspack コマンドの詳細は、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 ファイルをキャッシュにパックするように指定する方法を示しています。

```
$ cachefspack -p projects
```

次の例は、複数のファイルをキャッシュにパックするように指定する方法を示しています。

```
$ cachefspack -p projects updates master_plan
```

次の例は、ディレクトリをキャッシュにパックするように指定する方法を示しています。

```
$ cachefspack -p /usr/openwin/bin
```

パッキングリスト

cachefspack コマンドの機能の 1 つに、パッキングリストをパックするという機能があります。これによって、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。

パッキングリストには、キャッシュにバ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バックされます。

▼ 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。filesync コマンドの詳細は、filesync(1) 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、バ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バックされるファイルを指定しています。
- IGNORE 文は、バ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。

▼ パッキングリストに指定してファイルをキャッシュにバックする方法

パッキングリストを使用してファイルをバックするには、次に示すように、cachefspack コマンドに -f オプションを指定します。

```
$ cachefspack -f packing-list
```

これは、パッキングリストを読み取って、パッキングリストに指定された情報に基づいてファイルをバックすることを意味します。

-f パッキングリストを使用することを指定する。

packing-list パッキングリスト名を指定する。

例 — パッキングリストに指定したファイルをキャッシュにバックする

この例では、`list.pkg` ファイルを `cachefspack` コマンド用のパッキングリストとして使用しています。

```
$ cachefspack -f list.pkg
```

▼ 正規表現として扱うファイルをパッキングリストに指定する方法

(文字どおりのファイル名ではなく) 正規表現として扱う 1 つまたは複数のファイルをパッキングリストに指定するには、`cachefspack` コマンドの `-r` オプションと `-f` オプションを使用します。 `-r` オプションのみを使用することはできません。

```
$ cachefspack -rf packing_list
```

`packing_list` には、次のように定義された `LIST` コマンドが入っています。

```
LIST *.doc
```

-r `LIST` コマンドに定義したファイルまたは複数のファイルを (文字どおりのファイル名ではなく) 正規表現として扱うことを指定する

-f パッキングリストをキャッシュにバックすることを指定する

packing_list 正規表現として扱うファイルまたは複数のファイルを指定した `LIST` コマンドが入っているパッキングリスト名を示す

例 — 正規表現として扱うファイルをパッキングリストに指定する

次の例は、パッキングリスト `list.pkg` をキャッシュにバックするように指定する方法を示しています。 `list.pkg` には、正規表現を定義する `LIST` コマンドが入っています。

```
$ cachefspack -rf list.pkg
```

このコマンドは、list.pkg ファイルをキャッシュにパッキングして、LIST コマンドに定義されたファイル名を (文字どおりのファイル名ではなく) 正規表現として扱います。

▼ 共有ディレクトリのファイルをパックする方法

1. 共有ディレクトリにある、ユーザーが所有するファイルだけをパックするには、次のように LIST コマンドをパッキングリストに定義します。

```
LIST !find . -user your_user_name -print
```

2. cachefspack -sf コマンドを使用して、パッキングリストをキャッシュにパックします。

```
$ cachefspack -sf packing_list
```

-s	find コマンドの出力をパッキングリストに適するように調整する。
-f	パッキングリストを読み取ることを指定する。
filename	読み取るパッキングリスト名を指定する。

注 - -s オプションは -f オプションと併用しなければなりません。-s オプションのみを使用することはできません。

例 — 共有ディレクトリのファイルをパックする

次の例は、ユーザーが所有するベースディレクトリのファイルだけをパックするために、LIST コマンドをパッキングリストに定義する方法を示しています。

```
LIST !find . -user jones -print
```

次の例は、パッキングリストをパックするように指定する方法を示しています。

```
$ cachefspack -sf /projects/proj_1
```

ファイルのパッキング解除

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もあります。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をのパッキングを解除するには、`cachefspack` コマンドの `-u` オプションか `-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指定する。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名またはパッキングリスト名を指定する。 <code>cachefspack</code> コマンドの詳細は、マニュアルページを参照
<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指定する

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、キャッシュからファイル `/usr/openwin/bin/xlogo` のパッキングを解除するように指定する方法を示しています。

```
$ cachefspack -u /usr/openwin/bin/xlogo
```

次の例は、キャッシュから複数のファイルのパッキングを解除するように指定する方法を示しています。


```
$ cd /usr/openwin/bin
$ cacheftpck -u xlogo xview xcolor
```

また、次のように、パッキングリスト (ファイルのディレクトリのパスを含むファイル) でもパッキングを解除できます。

```
$ cacheftpck -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を解除するように `-U` オプションを使用する方法を示しています。

```
$ cacheftpck -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。

```
$ cacheftpck -U /local/mycache
cacheftpck: Could not unpack cache /local/mycache, no mounted
filesystems in the cache.
```

パックされたファイルの情報の表示

パックされたファイルについての情報やパッキングの状態を表示することも可能です。

▼ 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の情報を表示するには、`cacheftpck` コマンドの `-i` オプションを使用します。

```
$ cacheftpck -i [v] cached-filename-or-directory
```

- i パックされたファイルの情報を表示することを指定する

- v 詳細表示オプション

- cached-filename-or-directory* 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

次の例は、ファイル `doc_file` が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
```

次の例は、ディレクトリ `/usr/openwin` に関する情報を示しています。このディレクトリには、サブディレクトリ `bin` があり、サブディレクトリ `bin` には `xterm`、`textedit`、`resize` の3つのファイルがあります。`xterm`と `resize` 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。`textedit` ファイルは正常にパックされています。

```
$ cd /usr/openwin
$ cachefspack -i bin
.
.
.
cachefspack: file /bin/xterm marked packed YES, packed NO
cachefspack: file /bin/textedit marked packed YES,
packed YES
cachefspack: file /bin/resize marked packed YES,
packed NO
.
.
.
```

`-iv` オプションを使用した場合、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の追加情報も表示されます。たとえば、次のとおりです。

```
$ cd /usr/openwin
$ cachefspack -iv bin
.
.
.
cachefspack: file /bin/xterm marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/textedit marked packed YES,
packed YES, nocache NO
cachefspack: file /bin/resize 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。

cachefspack コマンドのヘルプの表示

cachefspack のすべてのオプションのヘルプの要約とその意味を表示するには、次のように `-h` オプションを使用します。

```
$ 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
```

cachefspack エラー

cachefspack コマンドを使用する時に、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。
```

エラーが発生した原因

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。

問題を解決する方法

適切なアクセス権を設定してください。

```
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。
```

エラーが発生した原因

正しいファイルまたはディレクトリがありません。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

コマンドを再実行してください。

```
cachefspack: pathname - ディレクトリをオープンできません。I/O エラー
```

エラーが発生した原因

ハードウェアの障害です。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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、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。
```

エラーが発生した原因

実行可能ファイルが壊れているか、そのフォーマットを認識できません。

問題を解決する方法

解決方法はありません。

```
cachefspack: filename - ファイルを pack できません。アクセス権がありません。
```

エラーが発生した原因

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。

問題を解決する方法

適切なアクセス権を入手してください。

```
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。
```

エラーが発生した原因

正しいファイルまたはディレクトリがありません。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

コマンドを再実行してください。

```
cachefspack: filename - ファイルを pack できません。I/O エラー
```

エラーが発生した原因

ハードウェアの障害です。

問題を解決する方法

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。
```

エラーが発生した原因

ディスク容量が足りなくなりました。キャッシュの最大容量に達しました。

問題を解決する方法

ディスク容量を増やす必要があります。キャッシュのサイズを増やしてください。

```
cachefspack: filename - ファイルを unpack できません。アクセス権がありません。
```

エラーが発生した原因

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。

問題を解決する方法

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。
```

エラーが発生した原因

正しいファイルまたはディレクトリがありません。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

コマンドを再実行してください。

```
cachefspack: filename - ファイルを unpack できません。I/O エラー
```

エラーが発生した原因

ハードウェアの障害です。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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。
```

エラーが発生した原因

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。

問題を解決する方法

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 bese はありません
```

エラーが発生した原因

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。

問題を解決する方法

BASE コマンドを定義してください。

CacheFS の統計情報

CacheFS の統計情報は、次のように利用できます。

- 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。

これらの結果から得られた情報を使用すると、キャッシュサイズと望ましいパフォーマンスを取捨選択して調整できます。

CacheFS の統計に関連する、次の 3 つのコマンドがあります。

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 (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。
cachefsstat (1M)	特定のファイルシステム、またはキャッシュされたすべての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せん。

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。

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、533ページの「ロギングプロセスを停止する方法」を参照してください。

CacheFS 統計情報を設定して表示するための前提条件

CacheFS 統計情報コマンドを使用する前に、次の操作を実行しておかなければなりません。

- cfsadmin (1M) 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、1 日、1 週間、1 カ月など、一般的な作業間隔に等しくする必要がある。
- 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 統計情報の設定

表 37-3 に、CacheFS 統計情報の設定に関連する手順を示します。

表 37-3 作業マップ : CacheFS 統計情報の設定

作業	説明	手順の説明
1. ログインの設定	<code>cachefslog</code> コマンドを使用して、キャッシュされたファイルシステムのログインを設定する。	531ページの「ログインプロセスを設定する方法」
2. ログファイルの検索	<code>cachefslog</code> コマンドを使用してログファイルの位置を特定する。	532ページの「ログファイルの場所を調べる方法」
3. ログインプロセスの停止	<code>cachefslog</code> コマンドを使用してログインプロセスを停止する。	533ページの「ログインプロセスを停止する方法」
4. キャッシュサイズの表示	<code>cachefswssize</code> コマンドを使用してキャッシュサイズを表示する。	533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」
5. キャッシュ統計情報の表示	<code>cachefsstat</code> コマンドを使用して統計情報を表示する。	535ページの「キャッシュ統計情報を表示する方法」

CacheFS ログイン

この節では CacheFS ログインを設定し、表示する方法を説明します。

▼ ログインプロセスを設定する方法

1. `cachefslog` コマンドを使用してログインプロセスを設定します。

```
$ cachefslog -f log-file-path mount-point
```

<code>-f</code>	ロギングプロセスを設定する
<code>log-file-path</code>	ログファイルの位置を表す。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
<code>mount-point</code>	統計情報を収集するマウントポイント (キャッシュされたファイルシステム) 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するには、次のように `cachefslog` コマンドを使用します。

```
$ cachefslog mount-point
```

例 — ロギングプロセスを設定する

次の例では、ログファイル `samlog` を設定して `/home/sam` に関する統計情報を収集します。`samlog` の場所は `/var/tmp/samlog` です。

```
$ cachefslog -f /var/tmp/samlog /home/sam  
/var/tmp/samlog: /home/sam
```

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(1M)` 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` 統計情報を表示したいキャッシュされたファイルシステム

例 — ログファイルの場所を調べる

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
```

ロギングプロセスを停止する方法

`cachefslog(1M)` コマンドの `-h` オプションを使用して、ロギングプロセスを停止します。

```
$ cachefslog -h mount-point
```

例 — ロギングプロセスを停止する

次の例では、`/home/stuff` のロギングを停止します。

```
$ 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、ロギングプロセス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。

キャッシュサイズの表示

キャッシュサイズを増やすべきかどうかを確認したり、特定のマウントポイントに関して前回 `cachefslog(1M)` コマンドを使用した後の作業から理想的なキャッシュサイズを決定したりできます。

▼ 作業セット (キャッシュ) のサイズを表示する方法

1. スーパーユーザーになります。
2. `cachefswssize(1M)` コマンドを使用して、現在のログのキャッシュサイズと最大キャッシュサイズを表示します。

```
# cachefswwsize log-file-path
```

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 cachefswwsize 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。

```
# cachefswwsize /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

/usr/openwin
  end size: 7168k
high water size: 7168k

/nfs/venus.svr4
  end size: 4688k
high water size: 5000k

/usr
  end size: 4992k
high water size: 4992k

total for cache
initial size: 110960k
  end size: 30872k
high water size: 30872k
```

統計情報の表示

キャッシュされた特定の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 37-4 統計情報の出力に表示される用語

用語	説明
ヒット率 (hit rate)	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かったときに発生する。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
チェック数 (checks)	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)	書き込みや作成など、変更操作の回数

▼ キャッシュ統計情報を表示する方法

`cachefsstat (1M)` 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

統計情報を表示したいキャッシュされたファイルシステム

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。

例 — 統計情報を表示する

```
$ cachefsstat /home/sam
  cache hit rate: 73% (1234 hits, 450 misses)
consistency checks: 700 (650 pass, 50 fail)
```

```

        modifies: 321
garbage collection: 0

```

キャッシュの構造と動作

各キャッシュには、その動作と構造を決定する一連のパラメータが付いています。各パラメータは、表 37-5 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 37-5 キャッシュパラメータとデフォルト値

キャッシュパラメータ	デフォルト値	定義
maxblocks	90%	CacheFS 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
minblocks	0%	CacheFS 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 が minblocks で指定したより多数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 がフロントファイルシステム内で要求できる使用可能 i ノードの最大数 (ファイル数) を指定する
minfiles	0%	CacheFS がフロントファイルシステム内で要求できる使用可能 i ノードの最小数 (ファイル数) を指定する
threshfiles	85%	CacheFS が minfiles で指定したより多数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 (ファイル数) 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したい場合は、`maxblocks` と `maxfiles` の設定を変更できます。そのためには、`cfsadmin(1M)` 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
```

キャッシュされたファイルシステムとバック ファイルシステムとの整合性チェック

CacheFS 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た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 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、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。

ユーザーがディレクトリやファイルの処理を要求すると、CacheFS は整合性を検査する時間があるかどうかをチェックします。時間があれば、バックファイルシステムから変更時刻を取得して比較します。

必要に応じて行う整合性チェック

`mount(1M)` コマンドの `demandconst` オプションを指定すると、このオプションでマウントしたファイルシステムに関して明示的に要求するときのみ、整合性チェックを実行できます。ファイルシステムをキャッシュにマウントするときに `demandconst` オプションを指定すると、`-s` オプションを指定して `cfsadmin` コマンドを使用し、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。 `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。

追加スワップ空間の構成 (手順)

この章の内容は次のとおりです。

- 540ページの「スワップ空間と仮想メモリー」
- 540ページの「スワップ空間と TMPFS ファイルシステム」
- 541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」
- 542ページの「スワップ空間の割り当て方法」
- 543ページの「スワップ空間の計画」
- 544ページの「スワップ資源の監視」
- 545ページの「スワップ空間の追加」
- 548ページの「スワップファイルを削除する」

スワップ空間について

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
- 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` メモリーとして参照されます。

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` ディレクトリ内）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。このため、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資源から `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` 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、アプリケーションの /tmp の使用量が大きい場合に、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資源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、スワップ資源 TMPFS をどの程度使用できるかを制御する。
- スワップ空間が足りなくなった場合は、コンパイラを TMPDIR 環境変数を使用し、より大きな実際のディレクトリを指すことができる。

コンパイラの TMPDIR 変数を使用すると、コンパイラが /tmp を使用するかどうかだけが制御される。他のプログラムによる /tmp の使用には影響しない。

スワップ空間の追加が必要かどうかを調べる方法

この節では、スワップ空間を使い果たしたときに表示されるエラーメッセージについて説明します。

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` メモリー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

```
directory: File system full, swap space limit exceeded
```

このメッセージは、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。このエラーは、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, memory allocation failed
```

このメッセージは、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。

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表 38-1 に示すようにデフォルトのスワップスライスを割り当てます。

表 38-1 デフォルトのスワップ空間割り当て

システム上の物理メモリー	割り当てられるディレクトリのスワップ空間
16 ~ 63M バイト	32M バイト
64 ~ 127M バイト	64M バイト
128 ~ 511M バイト	128M バイト
512M バイトを超える場合	256M バイト

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。スワップファイルを作成する方法については、545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ファイルは /etc/vfstab ファイル内に列挙され、システムのブート時に /sbin/swapadd スクリプトによって起動されます。

/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ファイルの完全パス名
- 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稼働される前にマウントしなければならないので、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起動するエントリより前に入っていることを、/etc/vfstab ファイル内で確認してください。

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。

一般にデータファイルのサイズが 10 ~ 20M バイトを超えるアプリケーションの場合は、スワップ空間の要件を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からスワップ空間の要件を入手できない場合は、次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。

- アプリケーションをサポートするために、次のように割り当てる。
 - xterm のような通常のアプリケーションごとに 1M バイト
 - カレンダーやメールアプリケーションのような軽量アプリケーションごとに 2 ~ 3M バイト
 - デスクトップパブリッシング (DTP) ソフトウェアのような大型アプリケーションには 20 ~ 50M バイト
- クラッシュダンプを節約するために、物理メモリー全体を割り当てて最悪の場合のクラッシュダンプを保存する。

- システムまたはアプリケーションの要件がわからない場合は、システムの物理メモリーのうち 50% ないし 100% を割り当てる。たとえば、32M バイトの物理メモリーを持つシステムの場合は、16 ~ 32M バイトのスワップ空間を割り当てる。これによって、仮想スワップ空間の合計は 48 ~ 64M バイトになる。
- 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、540ページの「スワップ空間と TMPFS ファイルシステム」を参照。

スワップ資源の監視

/usr/sbin/swap コマンドを使用してスワップ空間を管理します。2つのオプション -l と -s を使用して、スワップ資源に関する情報を表示します。

swap -l コマンドを使用してシステムのスワップ空間を調べます。有効な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/c0t2d0s1 32,17  8 205624 192704
```

swap -s コマンドを使用してスワップ資源を監視します。

```
# swap -s
total: 10492k bytes allocated + 7840k reserved = 18332k used, 21568k available
```

used と available の数値の統計は、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 38-2 に、`swap -s` コマンドの出力とその説明を示します。

表 38-2 `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。次に説明するように、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。その方法については、第 28 章で使用中のシステムに該当する説明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。
- `swap` 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。ステイ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。

表 38-3 に、`mkfile` コマンドのオプションを示します。

表 38-3 `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>	詳細モード。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が、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よいでしょう。

2. スワップファイルを作成します。

```
# mkfile nnn[k|b|m] filename
```

サイズ「nnn」(K バイト数、バイト数、M バイト数) の指定したファイル名のスワップファイルが作成されます。

3. 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。

4. ファイルの完全パス名を指定し、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 -
```

5. 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

次の例では、/files/swapfiles という 24M バイトのスワップファイルを作成します。

```
# mkfile 24m /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)
```

(続く)

```

/files/swapfile  -      -      swap      -      no      -
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/dsk/c0t2d0s1 32,17  8 205624 192704
/files/swapfile  -      -      8 40952  40952

```

スワップファイルを削除する

不要になった余分なスワップ空間は、削除することができます。

▼ 余分な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. 次のように `swap -d` コマンドを使用して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リストから削除されるので、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 swap-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。

ファイルシステムをマウントするについての詳細は、第 36 章を参照してください。

例 — 余分な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します。

```
# swap -d /files/swapfile
# (削除され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/c0t2d0s1 32,17      8  205624 192720
```


ファイルシステムの整合性チェック

この章の内容は以下のとおりです。

- 552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」
- 554ページの「fsck でチェックして修復される内容」
- 562ページの「ブート時のファイルシステムチェック機能の変更」
- 565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」
- 568ページの「不正なスーパーブロックの復元」
- 571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 (第 2 巻)』の「ファイルシステムで発生する問題の解決」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背景情報については、第 40 章を参照してください。

ファイルシステムの完全性

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、ファイルシステムが損傷したり、整合性が失われたりすることがあります。

- 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破損は重大ですが、あまり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` プログラムを使用して) 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。

この章では、`fsck` プログラムでチェックされ修復される問題と `fsck` のオプションについて説明します。また、次の作業についても説明します。

- ブート時に実行される自動チェック機能を変更する方法
- ファイルシステムのチェックが必要かどうかを調べる方法
- UFS ファイルシステムを対話形式でチェックして修復する方法
- 不良スーパーブロックを復元する方法
- `fsck` で修復できない UFS ファイルシステムを修復する方法

`fsck` のエラーメッセージについては、『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」を参照してください。

`fsck` プログラムは、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。各ファイルの `i` ノード番号が名前として割り当てられます。`lost+found` ディレクトリが存在しない場合は、`fsck` 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、そのサイズが `fsck` によって拡張されます。

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どうか判断されます。このフラグはブート時には `/sbin/rcS` スクリプトによって使用され、コマンド行からは `fsck` に `-m` オプションを指定した場合に使用されます。`fsck` の `-m` オプションの結果を無視するのなら、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。

表 39-1 に状態フラグを示します。

表 39-1 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、整合性がないので、fsck 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 を実行する。その後で、ルート (/) ファイルシステムを読み取り/書き込みとしてマウントし直す。
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 ロギングを有効にしてファイルシステムがマウントされている場合、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 でもなければ、mount (2) システムコールではファイルシステムは読み取り/書き込み用にマウントされない。

表 39-2 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 39-2 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
FSACTIVE	FSSTABLE	FSSTABLE	FSACTIVE
FSSTABLE	FSSTABLE	FSSTABLE	FSACTIVE
FSCLEAN	FSCLEAN	FSSTABLE	FSACTIVE
FSBAD	FSSTABLE	FSSTABLE	FSBAD
FSLOG	FSLOG	FSLOG	FSLOG

fsck 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 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。

非整合状態が発生する原因

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るので、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、実際へ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、CPU を停止したり、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` が適用する整合性チェックの種類について説明します。

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ます。

スーパーブロックの非整合性は、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。ファイルシステムのサイズとレイアウト情報は、fsck にとって最も重要な情報部分です。これらのサイズを実際にチェックする方法はありませんが、ファイルシステムの作成時に静的に判断されるので、fsck はサイズが妥当な範囲内にあるかどうかをチェックできます。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 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。

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 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 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。ブロック割り当てマップ内に間違いがあると、fsck 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 プログラムは、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 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 プログラムは、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 ノード数

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 は予約されています)。各 i ノードの非整合性は、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 ノードのフォーマットとタイプ

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 6 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)
- 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で空のはずです。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 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 プログラムは、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 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 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 はリード内の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 は *i* ノード内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 プログラムは、*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 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

プログラムは、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` プログラムは、*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` 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` 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` 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。非整合性は、次の面からチェックできます。

- 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整合性チェックも実行されます。

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 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非整合性は、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 はそのディレクトリエントリを削除します。この状況は、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 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트리

「.」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트리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트리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 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と、最初に見つかった方が「..」が指す実際の親であると見なされます。この場合、fsck 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 プログラムは、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 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 (i ノードはファイルシステムに書き込まれてたが、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と、この状態が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 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 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/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
```

(続く)

```

** 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	フラグメント率。ファイルシステム内の空きフラグメント x 100 / 全フラグメント

ブート時のファイルシステムチェック機能の変更

ブート中には、ブートスクリプト `/sbin/rcS` を使用して、ハードディスクからマウントされるファイルシステムごとに予備チェックが実行されます。このスクリプトは、ルート (`/`)、`/usr` および `/var` のファイルシステムをチェックします。他の `rc` シェルスクリプトが、`fsck` コマンドを使用して他のファイルシステムを順番にチェックします。ファイルシステムが並行にチェックされることはありません。

「`fsck pass`」の数値が 1 より大きい場合も、ブート中にはファイルシステムは順番にチェックされます。

/etc/vfstab ファイル

ファイルシステムを直接指定せずに、ファイルシステムをチェックしたりマウントするコマンドを実行すると、各コマンドは、ファイルシステムテーブル (/etc/vfstab) を調べ、各種フィールドで指定された情報を使用して処理を行います。fsck pass フィールドは、ファイルシステムチェックのための情報を指定します。「mount at boot」フィールドは、ブート時にファイルシステムをマウントするための情報を指定します。

新しいファイルシステムを作成する場合は、ブート時にチェックするかどうかとマウントするかどうかを示すエントリを /etc/vfstab に追加します。/etc/vfstab ファイルにエントリを追加する方法の詳細は、第 36 章を参照してください。

/etc/vfstab ファイル内の情報は、各システムのスライスとファイルシステムに固有です。次の例は、/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/c1d0s2 /dev/rdsk/c1d0s2 /usr      ufs     1        yes      -
/proc        -           /proc     proc    -        no       -
fd           -           /dev/fd   fd      -        no       -
swap         -           /tmp      tmpfs   -        yes      -

/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /      ufs     1        no       -
/dev/dsk/c0t0d0s1 -           -         swap    -        no       -
/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /usr     ufs     2        no       -
/dev/dsk/c0t0d0s7 /dev/rdsk/c0t0d0s7 /opt     ufs     3        yes      -
pluto:/usr/dist -           /usr/dist nfs     no       yes      -
$
```

表 39-3 に、「fsck pass」フィールドの機能を示します。

表 39-3 「fsck pass」フィールド

フィールドの設定	機能	備考
- (ハイフン)	ファイルシステムの状態に関係なく、汎用 fsck コマンドはファイルシステムをチェックしない。	読み取り専用ファイルシステム、リモートファイルシステム、/proc などの疑似ファイルシステムに対してはハイフンを指定してチェックを行わないように指示する。
0	ファイルシステム専用の fsck コマンドが呼び出される。	UFS ファイルシステムに対して 0 が指定された場合、ファイルシステムはチェックされない。
1 以上かつ fsck -o p を使用	ファイルシステム専用の fsck で、自動的に UFS ファイルシステムを並列にチェックする。	この値には 1 より大きい任意の数値を使用できる。

preen モードでは (-o p オプション)、fsck は 1 ディスクごとに 1 つしか有効なファイルシステムをチェックできず、あるチェックが完了した後でなければ次のチェックを開始しません。fsck は、ファイルシステムが存在するデバイスのメジャー番号とマイナー番号を使用して、自動的に異なるディスク上のファイルシステムを同時にチェックする方法を決定します。

fsck pass 番号が 1 であれば、ファイルシステムは /etc/vfstab ファイルに表示される順番にチェックされます。通常、ルート (/) ファイルシステムの fsck pass は 1 に設定されます。

注 - fsck は、fsck pass 番号を使用せずにファイルシステムのチェック順を決定します。

▼ ブート中に実行されるファイルシステムチェック機能を変更する方法

1. スーパーユーザーになります。
2. 「fsck pass」フィールド内の /etc/vfstab エントリを編集し、編集結果を保存します。
次にシステムをブートしたときに、新しい値が使用されます。

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
- 使用中に問題が発生する場合

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` を使用する前に、571ページの「`fsck` コマンドの構文とオプション」と『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」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どうかを調べる方法

1. スーパーユーザーになります。
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/rdisk/device-name
```

このコマンドで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、ファイルシステムがクリーンであるか、あるいは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` 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どうかを調べる

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/rdsk/c0t0d0s6
** /dev/rdsk/c0t0d0s6
ufs fsck: sanity check: /dev/rdsk/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ります。
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」フィールド内のエントリが 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 の引数として、マウントポイントディレクトリや /dev/rdsk/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。エラーメッセージのプロンプトに回答して 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する方法については、『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」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 を実行すると、fsck がなんらかの変更を行なった場合、システムがクラッシュする可能性があります。ただし、たとえばシングルユーザーモードで fsck 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 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 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、570ページの「fsck で修復できない UFS ファイルシステムを修正する方法」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 によって lost+found ディレクトリに入れられた各ファイル名は、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 に書き出さ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方式でチェックする

次の例では、/dev/rdisk/c0t0d0s6 がチェックされ、不正なブロック数が訂正されます。

```
# fsck /dev/rdisk/c0t0d0s6
checkfileysys: /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 の修復オプション (-o p) は、UFS ファイルシステムをチェックし、単純な問題を自動的に修正します。通常、この種の問題は予期しないシステム停止によるものです。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。また、修復オプションを使用する場合には、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。

状態がクリーンにマークされずにシステムが停止した後のファイルシステムの修復にも、-o p オプションを指定して fsck を実行することができます。このモードでは、fsck 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 を対話形式で実行した場合の処理のサブセットです。

▼ 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ります。
2. 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` 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 — ファイルシステムを修復する

次の例では、`/usr` ファイルシステムが修復されます。

```
# fsck -o p /usr
```

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` 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冗長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` を使用すると、スーパーブロックをそのコピーで置き換えることができます。

▼ 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ります。
2. データが破壊されたファイルシステム上のディレクトリがカレントディレクトリになっている場合は、カレントディレクトリを変更します。

3. 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。

4. `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/rdsk/device-name
```

このコマンドの出力には、`newfs` 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、589ページの「カスタムファイルシステムパラメータの決定」を参照してください。

5. `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/rdsk/device-name
```

`fsck` 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` 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

次の例では、`/files7` ファイルシステムのスーパーブロックがブロック番号 5264 に存在するコピーを使って復元されます。

```
# cd /
# umount /files7
# newfs -N /dev/rdsk/c0t3d0s7
/dev/rdsk/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,
```

(続く)

```
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 *****
#
```

ルート (/) ファイルシステム中のスーパーブロックが破壊されていてシステムをブートできない場合は、/kernel/unix を再インストールして newfs コマンドでルートファイルシステムを再構築してください。スーパーブロックは newfs コマンドによって作成されるので、ユーザーが復元する必要はありません。

fsck で修復できない UFS ファイルシステムを修正する方法

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ります。fsck 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 はクリーンになるまで動作を続けるわけではないので、手作業で実行しなければなりません。

fsck 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、メッセージは不正なディレクトリを指す場合があります。そのディレクトリを削除すると、fsck 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 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媒体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 44 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` 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(1M)` 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。新しいディスクをインストールする方法については、第 30 章または第 31 章を参照してください。

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非整合状態をチェックして修復します。次の 4 つのオプションがあります。

- ファイルシステムがマウント可能かどうかのチェックだけを行う (`fsck -m`)
- 修復する前に確認を促すプロンプトを表示する (`fsck`)
- すべての修復時のプロンプトに `yes` または `no` の応答が指定されたものとして処理を行う (`fsck -y` または `fsck -n`)
- 確認を促す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) 非整合箇所をすべて修復 (`preen`) するが、重大な問題にぶつかると終了する (`fsck -o p`)

汎用 fsck コマンドの構文、オプション、引数

`fsck` には、汎用コマンドとファイルシステム固有のコマンドの 2 種類があります。汎用コマンドはほとんどのタイプのファイルシステムに使用でき、固有のコマンドは特定のタイプのファイルシステムにのみ使用できます。通常は固有コマンドを使用してください。汎用コマンドは必要に応じてファイルシステム固有のコマンドを呼び出します。

通常、`fsck` 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。スーパーユーザーにならなくても `fsck` コマンドの実行はできますが、修復するにはファイルシステムをマウント解除しなければなりません。また、スライスの `raw` デバイスファイルの読み取り権を持っていないければなりません (セキュリティ上の問題のため、一般のユーザーには `raw` デバイスファイルの読み取り権はありません)。

汎用 `fsck` コマンドは、`/etc/vfstab` にアクセスして、どのファイルシステムをチェックするかを調べます。`fsck pass` 番号に `-` (ハイフン) が指定されている任意のファイルシステムと、`0` が指定されている `UFS` のファイルシステムを除き、指定された各ファイルシステム上で、該当するファイルシステム専用の `fsck` コマンドを実行します。

汎用 `fsck` コマンドの構文は次のとおりです。

```
/usr/sbin/fsck [-F type] [-V] [-m] [special]
/usr/sbin/fsck [-F type] [-V] [-y|Y] | [n|N] [-o specific-options] [special]
```

表 39-4 に、汎用 `fsck` コマンドのオプションと引数を示します。

表 39-4 `fsck` コマンドのオプションと引数

オプションのタイプ	オプション	説明
汎用	<code>-F</code>	ファイルシステムのタイプ (<code>type</code>) を指定する。コマンド行で <code>type</code> を指定しない場合は、 <code>/etc/vfstab</code> ファイル内のエントリを、指定した <code>raw</code> デバイス名と照合して取得される。エントリが見つからない場合は、 <code>/etc/default/fs</code> 内で指定されたデフォルトのローカルのファイルシステムのタイプが使用される。
	<code>-V</code>	実行される完全なコマンド行が表示される (詳細モード)。表示行には、 <code>/etc/vfstab</code> から取り出された追加情報が含まれる。このオプションを使用すると、コマンド行を検査して有効性を確認できる。コマンド自体は実行されない。
	<code>-m</code>	予備チェックのみを実行する。ファイルシステムの状態を示すコードを返す。 <code>0</code> は「クリーン」を示し、 <code>32</code> は「ダーティ」を示す。起動スクリプト <code>/sbin/rcS</code> は、このオプションを使用して、ファイルシステムのチェックが必要かどうかを判断する。

表 39-4 fsck コマンドのオプションと引数 続く

オプションのタイプ	オプション	説明
	-y、-Y、-n、または -N	実行されるコマンドのすべてのプロンプトに対して、自動的に、yes または no で応答する。
	c	静的にテーブルが割り当てられている古い SunOS 4.1 以前のファイルシステムを、新しい動的に割り当てられたテーブルに変換する。静的割り当ては最大テーブルサイズにハード制限が適用されることを示し、動的割り当ては、初期割り当ての後に必要に応じてテーブル用の領域を追加できることを意味する。ファイルシステムが新しい形式であれば、テーブル割り当てが古い形式で許される固定最大サイズを超えない限り、古い形式に変換される。fsck は変換方法を表示する。対話モードでは、fsck は変換前に確認を促すプロンプトを表示する。-o p オプションを使用すると、確認を促すプロンプトは表示されずに変換試行が実行される。このオプションは、多数のファイルシステムを一度に変換したい場合に便利である。ファイルシステムが古い形式か新しい形式かは、fstyp(1M) コマンドを実行し、最初に表示される行を調べれば判断できる。
	w	書き込みアクセスできるファイルシステムのみをチェックする。
	<i>special</i>	1 つまたは複数のファイルシステムのマウントポイントまたは raw デバイス名を指定する。マウントポイントのエントリは、/etc/vfstab 内になければならない。 <i>special</i> 引数を省略すると、/etc/vfstab 内で fsck デバイスが指定されていて、fsck pass の値が 0 より大きいエントリがチェックされる。修復 (-o p) が有効で、fsck pass の値が 1 より大きいエントリが複数個あると、異なるディスク上のファイルシステムは並列してチェックされる。
専用		-o オプションに続けて、オプションをコンマで区切ったリスト。UFS 固有の fsck コマンドが解釈できるように渡されるオプションを指定する。

表 39-4 fsck コマンドのオプションと引数 続く

オプションのタイプ	オプション	説明
	p	修復。コマンドはプロンプトを表示せずに実行され、検出されたエラーは自動的に訂正されるが、オペレータの介入が必要な問題が見つかったと終了する。このオプションを使用すると、UFS ファイルシステムを並列にチェックすることもできる。
	b=blocknumber	指定した位置にある代替 (冗長) スーパーブロックを使用する。このオプションを使用すると、不良スーパーブロックを修復できる。newfs -N コマンドを使用すると、代替スーパーブロックのリストを表示できる。

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。

- 575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」
- 585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」
- 589ページの「カスタムファイルシステムパラメータの決定」
- 594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 40-4 を参照してください。

表 40-1 および表 40-2 に、ルート (/) のデフォルトファイルシステム中のディレクトリを示します。

表 40-1 ルート (/) ファイルシステムのデフォルトディレクトリ (その 1)

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	特殊ファイルの一次位置
/dev/cfg	物理 ap_id へのシンボリックリンク
/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/md	論理ボリューム管理メタディスクデバイス
/dev/fd	ファイル記述子
/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
/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/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

表 40-1 ルート (/) ファイルシステムのデフォルトディレクトリ (その 1) 続く

ディレクトリ	説明
/etc/dmi	Solstice Enterprise Agents™ の構成ファイル
/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/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	/usr をマウントする前に必要な処理のためにファイルシステムタイプ別に編成されたバイナリ
/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
/etc/inet	インターネットサービスの構成ファイル
/etc/init.d	実行レベルを切り替えるためのスクリプト
/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/mail	メールサブシステムの構成情報
/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/rc0.d	実行レベル 0 を開始/停止した時に起動されるスクリプト

表 40-1 ルート (/) ファイルシステムのデフォルトディレクトリ (その 1) 続く

ディレクトリ	説明
/etc/rc1.d	実行レベル 1 を開始/停止した時に起動されるスクリプト
/etc/rc2.d	実行レベル 2 を開始/停止した時に起動されるスクリプト
/etc/rc3.d	実行レベル 3 を開始/停止した時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で起動するスクリプト
/etc/rpcsec	このディレクトリは NIS+ 認証の構成ファイルに含まれていることがある。
/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/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/tm	商標ファイル。内容はブート時に表示される。
/etc/uucp	uucp 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 40-4 を参照。

表 40-2 ルート (/) ファイルシステムのデフォルトディレクトリ (その 2)

ディレクトリ	説明
/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
/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
/stand	スタンドアロンプログラム
/tmp	一次ファイル。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。詳細は表 40-3 を参照してください。
/var	常に変化するファイル (一時ファイル、ロギングファイル、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/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/ftp	FTP サーバーのディレクトリ
/var/inet	IPv6 ルーターの状態ファイル
/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

表 40-2 ルート (/) ファイルシステムのデフォルトディレクトリ (その 2) 続く

ディレクトリ	説明
/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 (注: USENET 方式のニュースとは異なる)
/var/nis	NIS+ データベース
/var/nfs	NFS サーバーのログファイル
/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。
/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リングファイル
/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/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/uucp	待ち行列に入っている uucp のジョブ

表 40-2 ルート (/) ファイルシステムのデフォルトディレクトリ (その 2) 続く

ディレクトリ	説明
/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 (NIS との下位互換性を保つため。NIS+ への移行が完了した後は不要)

次の表に、/usr ファイルシステムのデフォルトのディレクトリを示します。

表 40-3 /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
adm	/var/adm ディレクトリへのシンボリックリンク
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

表 40-3 /usr ファイルシステムのデフォルトディレクトリ 続く

ディレクトリ	説明
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	ヘッダーファイル (C プログラム用など)
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	実装されたアーキテクチャ固有のバイナリとライブラリ
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) 実行環境に関するファイル
old	段階的に使用されなくなっているプログラム
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

表 40-3 /usr ファイルシステムのデフォルトディレクトリ 続く

ディレクトリ	説明
perl5	perl 5 のプログラムとマニュアル
platform	詳細は表 40-4 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/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
share	アーキテクチャに依存しない共有可能ファイル
share/lib	アーキテクチャに依存しないデータベース
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 40-3 /usr ファイルシステムのデフォルトディレクトリ 続く

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 40-4 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 40-4 /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。kernel (1M) 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。削除された /usr/kvm の内容に置き換わるオブジェクトを含む。
/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。
/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	説明
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

この節では、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プログラムが格納されます。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、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。その中でも、特に次の情報が重要です。

- 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)
- 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: クリーン、安定、または有効
- 最後のマウントポイントのパス名

スーパーブロックは、ディスクスライスの先頭にあり、各シリンダグループ内で複製されます。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。各スーパーブロックの複製は、シリンダグループの先頭からさまざまな大きさだけオフセットされます。複数プラッタを持つディスクドライブの場合、オフセットはスーパーブロックがドライブの各プラッタに表示されるように計算されます。つまり、最初のプラッタが失われても、いつでも代替スーパーブロックを取り出せます。最初のシリンダグループ内の先行ブロックを除き、オフセットによって作成される先行ブロックがデータの格納に使用されます。

集計情報ブロックは、スーパーブロックといっしょに保管されます。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。集計ブロックレコードには、ファイルシステムの使用時に発生した変化が記録され、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

- シンボリックリンク
- FIFO (名前付きパイプとも呼ばれます)
- ソケット
- ファイルのモード (読み込み権-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、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 番目のアドレスが必要な場合は、三重間接アドレスが格納されます。図 40-1 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。

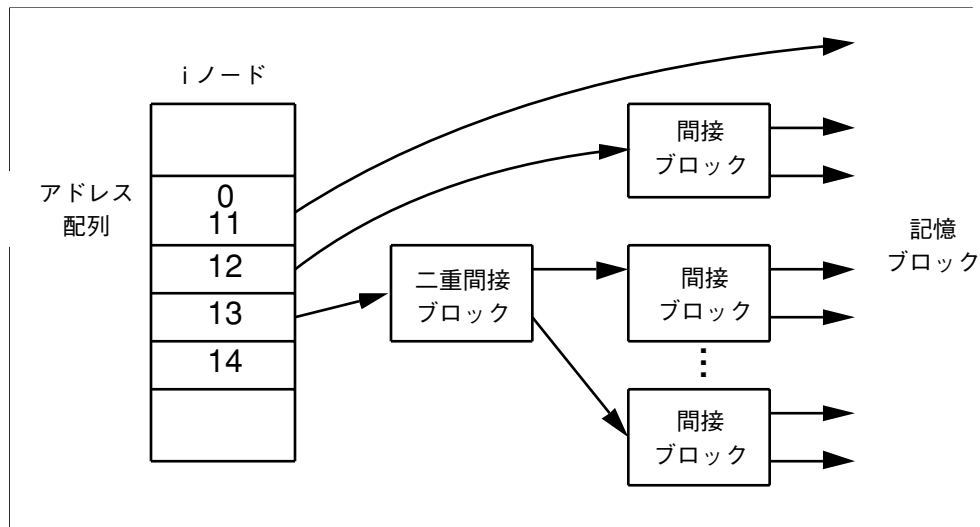


図 40-1 UFS システム内のファイルシステムアドレスチェーン

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、データブロックが入っています。この種のブロックは、記憶ブロックと呼ばれることもあります。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、ディスクのパフォーマンス低下を防止します。

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、図 40-2 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。

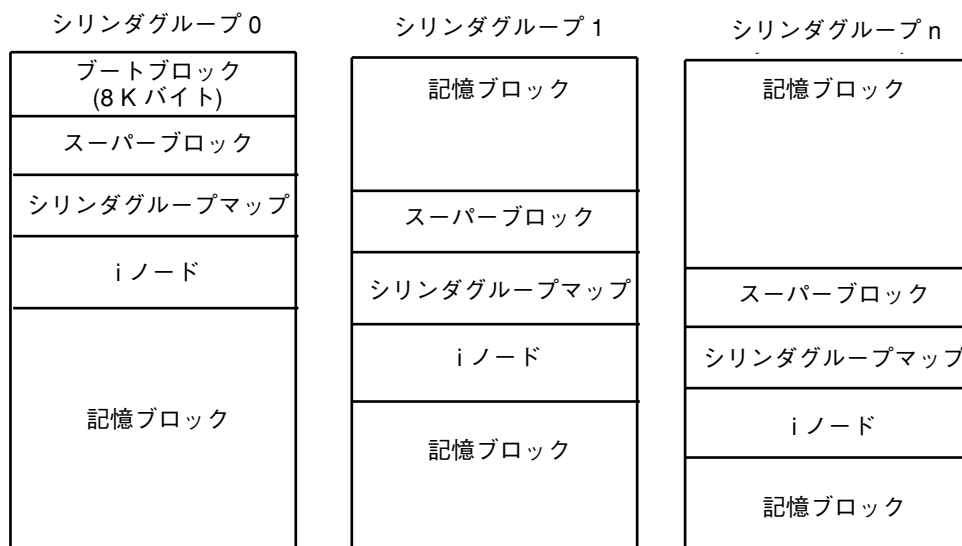


図 40-2 一般的な UFS ファイルシステム

カスタムファイルシステムパラメータの決定

`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。

- ブロックサイズ
- フラグメントサイズ
- 最小空き領域
- 回転の遅れ
- 最適化タイプ
- iノード数

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 (通常は 512K

バイト)とは異なります。物理ブロックサイズは、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。

ファイルシステムの論理ブロックサイズを指定できます。ファイルシステムの作成後は、ファイルシステムを再構築しなければ、このパラメータを変更できません。論理ブロックサイズの異なるファイルシステムを、同じディスクに格納できます。

デフォルトでは、UFS ファイルシステムの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。8K バイトは、論理ブロックの推奨サイズです。

注 [SPARC] - 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。

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。ファイルのデータを保持する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ディスク容量の「フラグメント」が低下し、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 512K 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。

注 - 上限を、まったくフラグメントのない場合の完全ブロックのサイズに等しくすることができます。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、その値を最も近い整数に切り捨てます。値は、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。tuneufs 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。

回転の遅れ (ギャップ)

「回転の遅れ」は、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予想される最小所要時間 (ミリ秒単位) です。デフォル

トの遅れはゼロです。これは、最近のオンディスクキャッシュと併用すると、遅れに基づく計算が無効になるためです。

ファイルに書き込むときに、UFS 割り当てルーチンは新しいブロックを同じファイル内の直前のブロックと同じディスクシリンダ上に配置しようとします。また、新しいブロックをトラック内で最適な位置に配置して、そこへのアクセスに必要なディスクの回転を最小限度に抑えようとします。

ファイルブロックを「回転して適切に動作」するように配置するには、割り当てルーチンは CPU による転送処理速度と、ディスクが 1 ブロックをスキップする所要時間を認識しなければなりません。mkfs コマンドのオプションを使用すると、ディスクの回転速度と 1 トラック当たりのディスクブロック (セクター) 数を指定できます。割り当てルーチンは、この情報を使用して、1 ディスクブロックをスキップするミリ秒数を求めます。次に、割り当てルーチンは予想転送時間 (回転の遅れ) を使用して、システムの読み込み準備ができたときに次のブロックがディスクヘッドの真下にくるようにブロックを配置します。

注 - 回転の遅れ (newfs の `-d` オプション) を指定しなくてもよいデバイスがあります。

各ブロックは、システムが同じディスクの回転中に読み込める処理速度の場合のみ、連続して配置されます。システムが低速であれば、ディスクはファイル内の次のブロックの先頭を通り過ぎてしまうので、そのブロックを読み込むには、もう 1 回転しなければならず、長時間かかります。次のディスク要求が発生するときに該当するブロックにヘッドがきているように、ギャップに適切な値を指定してください。

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このパラメータの値を変更できます。変更結果は、それ以後のブロック割り当てにのみ適用され、すでに割り当て済みのブロックには適用されません。

最適化のタイプ

「最適化のタイプ」には、「領域」と「時間」があります。

- 領域 - 領域の最適化を選択すると、フラグメントを最小限度に抑え、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。
- 時間 - 時間の最適化を選択すると、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、それほど細かく分割しなくても、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「時間」です。

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ます。

ファイルの数

i ノード数によって、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合計 i ノード数が決まります。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	8192

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。

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 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。`newfs` コマンドは、`/usr/sbin` ディレクトリに入っています。

構文は次のとおりです。

```
newfs [-Nv] [mkfs_options] raw_device
```

表 40-5 に、`newfs` コマンドのオプションと引数を示します。

表 40-5 `newfs` コマンドのオプションと引数

オプション	説明
<code>N</code>	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。
<code>-v</code>	<code>mkfs</code> コマンドに渡されるパラメータが表示される。
<code>mkfs-options</code>	後続のオプションを使用して、 <code>mkfs</code> コマンドに渡されるパラメータが設定される。次のオプションは、 <code>mkfs</code> 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。
<code>-s size</code>	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。
<code>-t ntrack</code>	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。

表 40-5 newfs コマンドのオプションと引数 続く

オプション	説明
-b <i>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)。
-f <i>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)。
-c <i>cgsiz</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。
-m <i>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、 $((64M \text{ バイト} / \text{パーティションサイズ}) * 100)$ で算出した値を最も近い整数に切り捨てます。値は、ディスク容量の 1% から 10% の範囲に制限されます。
-r <i>rpm</i>	1 分当たりのディスクの回転数。この設定はドライブ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 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 に渡される前に 1 秒当たりの回転数に変換される。
-i <i>nbpi</i>	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、上記の節を参照。
-o <i>opt</i>	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。 <i>opt</i> には <i>time</i> または <i>space</i> を指定する。デフォルトは <i>time</i> です。
-a <i>apc</i>	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。
-d <i>gap</i>	(回転の遅れ) 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予想される最小ミリ秒数。デフォルトは 0。
-d <i>nrpos</i>	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。

表 40-5 newfs コマンドのオプションと引数 続く

オプション	説明
-C <i>maxcontig</i>	<p>あるファイルに属し、回転の遅れ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) バッファータを持たないドライブ (または、内部バッファータが存在することを示していないドライブ/コントローラ) の場合は、デフォルトは 1 で、バッファータを持つドライブの場合はデフォルトは 7。</p> <p>このパラメータは、次のようにする必要がある。</p> <p><i>blocksize</i> x <i>maxcontig</i> <= <i>maxphys</i></p> <p><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) を指定する読み込み専用のカーネル変数である (この制限は、<i>newfs</i> や <i>mkfs</i> ではなく <i>mount</i> によって適用される)。</p> <p>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。詳細は、<i>tunefs(1M)</i> のマニュアルページを参照。</p>
<i>raw_device</i>	<p>ファイルシステムを入れるパーティションの特殊文字 (<i>raw</i>) デバイスファイル名。この引数は必須。</p>

例 — newfs コマンドのオプションと引数

次の *newfs* の例では、-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、ファイルシステム情報を表示します。

```
# newfs -N /dev/rdsk/c0t0d0s0
/dev/rdsk/
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* 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* では各種のファイルシステムがサポートされますが、実際には UFS 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` コマンドを使用できるようにソフトウェアを作成する必要があります。通常は、`mkfs` を直接実行せずに、`newfs` コマンドで呼び出します。

汎用 `mkfs` コマンドは、`/usr/sbin` にあります。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。

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。

大容量入出力処理の例としては、大量のデータがファイルに書き込まれる衛星データのダウンロードなどがあります。直接入出力では、オペレーティングシステムのページキャッシュ機構のオーバーヘッドは使用されず、データの書き込みと読み取りが直接メモリーとで行われます。

直接入出力の起動には潜在的な欠点があります。入出力を要求したファイルがすでに別のアプリケーションによってマップされていた場合、そのページがメモリーからフラッシュされるまで直接入出力処理を開始できません。

詳細は、`directio(3C)` のマニュアルページを参照してください。

`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。詳細は、`mount_ufs(1M)` のマニュアルページを参照してください。

▼ 直接入出力を UFS ファイルシステムで有効にする方法

1. スーパーユーザーになります。

2. mount コマンドの forcedirectio オプションでファイルシステムをマウントします。

```
# mount -F ufs -o forcedirectio /dev/dsk/c0t3d0s7 /datab
```

3. マウントされたファイルシステムで直接入出力が有効であることを確認します。

```
# mount
      .
      .
      .
/export/home on /dev/dsk/c0t3d0s7 read/write/setuid/forcedirectio ...
```

データのバックアップと復元

ここでは、Solaris オペレーティング環境でデータのバックアップを取る方法とデータを復元する方法について説明します。次の章で構成されています。

第 42 章	ufsdump コマンドと ufsrestore コマンドを使用してデータのバックアップを取り、復元する際のガイドラインと計画について説明します。
第 43 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 44 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 45 章	ufsdump の機能、ufsdump コマンドと ufsrestore コマンドの構文とオプションについて説明します。
第 46 章	ファイルシステムをディスクにコピーする手順、dd、cpio、tar コマンドをさまざまなバックアップ媒体で使用する手順、ファイルを異なるヘッダー形式でコピーする手順について説明します。
第 47 章	テープドライブを追加する方法、テープドライブ、バックアップデバイス名を判断し、テープドライブと磁気テープカートリッジで作業する手順を説明します。

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、ファイルシステム全体のバックアップをとって復元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。

この章の内容は以下のとおりです。

- 601ページの「バックアップと復元についての参照先」
- 602ページの「ファイルシステムのバックアップと復元とは」
- 603ページの「ファイルシステムをバックアップしなければならない理由」
- 603ページの「テープデバイスの選択」
- 604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」
- 607ページの「バックアップコマンドと復元コマンドの概要」
- 608ページの「バックアップタイプの選択」
- 609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」
- 611ページの「バックアップスケジュールの例」

バックアップと復元についての参照先

`ufsdump` コマンドと `ufsrestore` コマンドを使用してファイルシステムをバックアップおよび復元する手順については、次を参照してください。

- 第 43 章
- 第 44 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取り外し可能な媒体 (テープなど) 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取り外し可能な媒体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、「スケジュールされた」バックアップ処理と復元処理に使用するコマンド (ufsdump と ufsrestore) について説明しますが、他にもファイルを共有または転送するためにファイルとファイルシステムのコピーに使用できるコマンドがあります。表 42-1 に、個々のファイルやファイルシステムを媒体にコピーする各コマンドの説明箇所を示します。

表 42-1 ファイルとファイルシステムをコピーするコマンド

操作	使用するコマンド	参照先
ファイルシステム全体または個々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コピーする	ufsdump(1M) コマンド	第 43 章 または 第 45 章
ネットワーク上のすべてのシステムのファイルシステム全体を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(1M) コマンド と nisrestore(1M) コマンド	『Solaris ネーミングの管理』
ファイルをテープ上でコピー、表示、検索する	tar(1)、cpio(1)、pax(1) コマンド	第 46 章
ファイルをフロッピーディスク上でコピー、表示、検索する	tar(1) コマンド	

表 42-1 ファイルとファイルシステムをコピーするコマンド 続く

操作	使用するコマンド	参照先
マスターディスクをクローンディスクにコピーする	dd(1M) コマンド	第 46 章
ファイルシステム全体または個々のファイルシステムを、取り外し可能な媒体から作業ディレクトリに復元する	ufsrestore(1M) コマンド	第 44 章

ファイルシステムをバックアップしなければならない理由

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)
- システムをインストールし直したりアップグレードするときの問題

テープデバイスの選択

表 42-2 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。テープデバイスについての詳細は、第 47 章を参照してください。

表 42-2 ファイルシステムのバックアップに使用する典型的な媒体

媒体	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 4-mm カートリッジテープ (DAT)	12 ~ 24 G バイト
14-Gbyte 8-mm 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。表 42-3 と表 42-4 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 42-3 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	理由	バックアップ間隔
ルート (/) - パーティション 0	ルート (/) には、メールやアカウントリングなど、頻繁に変更されるファイルが保管されるカーネルと /var ディレクトリが入っている。	定期
/usr - パーティション 6	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 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

表 42-3 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 続く

バックアップするファイルシステム	理由	バックアップ間隔
/export/home	/export/home 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とサブディレクトリが入っている。	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 1 度
/export または /var を他の使用可能スライスにバックアップ	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 42-4 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 ...	理由	バックアップ間隔
ルート (/) - パーティション 0 /export - パーティション 3 /usr - パーティション 6, /opt	これらのファイルシステムには、カーネル、主要コマンド、実行可能プログラムが入っている。	サイトのニーズに応じて毎日 1 度ないし月に 1 度 ルート (/) - ネットワーク上でクライアントとハードウェアを頻繁に追加したり削除したりする場合は、カーネル構成ファイルなどの重要なファイルをルート (/) 内で変更せざるを得ない。この場合は、週に 1 度から月に 1 度の間隔で、ルート (/) ファイルシステム上で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いる場合は (その後クライアントシステムがマウント)、ルート (/) ディレクトリを毎日バックアップした方がよい。(あるいは、別のファイルシステムの場合は /var) /export - クライアントのルート (/) ディレクトリは、/export ファイルシステムに保管される。そこに入っている情報はサーバーのスライス 0 にあるルートディレクトリと同様なので、頻繁に変化することはない。随時バックアップを実行すればよい。ただし、サイトでメールをクライアントシステムに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。 /usr と /opt - 内容に変更がなく、週に 1 度ないし月に 1 度バックアップするだけでかまわない。
/export/home - パーティション 7	/export/home には、システム上のユーザー全員のホームディレクトリとサブディレクトリが入っている。そのファイルは変更が多い。	毎日または毎週

注 - サーバーの /export/swap のバックアップをとる必要はありません。

バックアップコマンドと復元コマンドの概要

`ufsdump` コマンドと `ufsrestore` コマンドは、ファイルシステム全体のバックアップを定期的に行う場合の推奨コマンドです。表 42-5 に、この 2 つのコマンドで実行できる作業を示します。各コマンドの機能と構文については、第 45 章を参照してください。

表 42-5 `ufsdump` と `ufsrestore` で実行できる作業

コマンド	実行できる作業	備考
<code>ufsdump</code>	ファイルシステム全体または一部のファイルシステムのバックアップを、ローカルまたはリモートのテープドライブに作成する。	テープデバイスは、ネットワーク内でユーザーがアクセスするどのシステム上にあってもかまわない。このコマンドは UFS ファイルシステムタイプの構造を認識し、 <code>raw</code> デバイスインターフェース経由で直接処理するので高速である。
	増分バックアップの実行	これによって、前回のバックアップ以降に変更があったファイルのバックアップのみを作成できる。
	単一システムからネットワーク経由でシステムグループをバックアップする。	あるシステムから、リモートシェルまたはリモートログインを通して、各リモートシステム上で <code>ufsdump</code> を実行し、ドライブがあるシステムに出力を転送できる。また、出力をパイプを通して <code>dd</code> コマンドまたはファイルに渡すことができる。
	自動バックアップ	<code>crontab</code> ユーティリティを使用して、 <code>ufsdump</code> コマンドを起動するスクリプトを実行する。
	バックアップテーブルへのユーザーアクセスを制限する。	<code>-a</code> オプションを使用する。
	実際にバックアップを実行せずに、バックアップのサイズを決定する。	<code>-s</code> オプションを使用する。
	ファイルシステムがバックアップされたときのログを保管する。	<code>-u</code> オプションを使用する。

表 42-5 ufsdump と ufsrestore で実行できる作業 続く

コマンド	実行できる作業	備考
	テープの内容をソースファイルシステムと対照して検査する。	-v オプションを使用する。
ufsrestore	個々のファイルシステムまたはファイルシステム全体を、ローカルまたはリモートのテープドライブから復元する	

バックアップタイプの選択

ufsdump コマンドを使用すると、完全バックアップまたは増分バックアップを実行できます。表 42-6 に、この 2 つのタイプのバックアップの違いを示します。

表 42-6 完全バックアップと増分バックアップの違い

バックアップのタイプ	コピー内容	長所	短所
完全	ファイルシステムまたはディレクトリ全体	すべてを 1 箇所にとめることができる。	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。
増分	前回のバックアップ以降に変更があり、ファイルシステム上で指定したファイルのみ	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。

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を作成する際に検討すべき要素、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、バックアップスケジュールの例について説明します。

バックアップスケジュールに影響する要素

どのようなスケジュールを設定するかは、次の要素によって左右されます。

- テープの本数を最小限度に抑える必要があるかどうか
- 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媒体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、これ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、各組を毎月使い廻ることになります。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。1 から 9 までの番号は、増分バックアップのスケジュール設定に使用されますが、特に意味が定義されているわけではありません。これら

の番号は、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。

注 - 次の例で、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になります。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。

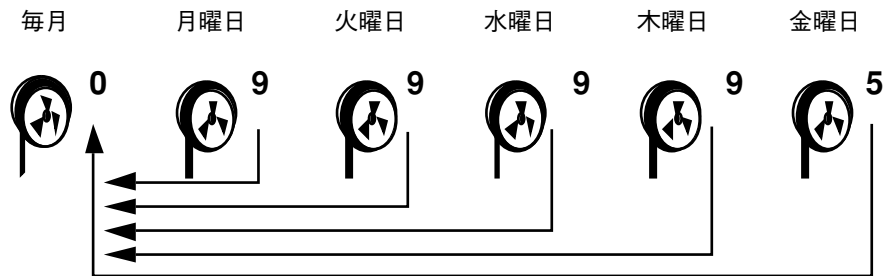


図 42-1 増分バックアップ: 日単位累積

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。

注 - 次の例では、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おらず、その大小に意味があります。

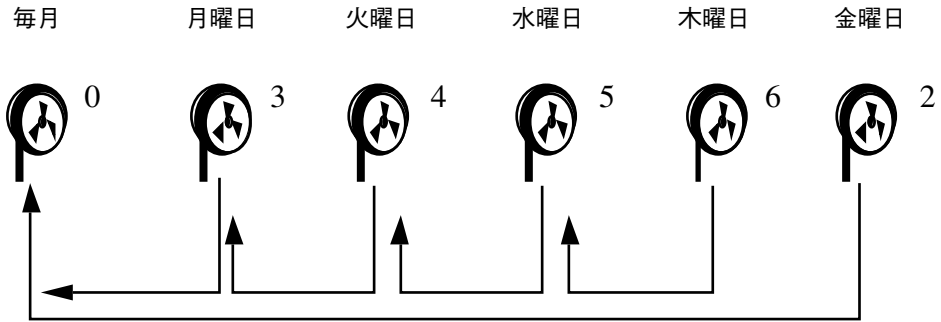


図 42-2 増分バックアップ: 日単位個別

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 (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

表 42-7 に、最も一般的に使用される増分バックアップを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 42-7 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

表 42-8 は、前のスケジュールを使用して、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。各文字は別のファイルを表します。

表 42-8 日単位/週単位累積スケジュールのテープの内容

	月曜日	火曜日	水曜日	木曜日	金曜日
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

必要なテープの本数

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、金曜日用に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

表 42-9 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 42-9 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

表 42-10 は、以前のスケジュールからテープの内容が 2 週間でのどのように変化するかを示しています。各文字はそれぞれ異なるファイルを表します。

表 42-10 日単位累積/週単位増分スケジュールのテープの内容

	月曜日	火曜日	水曜日	木曜日	金曜日
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

必要なテープの本数

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、金曜日用に 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、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

表 42-11 は、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っており、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っているスケジュールを示しています。

表 42-11 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 2 週	3	4	5	6	2
第 3 週	3	4	5	6	2
第 4 週	3	4	5	6	2

表 42-12 は、テープの内容が 2 週間でどのように変化するかを示しています。各文字はそれぞれ異なるファイルを示します。

表 42-12 日単位累積/週単位累積バックアップスケジュールのテープの内容

	月曜日	火曜日	水曜日	木曜日	金曜日
第 1 週	a b	c d	e f g	h	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

必要なテープの本数

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。その内訳は、レベル 0 に 1 本、金曜日用に 4 本、日単位テープ 4 本ですが、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おり、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。

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 — サーバーのバックアップスケジュール

表 42-13 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 42-13 サンプルサーバーの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
/	第1日曜日	0	<i>n</i> 本のテープ
/usr	第1日曜日	0	"
/export	第1日曜日	0	"
/export/home	第1日曜日	0	"
	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
/	第2日曜日	0	<i>n</i> 本のテープ
/usr	第2日曜日	0	"

表 42-13 サンプルサーバーのバックアップスケジュール 続く

ディレクトリ	日付	レベル	テープ名
/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
/	第 3 日曜日	0	n 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
	第 3 金曜日	9	Q
	第 3 土曜日	5	R
	/	第 4 日曜日	0
/usr	第 4 日曜日	0	"

表 42-13 サンプルサーバーのバックアップスケジュール 続く

ディレクトリ	日付	レベル	テープ名
/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、/export/homeの4回の完全バックアップに必要な本数)に加えて、/export/home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は次のとおりです。

1. 日曜日ごとに、ルート(/)、/usr、/export、/export/home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のレベル9のバックアップを実行します。ufsdump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をコピーします。
3. 月の第1火曜日に、テープBを使用して/export/homeのレベル9のバックアップを実行します。この場合も、ufsdumpは、下のレベル、つまり日曜日のレベル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 は、下のレベルのバックアップ、つまり水曜日のレベル 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

表 42-14 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 42-14 システムのバックアップスケジュールに関する他の推奨事項

必要事項	推奨事項	備考
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している	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度に抑える必要がある	<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</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>これは、毎日の変更のみが各日単位テープに保存されることを意味する。</p> <p>これは、変更が (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ことを意味する。</p> <p>これは、<code>ufsdump</code> コマンドの <code>rewind</code> オプションを使用して実行される。</p>

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` コマンドを使用してファイルシステムのバックアップを行う手順について説明します。

この章の内容は以下のとおりです。

- 622ページの「ファイルシステム名を検索する方法」
- 622ページの「完全バックアップをとるために必要なテープの本数を決定する方法」
- 624ページの「テープにバックアップをとる方法」

`ufsdump` コマンドの構文、オプション、引数についての詳細は、第 45 章を参照してください。

バックアップを実行するための準備

ファイルシステムのバックアップを実行する準備は、計画を作成することから始まります。詳細は、第 42 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。

- テープドライブ
- バックアップを作成するファイルシステム
- 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

この節では、次のように、ファイルシステムのバックアップを作成する前に実行しておかなければならない作業について説明します。

- 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。
- 完全バックアップをとるため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 カラムに表示されるファイルシステム名を調べます。
3. ファイルシステムをバックアップするときには、mount point カラムのマウントポイントを使用することになります。

例 — ファイルシステム名を検索する

```
$ more /etc/vfstab
#device          device          mount          FS   fsck  mount  mount
#to mount        to fsck         point          type pass  at boot options
#
#/dev/dsk/c1d0s2 /dev/rdisk/c1d0s2 /usr           ufs   1     yes   -
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/c0t0d0s7 /dev/rdisk/c0t0d0s7 /export/home   ufs   2     yes   -
mars:/share/kit  -               /kit           nfs   -     yes   -
mars:/db/doc     -               /db/doc        nfs   -     yes   -
```

▼ 完全バックアップをとるために必要なテープの本数を決定する方法

1. スーパーユーザーになります。
2. usfdump s コマンドを使用して、バックアップのサイズをバイト単位で見積もります。

```
# ufsdump S filesystem
```

S バックアップの実行に必要な予想バイト数を表示する。

3. 予想バイト数をテープの容量で除算して、必要な本数を調べます。
テープの容量リストについては、表 42-2 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
```

バックアップの実行

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。
- 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。
- 1 台のシステムから 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、その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 46 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ドライブを指定するには、リモートテープドライブを持つシステムの 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を実行しているシステム上のリモートドライブには `/dev/rst0` を使用し、Solaris 8 またはその互換バージョンを実行しているシステムには `/dev/rmt/0` を使用する。

注 - Solaris 2.5 およびその互換バージョンが動作している NIS+ マスターサーバーをバックアップするには、`nisbackup` コマンドを使用して下さい。このコマンドを使用する方法については、『Solaris ネーミングの管理』を参照してください。

▼ テープにバックアップをとる方法

`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ります。
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] `fsck` コマンドを指定してファイルシステムの整合性をチェックします。

`-m` オプションを使用して `fsck` 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、ファイルが非整合状態になることがあります。`fsck` コマンドについての詳細は、第 39 章を参照してください。

```
# fsck -m /dev/rdsk/ device-name
```

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          ufsdump を使用してバックアップを実行するシステムの名前
```

- 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. 書き込み保護されていないテープをテープドライブに挿入します。
7. `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します。
次の表を参照して、`ufsdump` コマンドの最も一般的なオプションと引数を選択します。その他のオプションと引数については、第 45 章を参照してください。

操作	使用するオプションまたは引数	例	参照先
完全バックアップを実行する	0 オプション	<code>ufsdump 0ucf /dev/rmt/0 /</code>	627ページの「例 — 完全バックアップ、ルート (/)」
増分バックアップを実行する	1 から 9 までのオプション	<code>ufsdump 9ucf /dev/rmt/0 /</code>	627ページの「例 — 増分バックアップ、ルート (/)」
個々のファイルのバックアップを実行する	ファイルまたはディレクトリを指定する	<code>ufsdump ucf /dev/rmt/0 /export/home/kryten</code>	

操作	使用するオプションまたは引数	例	参照先
ダンプを /etc/dumpdates ファイルに記録する	-u オプション	<code>ufsdump 9ucf /dev/rmt/0 /export/home</code>	627ページの「例 — 増分バックアップ、ルート (/)」
カートリッジテープを指定する	-c オプション	<code>ufsdump 9ucf /dev/rmt/0 /export/home</code>	627ページの「例 — 増分バックアップ、ルート (/)」
テープドライブを指定する	-f <i>dump-file</i>	<code>ufsdump 9ucf /dev/rmt/0 /export/home</code>	627ページの「例 — 増分バックアップ、ルート (/)」
ローカルファイルシステムをリモートホストのテープデバイスにバックアップする	<i>remote-system:dump-file</i>	<code>ufsdump 0ucf pluto:/dev/rmt/0 /export/home</code>	629ページの「例 — リモートシステムへの完全バックアップ (Solaris 8 のデータを Solaris 8 システムへ)」

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と交換します。
9. 各テープにボリューム番号、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. `ufsrestore` コマンドでテープの内容を表示して、バックアップが正常に実行されたことを確認します。
このコマンドについては、第 44 章で説明します。

例 — 完全バックアップ、ルート (/)

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 QIC-150 テープ (/dev/rmt/0) 上に作成します。

```
# shutdown -g30 -y
# ufsdump 0ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Jul 13 10:46:09 1999
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 71058 blocks (34.70MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 70936 blocks (34.64MB) on 1 volume at 64 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue Jul 13 10:46:09 1999
# 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/dhccpagent
11432   ./etc/default/fs
11433   ./etc/default/inetinit
11434   ./etc/default/kbd
11435   ./etc/default/nfslogd
11436   ./etc/default/passwd
      .
      .
      .
# (Press Control-d to bring system to run level 3)
```

例 — 増分バックアップ、ルート (/)

この例では、ルート (/) ファイルシステムの増分バックアップを 4 mm DAT テープ (/dev/rmt/0) に作成します。

```

# ufsdump 9ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 9 dump: Tue Jul 13 10:58:12 1999
DUMP: Date of last level 0 dump: Tue Jul 13 10:46:09 1999
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 Tue Jul 13 10:58:12 1999
# 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 — 完全バックアップ、個々のホームディレクトリ

この例では、/export/home/kryten ディレクトリの完全バックアップを 4 mm DAT テープに作成します。

```

# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Jul 13 11:30:45 1999
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]

```

(続く)

```

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 8 のデータを Solaris 8 システムへ)

次の例では、Solaris 8 システム上のローカルの /export/home ファイルシステムを、リモートの Solaris 8 システム starbug 上のテープデバイスに完全バックアップします。

```

# ufsdump 0ucf starbug:/dev/rmt/0 /export/home
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Jul 13 13:14:40 1999
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t3d0s7 (mars:/export/home) to starbug:/dev/rmt/0
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 476 blocks (238KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 376 blocks (188KB) on 1 volume at 21 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue Jul 13 13:14:40 1999
# ufsrestore tf starbug:/dev/rmt/0
  2      .
  3      ./lost+found
3776    ./kryten
3777    ./kryten/.cshrc

```

(続く)

```

3778      ./kryten/.login
3779      ./kryten/b
3780      ./kryten/memos
7552      ./kryten/letters
7553      ./kryten/letters/b
7554      ./kryten/letters/letter1
7555      ./kryten/letters/letter2
7556      ./kryten/letters/letter3
11328     ./kryten/reports
11329     ./kryten/reports/reportA
11330     ./kryten/reports/reportB
11331     ./kryten/reports/reportC
          .
          .
          .
#

```

例 — リモートシステムへの完全バックアップ (Solaris 8 のデータを Sun 4.1.4 システムへ)

次の例では、Solaris 8 システム上のローカルの /export/home ファイルシステムを、リモートの SunOS 4.1.4 システム mars 上のテープデバイスに完全バックアップします。

注 - ufsdump コマンドで SunOS 4.0/4.1 用のデバイス名 (/dev/rst0) が使用されているので注意してください。

```

# ufsdump 0ucf mars:/dev/rst0 /export/home
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Thu Jul 15 09:13:01 1999
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s7 (starbug:/export/home) to mars:/dev/
rst0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 5690 blocks (2.78MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 5542 blocks (2.71MB) on 1 volume at 77 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Thu Jul 15 09:13:01 1999
# ufsrestore tf mars:/dev/rst0
      2      .
      3      ./lost+found

```

(続く)

```

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 (SunOS 4.1.4 のデータを Solaris 8 システムへ)

次の例では、Sun 4.1.4 システム mars 上のローカルのルート (/) ファイルシステムを、Solaris 8 システム starbug 上のリモートテープデバイスに完全バックアップします。

注 - SunOS 4.1.4 およびその互換バージョンのシステム上のデータをバックアップするときには、`ufsdump` コマンドではなく `dump` コマンドを使用しなければなりません。

```

# dump 0ucf starbug:/dev/rmt/0 /
DUMP: Date of this level 0 dump: Wed Jul  7 06:19:33 1999
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rsd0a (/) to /dev/rmt/0 on host starbug
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 123706 blocks (60.40MB) on 1.41 tape(s).
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: level 0 dump on Wed Jul  7 06:19:33 1999
DUMP: Tape rewinding
DUMP: 123680 blocks (60.39MB) on 2 volumes
DUMP: DUMP IS DONE
# restore tf starbug:/dev/rmt/0
   2      .
   3      ./lost+found
 3776    ./export
 7552    ./home

```

(続く)

```
11328      ./usr
15104      ./pcfs
3777       ./tftpboot
3778       ./tftpboot/tftpboot
3794       ./tftpboot/boot.sun4c.sunos.4.1.4
7553       ./etc
7554       ./etc/sendmail.cf
7555       ./etc/aliases
7556       ./etc/aliases.dir
7557       ./etc/aliases.pag
7558       ./etc/holidays
7559       ./etc/dumpdates
          .
          .
          .
```

#

ファイルとファイルシステムの復元 (手順)

この章では、ファイルシステムを復元する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「使用するテープを決定する方法」
- 637ページの「対話式でファイルを復元する方法」
- 640ページの「対話式でない方法で特定のファイルを復元する方法」
- 642ページの「リモートドライブを使ってファイルを復元する方法」
- 643ページの「ファイルシステム全体を復元する方法」
- 646ページの「ルート (/) と /usr を復元する方法」

この章では、`ufsdump` コマンドによってバックアップしたファイルやファイルシステムを復元する場合の、`ufsrestore(1M)` コマンドの使用方法を説明します。ファイルやファイルシステムを、アーカイブ、復元、コピー、または移動できる他のコマンドについては、第 46 章を参照してください。

ファイルとファイルシステムを復元するための準備

`ufsrestore` コマンドを使うと、`ufsdump` コマンドで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対する相対パスで指定されるディスク上の位置にファイルがコピーされます。`ufsrestore` を使うと、ファイルシステムの階層構造全体をレベル 0 ダンプとその後に行われた増分ダンプから再ロードしたり、個々のファイ

ルを任意のダンプテープから復元できます。ufsrestore 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。

ファイルやファイルシステムの復元を始める前に、次のことがらをあらかじめ確認しておく必要があります。

- 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したい raw デバイス名
- 使用するテープドライブのタイプ
- テープドライブのデバイス名 (ローカルまたはリモート)

ディスクデバイス名の決定

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ディスクデバイス名 (`/dev/rdsk/devicename`) が使えるはずです。詳細は、622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。

必要なテープドライブのタイプの決定

ファイルを復元するには、バックアップ媒体と互換性のあるテープドライブを使用しなければなりません。バックアップ媒体の形式によって、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媒体が 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。

テープドライブ名の決定

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。媒体のデバイスとデバイス名についての詳細は、第 47 章を参照してください。

ファイルシステム全体の復元

ファイルシステムは全面的に復元しなければならないほど破壊される場合があります。一般的な例として、ディスクヘッドが破損した場合は、ファイルシステムは全面的に復元しなければなりません。ただし、ソフトウェアを復元する前に、ハードウェアを取り替える必要があります。ディスクの取り替え方については、第 30 章または第 31 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、長時間かかります。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。

個々のファイルとディレクトリの復元

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` 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です) 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、`docs` 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。

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。

ファイルシステムの復元

次のことを知っておく必要があります。

- 復元したいファイルのテープ
- 復元したいファイルのパス名

▼ 使用するテープを決定する方法

1. 復元したい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. 自分が作成したバックアップ計画を参照して、そのファイルまたはファイルシステムが入っていた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 (0) から最低レベル (9) 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. オンラインのアーカイブファイルがある場合には、`ufsrestore` コマンドを使って正しい媒体を指定します。

```
# ufsrestore ta archive-name ./path/filename ./path/filename
```

<code>t</code>	テープ上に現れる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。

4. バックアップが入っている媒体をドライブに挿入し、`ufsrestore` コマンドを使って正しい媒体であるかどうかを確認します。

```
# ufsrestore tf device-name ./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。入っていなければ、そのファイルは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せん。

5. 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、`s /dev/rmt/n` オプションを使って、自分が使用したいダンプがあるテープの位置に移動してください。

```
# ufsrestore tfs /dev/rmt/n tape_number
```

例 — 使用するテープを決定する

`ufsdump` を使って `/usr` スライスをダンプする場合、内容一覧には `/usr` の下にあるファイルとディレクトリが列挙されます。`/usr/bin/pwd` がオンラインアーカイブに入っているかどうかを調べるには、次のように入力します。

```
# ufsrestore ta archive-name ./bin/pwd
```

`/usr/bin/pwd` がバックアップテープに入っているかどうかを調べるには、次のように入力します。

```
# ufsrestore tf /dev/rmt/n ./bin/pwd
```

▼ 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ります。
2. テープを書き込み保護の状態にします。
3. 最初のボリューム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
```

他のユーザーとの衝突を防ぐために、`/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は、復元後のファイルを置くファイルシステムと同じファイルシステム上の一時ディレクトリに、ファイルを復元します。復元後、`mv` コマンドで、階層全体を目的の場所に移動します。

5. `ufsrestore` コマンドを使って対話式で復元を開始します。

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。

```
# ufsrestore if /dev/rmt/n
```

6. 復元したい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. 復元したい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. ディレクトリや、復元したいファイルのリストからファイル名を削除しなければならない場合は、`delete` コマンドを使います。

```
ufsrestore> delete filename
```

7. `verbose` モードをオンにして、復元するファイル名を表示します。

```
ufsrestore> verbose
```

8. リストが完了したら、`extract` コマンドを使います。

```
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` が最後の整理が完了すると、`ufsrestore` 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. ファイルが復元されたことを確認します。
 - 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. リストで指定したファイルとディレクトリがすべて復元されているかどうかを確認するため、リストを調べます。
 - c. ファイルを正しい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、ファイル `/etc/passwd` と `/etc/shadow` がバックアップテープから復元されています。

```
# 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
#
```

▼ 対話式でない方法で特定のファイルを復元する方法

1. スーパーユーザーになります。
2. 安全のためにテープを書き込み保護の状態にします。
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
```


他のユーザーとの衝突を防ぐために、`/var/tmp` 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は、復元後のファイルを置くファイルシステムと同じファイルシステム上の一時ディレクトリに、ファイルを復元します。復元後、`mv` コマンドで、階層全体を目的の場所に移動します。

5. `ufsrestore` コマンドを使ってファイルを復元します。

```
# ufsrestore xvf /dev/rmt/n filename ...
```

x	引数 <i>filename</i> 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 <code>ufsrestore</code> コマンドに指定する。
v	復元する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。
<i>filename ...</i>	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。例： <code>./export/home/user1/mail</code> <code>./export/home/user2/mail</code> 。

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` 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。
- c. 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

この例では、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 Jul 14 08:42:42 1999
Dumped from: the epoch
Level 0 dump of a partial file system on starbug:/etc
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
Directories already exist, set modes anyway? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

▼ リモートドライブを使ってファイルを復元する方法

ufsrestore コマンドを使用するときにテープデバイス名の前に *remote-host:* を追加して、ファイルをリモートドライブから復元することができます。構文は次のようになります。

```
ufsrestore xf [user@]remote-host:/dev/rmt/n filename
```

例 — リモートドライブを使用してファイルを復元する

次の例では、システム `venus` 上にある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 filename
```

▼ ファイルシステム全体を復元する方法

注 - この手順は、ルート (`/`) または `/usr` の復元には使用できません。これらのファイルシステムを復元する方法については、646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ります。
2. 必要があれば、ファイルシステムをマウント解除します。

```
# umount /dev/rdisk/device-name
```

3. `newfs` コマンドを使って新しいファイルシステムを作成します。

```
# newfs /dev/rdisk/device-name
```

`raw` デバイス上で新しいファイルシステムを構築したいかどうかをたずね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、デバイス名が正しいことを確認します。

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/rdisk/cwtxdysz:(y/n)? y
```

新しいファイルシステムが作成されます。

5. 一時的なマウントポイントに新しいファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /mnt
```

6. ディレクトリを `/mnt` に変更します。

```
# cd mnt
```

マウントポイントディレクトリに変更しました。

7. テープを書き込み保護します。
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. `ufsrestore` コマンドを使ってテープ上の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、番号の順で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. テープを取り出して、ドライブに次のレベルのテープをロードします。
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. 最低レベルから最高レベルまで、ダンプのレベルごとに、644ページの手順 7 から 644ページの手順 10 までを繰り返します。
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
```

`ufsrestore` によって作成された `restoresymtable` が削除されます。

14. 別のディレクトリに変更します。

```
# 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
```

- 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。
- ufsdump コマンドを使って、新しく復元されたファイルシステムのレベル **0** のバックアップをとります。

```
# ufsdump 0uf /dev/rmt/n /dev/rdisk/device-name
```

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、新しく作成したファイルシステムについては、ただちにバックアップをとるようにしてください。(復元されたファイルシステムは、以前のバックアップ以降に変更されているように見えます)。

- 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。

- 復元され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

この例では、/export/home ファイルシステムが復元されます。

```
# 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: Wed Jul 14 08:49:33 1999
Dumped from: the epoch
```

(続く)

```
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. 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。
システムディスクを追加する方法についての詳細は、第 30 章または第 31 章を参照してください。
2. 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
```

3. /mnt ディレクトリに変更します。

```
# cd /mnt
```

4. テープを書き込み保護します。
5. ufsrestore コマンドを使って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。

6. 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、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。
7. ufsrestore コマンドは必要なだけ続けて使用します。

```
# ufsrestore rvf /dev/rmt/n
```

次のレベルのテープが復元されます。

8. テープを追加するたびに、手順 6 と手順 7 を繰り返します。
9. 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
```

10. restoresymtable ファイルを削除します。

```
# rm restoresymtable
```

ufsrestore によって作成され、復元のチェックポイントに使用された restoresymtable ファイルを削除します。

11. ルートディレクトリに変更します。

```
# cd /
```

12. 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
```

13. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

復元されたファイルシステムが完全であるかどうかチェックされます。

14. `installboot` コマンドで、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/rdisk/ devicename
```

SPARC システム上で `installboot` コマンドを使用する方法については 649 ページの「SPARC: 例 — ルート (/) ファイルシステムを復元する」を、IA システム上で `installboot` コマンドを使用する方法については 649 ページの「IA: 例 — ルート (/) ファイルシステムを復元する」をそれぞれ参照してください。

15. 新しいテープをテープドライブに挿入します。

16. 新しいファイルシステムのバックアップをとります。

```
# ufsdump 0uf /dev/rmt/n /dev/rdisk/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、必ずすぐにとるようにしてください。

17. 必要があれば、`/usr` ファイルシステムについて 647 ページの手順 5 から 648 ページの手順 18 を実行します。

18. 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

```
# 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/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

IA: 例 — ルート (/) ファイルシステムを復元する

```
# 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
/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```


ufsdump コマンドと ufsrestore 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。

この章の内容は以下のとおりです。

- 651ページの「ufsdump の機能」
- 657ページの「ufsdump コマンドのオプションと引数」
- 660ページの「ufsdump とセキュリティに関する注意事項」
- 661ページの「ufsrestore コマンドのオプションと引数」

ufsdump の機能

ufsdump コマンドは、ファイルシステムのバックアップを作成するときに 2 つのパスを作成します。最初のパスでは、ufsdump 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媒体に書き込みます。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媒体に書き込みます。

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、どのようにして媒体の終りを検出するかということです。

媒体の終りの検出

ufsdump は一連の固定長レコードを書き込みます。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、ufsdump は媒体の物理的な終りに達したものと想定します。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は、ufsdump が書き込もうとすると媒体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、媒体の終りが検出されます。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、媒体の終りは検出されません。

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` 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、バックアップ日時と曜日を表しています。次の例は、ファイルサーバーからの典型的な `/etc/dumpdates` ファイルを示しています。

```
/dev/rdsk/c0t0d0s0          9 Tue Jul 13 10:58:12 1999
```

(続く)

```

/dev/rdisk/c0t0d0s0          0 Tue Jul 13 10:46:09 1999
/dev/rdisk/c0t0d0s1          0 Tue Jul 13 13:41:04 1999

```

増分バックアップを実行するときに、`ufsdump` コマンドは `/etc/dumpdates` を検索して、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媒体にコピーします。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) がどれであるか分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、バックアップ先を指定します。次のいずれかを指定できます。

- 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* ではテープドライブかフロッピーディスクドライブの *raw* デバイスファイルを指定します。*ufsdump* が出力デバイスに書き込むときには、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、中密度のフォーマットを使用する 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、次のように入力します。

```
/dev/rmt/0mn
```

テープに複数のファイルを格納したい場合は、“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* 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についての詳細は、700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、リモートのテープドライブまたはフロッピーディスクドライブを指定します。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れば、*ufsdump* 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* を実行するのであれば、ルートシステム名がリモートシステムの */.rhosts* ファイルに記述されている必要があります。デバイスを *user@host:device* と指定した場合、*ufsdump* は指定されたユーザーとしてリモートシステム上でデバイスにアクセスしようとします。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 5.8 でのデバイス名 (`/dev/rmt/0`) を使用します。

注 - `dump-file` 引数を使用して、リモートデバイスを明示的に指定しなければなりません。SunOS の旧バージョンでは、`rdump` コマンドは出力を `dumphost` 別名で定義されたリモートデバイスに送っていました。ufsdump にはそれに相当する `rufsdump` がありません。

ufsdump 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 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/rdisk/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、必ずバックアップするファイル (*files-to-back-up*) を指定しなければなりません。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。通常はファイルシステムを識別しますが、個々のファイルやディレクトリを識別することもできます。

ファイルシステムの場合は、ディスクスライスの `raw` デバイスファイルを指定します。これには、ファイルコントローラの省略形 (`c`)、SCSI デバイス専用のターゲット番号 (`t`)、ディスク番号 (`d`) とスライス番号 (`s`) の識別番号が含まれます。たとえば、スタンドアロンシステム (またはサーバー) 上に SCSI ディスクコントローラ

があり、スライス 6 に入っている /usr のバックアップを作成したい場合は、デバイスを次のように指定します。

```
/dev/rdisk/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 (/home など) を使用して指定できます。

デバイスの命名規則についての詳細は、700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 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または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。

媒体の終りの検出

ufsdump は、ほとんどのデバイスの媒体の終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。

媒体の終りオプションを使用する必要があるのは、デバイスによる媒体の終りの検出方法が ufsdump で認識されないときや、restore コマンドの旧バージョンを使用してシステム上のファイルを復元しようとしているときです。restore コマンドの旧バージョンとの互換性を確保するために、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。

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

表 45-1 に、ufsdump コマンドでは実行できない作業を示します。

表 45-1 ufsdump で実行できない作業

実行できない作業	備考
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 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 コマンドの構文は次のとおりです。

```
/usr/sbin/ufsdump [options] [arguments] files-to-back-up
```

<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とは同じ順序で並べなければならない。
<i>files-to-back-up</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 files-to-back-up
```

`ufsdump` では、デフォルトで次のオプションが使用されます。

```
ufsdump 9uF /dev/rmt/0 files-to-back-up
```

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

表 45-2 に、`ufsdump` コマンドのオプションを示します。

表 45-2 `ufsdump` コマンドのオプション

オプション	説明
0-9	バックアップレベル。レベル 0 は <code>files-to-back-up</code> 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、 <code>ufsrestore</code> 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、もしあればどの媒体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。
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。媒体の終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、 <code>ufsdump</code> で媒体の終りを検出できないときのみ使用する必要がある。

表 45-2 ufsdump コマンドのオプション 続く

オプション	説明
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> 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。このコマンドをローカルシステム上で実行しリモートシステムにアクセスできるように、指定されたユーザーはリモートシステム上に <i>.rhosts</i> ファイルを持っていないなければならない。
l	自動ロード。このオプションは、オートロード (スタックロード) テープドライブがある場合に使用する。テープの終りに達すると、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 分経過しても準備ができなければ、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を表示する。
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)、可能であれば媒体をはずす (たとえば、フロッピーディスクを取り出したり、8mm の自動ロードテープをはずす)。
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、 <i>ufsdump</i> で媒体の終りを検出できないときのみ使用する必要がある。
S	バックアップの予想サイズ。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。このオプションを使用する必要があるのは、 <i>ufsdump</i> で媒体の終りを検出できないときだけである。

表 45-2 ufsdump コマンドのオプション 続く

オプション	説明
u	ダンプレコードのアップデート。ファイルシステムの完全バックアップをとる場合には、ファイル <code>/etc/dumpdates</code> ファイルにエントリを追加する。このエントリは、ファイルシステムのディスクスライスのデバイス名、バックアップレベル (0 ~ 9)、日付を示す。u オプションを使用しないと、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が書き込まれた後に、ソースファイルシステムと対照して媒体の内容を検査する。矛盾が発生すると、オペレータに新しい媒体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動作が発生すると矛盾が報告されるので、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専用である。
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入っていません。

ufsdump とセキュリティに関する注意事項

セキュリティに関する注意事項は次のとおりです。

- ufsdump コマンドを使用するには root アクセス権が必要である。
- 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しておかなければならない。

セキュリティについての詳細は、『Solaris のシステム管理 (第 2 巻)』の「システムセキュリティの管理」を参照してください。

ufsrestore コマンドのオプションと引数

ufsrestore コマンド構文

ufsrestore の構文は次のとおりです。

```
ufsrestore [options] [arguments] [filename ...]
```

<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。
<i>arguments</i>	オプションに対応する引数。オプションと同じ順序で並んでいなければならない。
<i>filename</i>	x または t オプションの引数として、復元するファイルを指定する。常に最後に指定する。

ufsrestore のオプションと引数

ufsrestore コマンドには、表 45-3 に示すオプションのうちの 1 つを指定する必要があります。

表 45-3 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 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媒体の内容を表示し、復元するファイルやディレクトリを個別に選択できる。利用できるコマンドのリストについては、664ページの「対話式の復元コマンド」を参照。
r	再帰。媒体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 (例えば <code>restoresymtable</code>) 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。これは新しいファイルシステム (<code>newfs</code> コマンドで作成したばかりのファイルシステム) 用だが、バックアップ媒体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 (<code>r</code> 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 <code>ufsrestore</code> コマンドを実行し直す。
x [<i>filename...</i>]	抽出。 <i>filename</i> 引数で指定したファイルを選択的に復元する。 <i>filename</i> では、ファイルとディレクトリのリストを指定できる。 <code>h</code> 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ファイルがすべて復元される。 <i>filename</i> を省略するか、ルートディレクトリを表す「 <code>.</code> 」を入力すると、媒体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>filename...</i>]	内容一覧。 <i>filename</i> 引数で指定したファイルが媒体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 <code>i</code> 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)。 <i>filename</i> 引数を入力しなければ、媒体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)。 <code>h</code> オプションも指定すると、内容ではなく <i>filename</i> 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、媒体の最初のボリュームから読み込まれるか、 <code>a</code> オプションを使用すると指定したアーカイブファイルから読み込まれる。このオプションは <code>x</code> 、 <code>r</code> オプションと併用して使用できない。

次の表に、ufsrestore コマンドの追加オプションを示します。

表 45-4 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>filename...</i>]	ダンプの内容一覧は、媒体 (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 <i>t</i> 、 <i>i</i> 、または <i>x</i> オプションと組み合わせて使用すると、媒体をマウントしなくてもダンプ内のファイルをチェックできる。 <i>x</i> コマンドと対話型抽出オプションといっしょに使用すると、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 処理ごとにテープに書き込む 512 バイトのブロック数。デフォルトでは、 <i>ufsrestore</i> はテープの書き込みに使用したブロック数を表示しようとする。
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 <i>/dev/rmt/0m</i> 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> の値を指定しなければならない。 <i>backup-file</i> が <i>system:device</i> 形式であれば、 <i>ufsrestore</i> 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>i</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 <i>/files</i> であれば、 <i>i</i> ノード番号が 42 のバックアップ <i>./dready/fcs/test</i> 内のファイルは、 <i>/files/42</i> として復元される。このオプションが便利なのは、少数のファイルを抽出するときだけである。
s <i>n</i>	最初のボリューム媒体上の <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。

表 45-4 ufsrestore コマンドの追加オプション 続く

オプション	説明
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	媒体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。

対話式の復元コマンド

表 45-5 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が表示される。ディレクトリは接尾辞 / 付きで表示され、現在のリスト内で復元 (抽出) されるエンタリは接頭辞 * 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切り替える。
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のリビ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ことがある。
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すべてのファイルが削除される。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。媒体またはファイルシステムからは削除されない。

表 45-5 対話式の復元コマンド 続く

オプション	説明
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。1つのボリュームのバックアップについてボリューム番号をたずね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ときに使用する。 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。
verbose	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 v と入力することもできる)。詳細モードがオンになっていると、対話型の ls コマンドでは i ノード番号が表示され、ufsrestore 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。
what	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。

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。

- 671ページの「ディスクをクローン化する方法 (dd)」
- 674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)」
- 678ページの「ファイルをテープにコピーする方法 (tar)」
- 679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)」
- 680ページの「テープからファイルを取り出す方法 (tar)」
- 683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)」
- 684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)」
- 685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)」
- 686ページの「テープから特定のファイルを取り出す方法 (cpio)」
- 688ページの「ファイルをリモートテープドライブにコピーする方法 (tar と dd)」
- 689ページの「ファイルをリモートテープドライブから取り出す方法」
- 691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)」

- 692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)」
- 693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)」
- 694ページの「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」
- 695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」
- 696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したいときは、第 45 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したいときは、`ufsdump` と `ufsrestore` の代わりに、この章で説明する手順を使用できます。

表 46-1 に、各種バックアップコマンドの用途を示します。

表 46-1 バックアップコマンドの用途

目的	使用するコマンド	参照ページ
ファイルシステムをテープにバックアップする	<code>ufsdump (1M)</code>	624ページの「テープにバックアップをとる方法」
ファイルシステムをテープから復元する	<code>ufsrestore (1M)</code>	643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 (1)</code> , <code>tar (1)</code> , <code>cpio (1)</code>	676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」

表 46-1 バックアップコマンドの用途 続く

目的	使用するコマンド	参照ページ
ファイルまたはファイルシステムをディスクにコピーする	dd(1M)	671ページの「ディスクをクローン化する方法 (dd)」
ファイルをフロッピーディスクにコピーする	tar(1)	691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)」

表 46-2 に、各種のバックアップコマンドと復元コマンドを示します。

表 46-2 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
volcopy	する	する	物理
tar	しない	しない	論理
cpio	しない	する	論理
pax	する	する	論理
dd	する	しない	物理
ufsdump/ufsrestore	する	する	論理

次の節では、各方法の長所と短所を説明し、コマンドの使用例を示します。

ファイルシステムをディスクにコピーする

ファイルシステムをディスクにコピーするには、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、読み込み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* は設定したいオプションで、*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
```

▼ ディスクをクローン化する方法 (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持っているかどうかを確認します。
2. スーパーユーザーになります。
3. /reconfigure ファイルをシステムに作成します。これによって、システムは追加されるクローンディスクをリブート時に認識します。

```
# 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. dd コマンドを使用して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/dsk/device-name of=/dev/dsk/device-name bs=blocksize
```

<code>if=/dev/dsk/device-name</code>	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。
<code>of=/dev/dsk/device-name</code>	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。
<code>bs=blocksize</code>	ブロックサイズ (128K バイト、256K バイトなど)。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。

8. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdsk/device-name
```

9. 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
```

10. 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` をすべて `c0t1d0` に変更します。

11. 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
```

12. システムを停止します。

```
# init 0
```

13. クローンディスクからシングルユーザーモードにブートします。

```
# boot disk1 -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。

14. 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。

15. 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
```

16. 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 — ディスクのクローン化 (dd)

```
# init 0
ok boot
# dd if=/dev/dsk/c0t0d0s2 of=/dev/dsk/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取り出して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媒体デバイスにコピーできます。

cpio コマンドは、媒体の終わりを認識し、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 (ufsdump よりも) 効率のよいコマンドです。

cpio の使用時には、しばしば ls や find などのコマンドを使用してコピーしたいファイルを選択し、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ります。
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て実行し、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 -pdmu /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。

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と、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
```

詳細は、cpio(1) のマニュアルページを参照してください。

ファイルとファイルシステムをテープにコピーする

pax、tar、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 46-3 cpio、pax、tar コマンドの長所と短所

コマンド	機能	長所	短所
pax	POSIX 準拠システムとの間でファイルをコピーする場合、または複数のテープボリュームが必要となる、ファイル、特殊ファイル、またはファイルシステムのコピー	<ul style="list-style-type: none"> ■ POSIX 準拠システムに対する可搬性は、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 コマンドの欠点を参照。ただし、pax は、複数のテープボリュームを作成できる。
tar	1 本のテープへの、ファイルやディレクトリのサブツリーのコピー	<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。
cpio	SunOS 5.8 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、または複数のテープボリュームが必要となる、ファイル、特殊ファイル、またはファイルシステムのコピー	<ul style="list-style-type: none"> ■ tar 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可搬性のために、異なるヘッダーフォーマット (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。 	

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と構成によって異なります。テープドライブとデバイス名についての詳細は、699ページの「使用する媒体の選択」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 コマンドでファイルをテープにコピーする前に、次のことを知っ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 (*) を使用して指定できる。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. コピーしたいファイルが入っているディレクトリに変更します。
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. tar コマンドを使用してファイルをテープにコピーします。

```
$ tar cvf /dev/rmt/n filename ...
```

<code>c</code>	アーカイブが作成されるように指定する。
<code>v</code>	アーカイブされるたびに、各ファイルの名前を表示する。
<code>f /dev/rmt/n</code>	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。
<code>filename ...</code>	コピーしたいファイルとディレクトリを指定する。

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. テープの内容を表示する `t` オプションを指定し、`tar` コマンドを使用して、コピーされたファイルがテープに入っているかどうかを確認します。**tar** テープ上のファイルを表示する方法についての詳細は、679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。

```
$ tar tvf /dev/rmt/n
```

例 — ファイルをテープにコピーする (tar)

次の例では、3つのファイルがテープドライブ0のテープにコピーされます。

```
$ 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. `tar` コマンドを使用してテープの内容を表示します。

```
$ tar tvf /dev/rmt/n [filename]
```

t	テープ上のファイルの内容一覧が表示される。
v	t オプションと併用すると、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。
filename ...	リスト表示したいファイルとディレクトリを指定する。

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 のテープに含まれてい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 1999 reports/
-rw-----t  0/1    30000 Jul 14 09:23 1999 reports/reportA
-rw-----t  0/1    31000 Jul 14 09:24 1999 reports/reportB
-rw-----t  0/1    32000 Jul 14 09:24 1999 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. tar コマンドを使用してテープからファイルを取り出します。

```
$ tar xvf /dev/rmt/n [filename ...]
```


x	指定したアーカイブファイルからファイルを抽出するように指定する。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。
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	アーカイブが入っているテープデバイスを示す。
filename	取り出すファイルを指定する。

4. 現在のディレクトリの内容をリストして、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。手順については、679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。

詳細は、tar(1) のマニュアルページを参照してください。

pax を使用してファイルをテープにコピーする

この節では、pax コマンドでファイルをコピーする方法について説明します。

▼ ファイルをテープにコピーする方法 (pax)

1. コピーしたいファイルが入ってい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. pax コマンドを使用してファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/0 filename ...
```

-w	書き込みモードを有効にする。
-f /dev/rmt/0	テープドライブを識別する。
filename ...	コピーしたいファイルとディレクトリを指定する。

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 -l -f /dev/rmt/0
```

5. ドライブからテープを取り出して、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)

```
$ pax -w -f /dev/rmt/0 .  
$ pax -f /dev/rmt/0  
filea  
fileb  
filec
```

詳細は、pax(1) のマニュアルページを参照してください。

▼ 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. 書き込み保護されていないテープをテープドライブに挿入します。
2. `ls` コマンドと `cpio` コマンドを使用して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 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。これによって他のベンダーのシステムとの可搬性を保つ。

`> /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。

3. 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/0
```

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。

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、ディレクトリ `/export/home/kryten` 内のすべてのファイルを、テープドライブ 0 のテープにコピーします。

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
92 blocks
$ cpio -civt < /dev/rmt/0
```

(続く)

```

-rw-----t 1 kryten users 400 Jul 14 09:28 1999, b
drwx--x--x 2 kryten users 0 Jul 14 09:26 1999, letters
-rw-----t 1 kryten users 10000 Jul 14 09:26 1999, letter1
-rw-----t 1 kryten users 10100 Jul 14 09:26 1999, letter2
-rw-----t 1 kryten users 11100 Jul 14 09:27 1999, letter3
-rw-----t 1 kryten users 12300 Jul 14 09:27 1999, letter4
drwx--x--x 2 kryten users 0 Jul 14 09:27 1999, memos
-rw-----t 1 kryten users 400 Jul 14 09:28 1999, memosmemoU
-rw-----t 1 kryten users 500 Jul 14 09:28 1999, memosmemoW
-rw-----t 1 kryten users 100 Jul 14 09:27 1999, memosmemoX
-rw-----t 1 kryten users 200 Jul 14 09:28 1999, memosmemoY
-rw-----t 1 kryten users 150 Jul 14 09:28 1999, memosmemoZ
drwx--x--x 2 kryten users 0 Jul 14 09:24 1999, 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読み取るので、そのために要するのと同じ時間がかかります。

1. テープをテープドライブに挿入します。
2. cpio コマンドを使用して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が表示される。
- < /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 のテープに含まれてい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 1999, answers
-rw-----t  1 kryten  users     800 Jul 14 09:36 1999, b
drwx--x--x  2 kryten  users      0 Jul 14 09:32 1999, sc.directives
-rw-----t  1 kryten  users    200000 Jul 14 09:35 1999, direct241
drwx--x--x  2 kryten  users      0 Jul 14 09:32 1999, tests
-rw-----t  1 kryten  users     800 Jul 14 09:36 1999, 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. ファイルを入れたいディレクトリに変更します。
2. テープをテープドライブに挿入します。
3. cpio コマンドを使用して、すべてのファイルをテープから現在のディレクトリにコピーします。

```
$ cpio -icvd < /dev/rmt/n
```

-i	テープの内容を読み込む。
-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/ <i>n</i>	入力ファイルを指定する。

- 現在のディレクトリの内容を表示して、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`)

- ファイルを置きたいディレクトリに移動します。
- テープをテープドライブに挿入します。
- `cpio` コマンドを使用して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	テープの内容を読み込む。
-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。

4. 現在のディレクトリの内容を表示して、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
```

詳細は、`cpio(1)` のマニュアルページを参照してください。

▼ ファイルをリモートテープドライブにコピーする方法 (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。
 - ローカルホスト名 (および、オプションで、コピーしているユーザーのユーザー名) が、リモートシステムの `/etc/hosts.equiv` ファイル内になければならない。あるいは、コピーしているユーザーは、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ておかなければならない。さらに、`$HOME/.rhosts` 内にローカルマシン名がなければならぬ。詳細は、`hosts.equiv(4)` を参照。
 - 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. リモートコマンドを実行するための適切なアクセス権を持っているかどうかをテストするには、次のように入力します。

```
$ rsh remotehost echo test
```

「test」と表示された場合、リモートコマンドを実行するためのアクセス権を持っています。「Permission denied」と表示された場合、上記の手順 1 の内容を確認してください。

3. `tar` コマンドと `dd` コマンドを使用して、ファイルをリモートのテープドライブにコピーします。

```
$ tar cf - files | rsh remotehost dd of=/dev/rmt/n obs=blocksize
```

<code>tar cf</code>	テープアーカイブを作成し、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>files</code>	コピーするファイル
<code> rsh remotehost</code>	<code>tar</code> コマンドの出力がパイプを通してリモートシェルに渡され、ファイルがコピーされる。

`dd of=/dev/rmt/n` 出力デバイスを指定する。

`obs=blocksize` ブロック係数を指定する。

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。

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ドライブから取り出す方法

1. 一時ディレクトリに変更します。

```
$ cd /var/tmp
```

2. tar コマンドと dd コマンドを使用して、ファイルをリモートのテープドライブに抽出します。

```
$ rsh remotehost dd if=/dev/rmt/n | tar xvBpf -
```

<code>rsh remotehost</code>	<code>dd</code> 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されるリモートシェル。
<code>dd if=/dev/rmt/n</code>	入力デバイスを指定する。
<code> tar xvBpf -</code>	<code>dd</code> コマンドの出力は <code>tar</code> コマンドにパイプされ、復元されたファイルに使用される。

3. ファイルが抽出されたかどうかを確認します。

```
$ 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

```
$ 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 /var/tmp
```

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しなければなりません。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 16 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 は媒体の終りを認識し、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。

両面高密度 3.5 インチフロッピーディスク (フロッピーディスクには「DS、HD」マークが付いています) を使用してください。

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ァイルをフォーマット済みフロッピーディスクにコピーすると、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。
- すでに tar イメージが入っているフロッピーディスクはマウントできない。

▼ 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. コピーしたいファイルが入ってい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. volcheck コマンドを使用してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

4. フロッピーディスク上のファイルシステムをすべてマウント解除し、再度フォーマットします。

```
$ fdformat -U /vol/dev/aliases/floppy0
```

5. tar コマンドを使用して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/rdiskette0/unlabeled 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上書きされます。

6. フロッピーディスクの内容を表示する `-t` オプションをつけて tar コマンドを使用し、コピーしたファイルがフロッピーディスクに入っているかどうかを検査します。ファイルのリストを表示する方法についての詳細は、692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。

```
$ tar tvf /vol/dev/rdiskette0/unlabeled
```

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。

```
$ cd /home/smith
$ ls evaluation*
evaluation.doc  evaluation.doc.backup
$ tar cvf /vol/dev/rdiskette0/unlabeled evaluation*
a evaluation.doc 86 blocks
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/rdiskette0/unlabeled
```

▼ 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. `volcheck` を実行してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

3. tar コマンドを使用して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/rdiskette0/unlabeled
```

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を表示します。

```
$ tar tvf /vol/dev/rdiskette0/unlabeled
rw-rw-rw-6693/10  44032 Oct 23 14:54 1996 evaluation.doc
rw-rw-rw-6693/10  43008 Oct 23 14:47 1996 evaluation.doc.backup
$
```

詳細は tar(1) のマニュアルページを参照してください。

複数のボリュームを扱いたい場合は、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。

▼ 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. volcheck を実行してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

4. tar コマンドを使用して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/rdiskette0/unlabeled
```

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。

- 現在のディレクトリの内容を表示して、ファイルが取り出されたことを確認します。

```
$ ls -l
```

- 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。

```
$ /home/smith/Evaluations
$ tar xvf /vol/dev/rdiskette0/unlabeled
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

tar コマンドを使用して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。

```
$ tar xvf /vol/dev/rdiskette0/unlabeled 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場合は、一杯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 の使用方法については、683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 5.8 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は次のとおりです。

```
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
```

-H オプションは、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、次のように cpio コマンドが失敗に終わります。

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
```

(続く)

```

$ 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、必ず媒体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いっしょ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見つけてください。

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 4.0/4.1 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 `-i` をいっしょに使用しなければなりません。`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. `volcheck` を実行して、フロッピーディスクを使用できるようにします。


```
$ volcheck
```

4. `cpio` コマンドを使用して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/rdiskette/unlabeled
```


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。

- 702ページの「テープドライブの状態を表示する方法」
- 704ページの「磁気テープカートリッジのたるみを直す方法」
- 704ページの「磁気テープカートリッジを巻き戻す方法」

使用する媒体の選択

通常は、次の媒体を使用して 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
- 8 mm のカートリッジテープ
- 4 mm 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。

どの媒体を選択するかは、媒体をサポートする機器とファイルの格納に使用する媒体 (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めます。

表 47-1 に、ファイルシステムの典型的なバックアップ媒体と、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 47-1 媒体の記憶容量

媒体	容量
1/2 インチ リールテープ	140 Mbytes (6250 bpi)
2.5G バイト 1/4 インチ カートリッジ (QIC) テープ	2.5G バイト
DDS3 4-mm カートリッジ テープ (DAT)	12 ~ 24G バイト
14-Gbyte 8-mm 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」のデバイスファイルが入っているサブディレクトリを指し、ドライブの論理デバイス番号が含まれています。テープドライブの命名規則では、物理デバイス名ではなく論理デバイス名を使用します。表 47-2 に、この命名方式を示します。

表 47-2 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、図 47-1 のようにテープドライブデバイスを指定します。

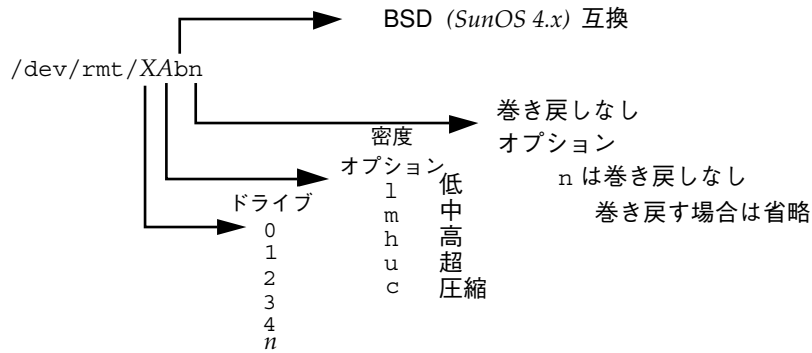


図 47-1 テープドライブデバイス名

密度を指定しないと、テープドライブは通常それがサポートする一番高い密度で書き込みます。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見てください。このディレクトリには、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ています。

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。

テープドライブのデフォルト密度を指定する

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。表 47-3 に、デフォルトの密度設定を使用してテープデバイス名を指定する方法を示します。

表 47-3 テープドライブのデフォルト密度指定

指定するドライブ	使用する番号
第 1 のドライブ、巻き戻し	/dev/rmt/0
第 1 のドライブ、巻き戻しなし	/dev/rmt/0n
第 2 のドライブ、巻き戻し	/dev/rmt/1m
第 2 のドライブ、巻き戻しなし	/dev/rmt/1n

デフォルトでは、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。

テープドライブに別の密度を指定する

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。表 47-4 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 47-4 テープドライブに別の密度を指定する

指定	番号
第 1 のドライブ、巻き戻し、低密度	<code>/dev/rmt/0l</code>
第 1 のドライブ、巻き戻しなし、低密度	<code>/dev/rmt/0ln</code>
第 2 のドライブ、巻き戻し、中密度	<code>/dev/rmt/1m</code>
第 2 のドライブ、巻き戻しなし、中密度	<code>/dev/rmt/1mn</code>

デバイス名および密度オプションについては、図 47-1 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. 情報を表示したい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) の状態を示しています。

```
$ 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。

```
$ mt -f /dev/rmt/n retension
```

例 — 磁気テープドライブのたるみを直す方法

次の例では、ドライブ /dev/rmt/1 内のテープのたるみを直します。

```
$ 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。

```
$ mt -f /dev/rmt/n rewind
```

例 — 磁気テープカートリッジを巻き戻す

次の例では、ドライブ /dev/rmt/1 内のテープを巻き戻します。

```
$ mt -f /dev/rmt/1 rewind
$
```


ドライブの管理と媒体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。そこで、テープドライブが正常に動作するように定期的に掃除してチェックするとよいでしょう。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2つの方法のうち、どちらかを使って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、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` オプションを使用すると、媒体の内容をソースファイルシステムと比較して検査できます。`-v` オプションを機能させるには、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しなければなりません。

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 42 章で説明したようなバックアップ方法を採用した場合は、ラベルに「テープA」、「テープB」というようにしておく必要があります。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) がどの媒体 (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

- 1.2M バイト フロッピーディスク 208, 209
- 1.44M バイトフロッピーディスク 185, 208, 209
- 2.88M バイト フロッピーディスク 185, 208
 - フォーマットオプション 208
- 4.3 Tahoe ファイルシステム 449
- 360K バイト フロッピーディスク 208, 209
- 720K バイトフロッピーディスク 208, 209
- 9660 CD フォーマット 193, 264

A

Admintool

ソフトウェア管理 271 - 273, 288

Admintool: ユーザー 269

アカウントを無効にする 50

「削除」オプション (編集メニュー) 50

初期設定ファイル 50

説明 49

長所 49

「追加」オプション (「編集」メニュー) 79

パスワード管理 50

「変更」オプション (編集メニュー) 49

aliases ディレクトリ 262

aliases ファイル 50

ARCH 環境変数 63

autofs 463

auto_home ファイル 50

awk コマンド 46

B

banner PROM コマンド 143

bar コマンド

作成したファイルを取り出す 696

basedir キーワード (管理ファイル) 274, 276

bin グループ 38

Bourne シェル

環境変数 62, 67

基本機能 60, 61

シェル (ローカル) 変数 62, 65

ユーザー初期設定ファイル 58 - 60, 69, 75

Break キー 155, 157

BSD Fat Fast ファイルシステム 449

C

CALENDAR 環境変数 63

CD

CD が使用中かどうかを調べる 195

ISO 9660 フォーマット 193, 264

UFS CD 193, 213, 264, 265

アクセス 189, 257, 259, 260, 262

アクセスするプロセスを終了させる 195

音楽 CD 201

コマンド 203

混合フォーマット 264, 265

情報のコピー 194, 195

取り出す 196

内容の検査 193

名前 192, 196

ボリューム管理下のディレクトリ 257, 259, 260

- マウント 188, 196, 197
- マウントポイント 196, 260, 262
- リモートアクセス 196 - 198
- ロード 193
- CD-ROM デバイス (命名) 342
- CD-ROM ドライブ
 - アクセスポイント 189
 - 新しいドライブ用の準備 202
 - 共有 198
 - ソフトウェアのインストール 280, 282
 - ボリューム管理下のディレクトリ 259, 260
- CD-ROM ドライブの共有 198
- CDPATH 環境変数 63
- /cdrom マウントポイント 260, 262
- cfgadm
 - PCI ホットプラグ機能 315
 - SCSI ホットプラグ機能 315
- cfsadmin コマンド 502, 513
- clri コマンド 453
- cpio コマンド 674, 683, 687
 - 指定したファイルをテープから取り出す 686
 - 使用する場合 678
 - ディレクトリ内のすべてのファイルをテープにコピー 683
 - テープからすべてのファイルを取り出す 685
 - テープ上のファイルのリスト 684
 - ファイルシステム間のディレクトリのコピー 674
 - 別のヘッダー形式でファイルをコピー 695
- cp コマンド
 - CD 情報のコピー 194, 195
 - フロッピーディスクからコピー 222
 - フロッピーディスクへコピー 224
- cred ファイル 50
- .cshrc ファイル
 - カスタマイズ 44, 60, 69
 - 説明 58
- C シェル
 - 環境変数 62, 67
 - 基本機能 60, 61
 - シェル (ローカル) 変数 62, 65
 - ユーザー初期設定ファイル 58, 60, 69, 75

D

- daemon グループ 38
- dd コマンド 670, 673
 - ディスクのクローン化 671
 - ファイルをリモートテープから取り出す (tar) 689
 - ファイルをリモートテープにコピー (tar) 688
- DD (中密度) フロッピーディスク
 - フォーマットオプション 208, 209
- DESKSET 環境変数 63
- /dev/dsk ディレクトリ 337
- /dev/rdisk ディレクトリ 337
- devfsadm 302
- dfstab ファイル
 - 共有 CD の構成 199
 - ユーザーのホームディレクトリ 81
- df コマンド 338, 453
- diskette ディレクトリ 258
- dmesg コマンド 309
 - IA の例 310
 - SPARC の例 309
- DOS
 - フロッピーディスクのフォーマット 207 - 209, 215, 217
- driver not attached メッセージ 306
- DVD-ROM 443

E

- eject コマンド
 - CD 196
 - フロッピーディスク 225
- env コマンド 62
- /etc/dfs/dfstab ファイル
 - 共有 CD の構成 199
 - ユーザーのホームディレクトリ 81
- /etc/dumpdates ファイル 652
- /etc/init.d ディレクトリ 124
- /etc/inittab ファイル 117, 120
 - エントリの説明 118
 - デフォルトの例 118
- /etc/passwd ファイル 51, 154, 169
 - 説明 51
 - フィールド 51, 52
 - ユーザー ID 番号の割り当て 39
 - ユーザーアカウントの削除 50

/etc/passwd ファイルの修復
IA 169
SPARC 154
/etc/rmmount.conf ファイル
CD の共有 199
音楽 CD の演奏 201
フロッピーディスクの共有 229, 230
/etc/shadow ファイル
説明 51
/etc/skel ディレクトリ 58
/etc/vfstab ファイル 82
/etc ファイル
ユーザーアカウント情報 37, 51
/export/home ディレクトリ 455
/export/home ファイルシステム 43
export コマンド 62

F

fdformat コマンド 236
DOS フロッピーディスクのフォーマット 215, 217
UFS フロッピーディスクのフォーマット 210
オプション 236
確認メッセージ 211
構文 210, 211, 215, 217
サイズと密度のオプション 208, 209, 215
正しいフォーマットの検査 210
便利なオプション 210, 215
FDFS ファイルシステム 452
ff コマンド 453
FIFOFS ファイルシステム 452
FIFO i ノード 557
/floppy マウントポイント 260, 262
format.dat ファイル
エントリの作成 380, 381
キーワード 431, 434
構文規則 430
内容 429
format ユーティリティ
analyze メニュー 426, 428
defect メニュー 428
fdisk メニュー 425
SCSI ディスクドライブの自動構成 381, 384
Solaris fdisk パーティションの作成 408, 412

概要 353
関連するマニュアルページ 437
機能と利点 353
コマンド名の入力方法 436
システム上のディスクの確認 365, 366
使用条件 421
使用上のガイドライン 355
使用する場合 354
情報の保存についての推奨事項 422
ディスクスライス情報の表示 370, 372
ディスクスライスとディスクラベルの作成 396, 416, 418
ディスクのフォーマット 368, 370
ディスクラベルの作成 373, 375
入力 435, 437
パーティションメニュー 424, 425
破損したディスクラベルの復元 377, 379
フォーマットされているかを調べる 368
ブロック番号の指定方法 435
ヘルプ機能の使用法 437
メインメニュー 422
fsck pass フィールド (vfstab) 563
fsck コマンド 338, 453
FSACTIVE 状態フラグ 553
FSBAD 状態フラグ 553
FSCLEAN 状態フラグ 553
FSSTABLE 状態フラグ 553
i ノードリストサイズのチェック 556
空き i ノードのチェック 556
空きブロックのチェック 556
構文とオプション 571, 574
修正する条件 554
修復 567
状態フラグ 552
スーパーブロックのチェック 555
対話式 565
fsdb コマンド 453
fstypes ファイル 466
fstyp コマンド 453
fs ファイル 466
fuser コマンド 249
CD が使用中かどうかを調べる 195
CD にアクセスするプロセスを終了させる 195
フロッピーディスクが使用中かどうかを調べる 224

G

GECOS フィールド (passwd ファイル) 52
GID 38
 大きな値 39
 グループファイル 56
 定義 45
 割り当て 45
grep コマンド 46, 466
groupadd コマンド 46
groupdel コマンド 46
groupmod コマンド 46
groups コマンド 45
group* コマンド 46
group ファイル
 説明 51
 フィールド 55
 ユーザーアカウントの削除 50

H

halt コマンド 131
HD (高密度) フロッピーディスク
 フォーマットオプション 208, 209
High Sierra ファイルシステム 449
\$HOME/\$ENV ファイル 58
HOME 環境変数 63
home シェル変数 62, 63
\$HOME ディレクトリ 44, 60
/home の自動マウント 464
/home ファイルシステム
 ユーザーのホームディレクトリ 43

I

IA システム
 UFS フォーマット 193, 264
 フロッピーディスクのフォーマット 207,
 209
ID 番号
 グループ 38, 45, 56
 ユーザー 38, 39, 49
init コマンド
 スタンドアロンシステムのシャットダウン 137
 説明 130
installboot コマンド 402, 420
iso_8895_1 67
ISO 標準

710 Solaris のシステム管理 (第 1 巻) ◆ 2001 年 5 月

9660 CD フォーマット 193
aab9660 CD フォーマット 264
i ノード 586
 FIFO 557
 キャラクタ型特殊ファイル 557
 形式とタイプのチェック 557
 サイズ 559
 シンボリックリンク 557
 通常 557
 ディレクトリ 557
 バイト数 593
 不良番号 560
 ブロック型特殊ファイル 557
 リンク数 558
i ノードの形式 557
i ノードの状態 557
i ノードのタイプ 557
i ノードのリンク数 558
i ノードリストサイズ 556

K

/kernel/drv ディレクトリ 305
Korn シェル
 環境変数 62, 67
 基本機能 60, 61
 シェル (ローカル) 変数 62, 65
 ユーザー初期設定ファイル 58, 60, 69, 75

L

L1-A キー 155, 157
labelit コマンド 453
LANG 環境変数 63, 66, 67
LC 環境変数 66, 67
LK パスワード 50, 54
local.cshrc ファイル 58
local.login ファイル 58
local.profile ファイル 59
locale 環境変数 63
.login ファイル
 カスタマイズ 44, 60, 69
 説明 58
LOGNAME 環境変数 63
loopback ファイルシステム
 作成 475
 マウント 485

lost+found ディレクトリ 552
LPDEST 環境変数 63
ls コマンド
 CD の内容の検査 193
 フロッピーディスクの内容の検査 221

M

MAIL 環境変数 62, 63
make コマンド 46
MANPATH 環境変数 63
MANSECT 環境変数 63
media was found メッセージ 220
mkfile コマンド 545, 547
mkfs コマンド 453, 470
mnttab ファイル 461
mount コマンド 338
mountall コマンド 453
mount コマンド 453
 リモート CD 197
 リモートのフロッピーディスク 227
mt コマンド 704

N

ncheck コマンド 453
nec オプション 209, 215
newfs コマンド 213, 338, 470, 594
newgrp コマンド 45, 56
NFS 66, 463
nfsd デーモン
 起動 199, 229
 実行中かの確認 198, 199
 実行中かの検査 228, 229
NFS サーバー 463
NFS ファイルシステム 485
NIS
 グループ 46
 ユーザーアカウント 37, 46, 51
NIS+ 269
 グループ 45, 46
 ユーザーアカウント 37, 46, 50, 51
nistbladm コマンド 46
nis* コマンド 46
noaccess グループ 38, 56
noask_pkgadd 管理ファイル 275, 281
nobody グループ 38, 56
no media was found メッセージ 220

NP パスワード 54

O

OPENWINHOME 環境変数 63
/opt ディレクトリ 456

P

passwd コマンド 46
passwd ファイル 51
 フィールド 51, 52
 ユーザー ID 番号の割り当て 39
 ユーザーアカウントの削除 50
patchadd コマンド 294, 297
patchrm コマンド 294, 298
PATH 環境変数
 設定 65, 66
 説明 64, 65
path シェル変数 62, 64
PC BIOS (とブート) 177
PCI ホットプラグ機能 301
PCMCIA メモリーカード
 DOS 用のフォーマット 241
 UFS ファイルシステム用のフォーマット 236
 識別 234
 情報のコピーまたは移動 247
 情報をカードへコピーまたは移動 248
 他のシステム上で使用できるようにする 253
 他のシステム上でのアクセス 251
 デフォルト名 235
 取り出す 250
 内容の表示 246
 マウント 252
 ロード 244
PC ファイルシステム 449
pkgadd コマンド
 -a オプション (管理ファイル) 275, 276, 279, 281
 -d オプション (デバイス名) 279 - 283
 -s オプション (スプールディレクトリ) 282, 283
 概要 271, 277
 使用の前提条件 273
 スタンドアロンシステム 279, 281, 287

スプールディレクトリ 282, 283
代替 ベースディレクトリ 276
ユーザーの対話操作の省略 275, 276
pkgchk コマンド
 オプション 284, 286
 概要 277, 284
 使用方法 284, 286
pkginfo コマンド
 -l オプション (詳細情報) 286
 インストール済みのすべてのパッケージ 284
 概要 273, 277, 284
 使用 282
 使用方法 284
pkgparam コマンド 277
pkgrm コマンド 287
 -s オプション (スプールされたパッケージ) 288
 rm コマンドとの比較 274, 287
 概要 271, 277
 基本手順 287
 使用の前提条件 273
 注意 274, 287
/pkg ディレクトリ 282, 283
PROCFS ファイルシステム 452
/proc ディレクトリ 452, 456
.profile ファイル
 カスタマイズ 44, 60, 69
 説明 58
PROM
 ok プロンプトへの切り替え 142
 ROM Rev 番号の確認 143
 ブート設定の変更 143
 モニター 175
 リリースレベルの確認 142
prompt シェル変数 64
PROM のリリースレベル 142
prtconf コマンド 307
prvtoc コマンド 338, 375
PS1 環境変数 64

R

raw ディスクデバイスインタフェース 337,
 338
rdiskette ディレクトリ 258
reboot コマンド 131

removef コマンド 274
reset コマンド 145
rmmount.conf ファイル
 CD の共有 199
 音楽 CD の演奏 201
 フロッピーディスクの共有 229, 230
rm コマンド 274, 287
Rock Ridge 拡張 (HSFS ファイルシステム) 449
root グループ 38

S

/sbin/rc0 スクリプト 125
/sbin/rc1 スクリプト 126
/sbin/rc2 スクリプト 127
/sbin/rc3 スクリプト 127
/sbin/rc5 スクリプト 128
/sbin/rc6 スクリプト 128
/sbin/rcS スクリプト 128
SCSI ホットプラグ機能 301
SCSI ディスクドライブ 381
SCSI テープドライブ 701
setenv コマンド 62
set コマンド 62
shadow ファイル
 説明 51
 フィールド 54, 55
shareall コマンド 463
share コマンド 463
 他のシステム上でディスクを使用できるようにする 199, 229
SHELL 環境変数 64
shutdown コマンド
 サーバーのシャットダウン 110, 133
 説明 130
 ユーザーへの通知 131
/skel ディレクトリ 58
Solaris fdisk パーティション 405
Solaris アップグレードオプション 274
Solaris のフロッピーディスクのフォーマット 207, 209
Solstice AdminSuite 46
sort コマンド 46
SPARC システム
 UFS フォーマット 193, 264

フロッピーディスクのフォーマット 207, 209
SPARC システムの再設定 145
staff グループ 45
start コマンド 203
Stop-A キー 155, 157
stop コマンド 203
stty コマンド 66
SunOS のデフォルトファイルシステム 455
SUNW接頭辞 273
Sun ソフトウェアパッケージ
インストール 280 - 283
swapadd コマンド 543
swap コマンド 545
swmtool コマンド 269, 272
sync コマンド 156, 157
sync コマンドによるディスクの同期 156, 157
sysdef コマンド 307

T

tar コマンド 678, 681
1 枚のフロッピーディスクにコピー 691
テープからファイルを取り出す 680
テープ上のファイルのリスト 679
ファイルをテープへコピー 678
ファイルをリモートテープから取り出す
(dd) 689
ファイルをリモートテープにコピー
(dd) 688
フロッピーディスクからファイルを取り
出す 693
フロッピーディスク上のファイルのリス
ト 693
TERMINFO 環境変数 64
TERM 環境変数 65
term シェル変数 62, 65
TMPFS ファイルシステム 450
/tmp ディレクトリ 451, 456
tty (疑似) 39
tty タイプの疑似ユーザーログイン 39
TZ 環境変数 65

U

UDF ファイルシステム 442
UFS CD
SPARC と IA フォーマット 193, 264

混合フォーマット 264, 265
ufsdump コマンド 623, 630
オプションと引数 657
機能 651
制限 657
データのコピー方法 652
テープへファイルをバックアップ 624
媒体の終わりの検出 652
ufsrestore コマンド 634, 649, 661
使用する準備 634
テープからのファイルシステム全体の復
元 644
UFS ファイルシステム 449, 457
PCMCIA メモリーカード上に作成 238
拡張基礎タイプ 457
状態フラグ 457
大規模ファイルシステム 457
フロッピーディスクへ追加 213, 215
マウント 485
UFS フロッピーディスク
SPARC と IA フォーマット 207, 264
UFS ファイルシステムの追加 213, 215
フォーマット 207 - 211, 213
UID 49
大きな値 39
定義 38
割り当て 39
umask コマンド 67
umountall コマンド 454
umount コマンド 453
UNIX グループ 45, 269
UNIX ファイルシステム 449
useradd コマンド 46
userdel コマンド 46
usermod コマンド 46
user シェル変数 62
/usr ファイルシステム 455
uucp グループ 39

V

/var/run ファイルシステム 441
/var/sadm/install/admin ディレクトリ 275
/var/sadm/patch 297
/var/spool/pkg ディレクトリ 281 - 283
/var ディレクトリ 456
vfstab ファイル 466, 543

fsck pass の変更 563
一時ファイルシステム用エントリ 476
エントリの作成 484
すべてのファイルのマウント 486
スワップの追加 543
デフォルト 462
ファイルシステム名の検索 622
vipw コマンド 46
/vol/dev/dsk ディレクトリ 259
/vol/dev/rdisk ディレクトリ 259, 260
/vol/dev ディレクトリ 257, 258
 cdrom マウントポイント 260, 262
 CD サブディレクトリ 257, 259, 260
 floppy マウントポイント 260, 262
 シンボリックリンク 262
/vol/dsk ディレクトリ 260
volcopy コマンド 454

W

who コマンド 117, 132

Y

yp* コマンド 46

あ

アーカイブ (cpio) 695
空き i ノード 556
空きブロック 556, 588
空き容量 (最小) 591
アクセス
 CD アクセスポイント 189
 他のシステム上の PCMCIA メモリーカード 251
 ディスクデバイス 337, 341
 テープデバイス 341
 フロッピーディスクのアクセスポイント 189
アクセス権 67, 76, 269
 CD からファイルをコピー 194
アップグレードオプション (Solaris) 274
暗号化 51

い

一次グループ 45, 269
一時ファイルシステム 450, 473
移動
 フロッピーディスク情報 222 - 224

お

遅れ (回転) 591
オプション
 mkfile 546
 ufsdump コマンド 657
 ufsrestore コマンド 661
音楽 CD 201
音楽 CD の演奏 201

か

カートリッジテープ
 たるみを直す 704
回転の遅れ 591
書き込み権
 説明 194
書き込み保護 (PCMCIA メモリーカード) 236
拡張基礎タイプ (UFS ファイルシステム) 457
拡張密度フロッピーディスク
 フォーマットオプション 208
確認
 PROM リリースレベル 142
 nfsd デーモンが実行中かの 199
 システム上のディスク 365
 システムにログインしているユーザー 132
 ファイルシステムのタイプ 466
カスタマイズ 269
仮想ファイルシステム 450
仮想ファイルシステムテーブル 462
仮想メモリー記憶域 456, 540
環境
 シェル 61, 65
環境変数 269
 説明 61, 67
間接ブロック 559
完全バックアップ
 ufsdump コマンドによる 624
 定義 608
 必要なテープの本数 622

管理コマンド 453
管理ファイル 275, 276, 279

き

記憶域 (仮想メモリー) 456, 540
記憶ブロック 588
記憶容量 (媒体) 603, 700
疑似 tty 39
疑似ユーザーログイン 39
起動
 nfsd デーモン 199, 229
 ボリューム管理 203
キャッシュされたファイルシステム 500
 削除 513
 作成 502
 情報の表示 511
 設定 502
 チェック (fsck) 514
 パラメータ 536
 パラメータの設定 505
キャラクタ型特殊 i ノード 557
共有
 クライアントとサーバーによるソフトウェアの共有 273
 ファイル 463
 ユーザーのホームディレクトリ 80, 82
記録
 増分バックアップ 653
 ダンプ 652

く

空白 (ユーザーログイン名における) 37
クライアント
 ソフトウェア管理 273, 274, 284, 286
グループ
 ID 番号 38, 45, 56
 UNIX 45
 アクセス権の設定 68
 一次 45
 一次グループの変更 45
 管理ツール 46
 管理のガイドライン 45
 コマンド 46
 削除 46
 情報の格納 51, 55
 説明 36, 45

追加 46, 77
デフォルト 45
名前 45, 85
名前の変更 85
二次 45, 56
ネームサービス 45, 46
パスワード 55
ユーザーが所属するグループの表示 45
ユーザーの変更 46, 49
グループ ID 番号 38, 45, 56
グループファイル
 フィールド 56
グループ名
 説明 45
 変更 85

け

検査
 CD が使用中かどうか 195, 196
 nfsd デーモンが実行中かについて 198
 nfsd デーモンの実行 228, 229
 ソフトウェアパッケージのインストール 277, 282, 284, 286, 287
 フロッピーディスクが使用中か 224
検索
 ユーザーアカウント 46
検出、媒体の終わり
 cpio コマンド 674
 ufsdump コマンド 652, 656

こ

構造、シリンダグループの 585
構文
 fsck コマンド 571, 573
 newfs 594
高密度 (HD) フロッピーディスク
 フォーマットオプション 208, 209
互換性のあるアーカイブ 695
コピー
 CD 情報 194, 195
 PCMCIA メモリーカードから情報をコピー 247
 PCMCIA メモリーカードへ情報をコピー 248
 個別のファイル (cpio) 674

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) 683
データ (ufsdump) 652
テープからファイルをコピー (pax) 682
ファイルシステム間でディレクトリをコ
ピー (cpio) 674
ファイルシステム全体 (dd) 670
ファイルのグループ (cpio) 674
ファイルをテープへコピー (tar) 678
ファイルをフロッピーディスクにコ
ピー 690
ファイルをリモートテープにコピー (tar
と dd) 688
フロッピーディスク情報 222 - 224
フロッピーディスクにファイルをコピー
(tar) 691
別のヘッダー形式でファイルをコピー
(cpio) 695
ユーザーアカウント 46

さ

サーバー 269
説明 103
ソフトウェア管理 273, 287
サーバーとクライアントのサポート
説明 102
サービスの起動と停止 122
再構成ブート 382
IA の例 407
SPARC の例 394
最小空き容量 591
最小文字数
ユーザーパスワードの長さ 42
ユーザーログイン名の長さ 37
サイズ
i ノード 559
ファイルシステムのチェック 556
フラグメント 590
フロッピーディスクのフォーマット 208
- 211, 215
最大数
ユーザーが所属できる二次グループ数 45
最大値
ユーザー ID 番号 38
最大文字数
ユーザーパスワードの長さ 42
ユーザーログイン名の長さ 37

最適化のタイプ 592
サイトの初期設定ファイル 59
削除
キャッシュされたファイルシステム 513
グループ 46
メール別名 50
ユーザーアカウント 46, 50
ユーザーのホームディレクトリ 50
ユーザーのメールボックス 50
「削除」オプション (編集メニュー)
Admintool: ユーザー 50
作成 269
Solaris fdisk パーティション 405, 408
UFS ファイルシステム 471
format.dat エントリ 380, 381
loopback ファイルシステム 475
一時ファイルシステム 473
互換性のあるアーカイブ (cpio) 695
スワップファイル 545
ディスクスライスとディスクラベル 396,
416
ファイルシステム 470
マウントポイント 460
サポートされていないデバイス 305

し

シェル
環境 61, 65
環境変数 61, 62, 67
基本機能 60, 61
ユーザー初期設定ファイル 58, 60, 69, 75
ローカル変数 62, 65
シェル変数 62, 65
時間 (最適化のタイプ) 592
時間帯の環境変数 65
磁気テープカートリッジ
たるみを直す 704
巻き戻し 704
識別
PCMCIA メモリーカード 234
デバイス 307
システムアーキテクチャ 63
システムアカウント 38
システムシャットダウンコマンド 130
システム初期設定ファイル 44
システムタイプ

- 概要 102
- サーバー 103
- スタンドアロンシステム 104
- システムディスク
 - 接続 393, 406
 - 説明 352
 - ブートブロックのインストール 401, 419
- システムのシャットダウン 110
- システムのブート(ガイドライン) 110
- 実行制御スクリプト 121
 - サービスの起動と停止 122
 - 追加 123
 - 無効にする 124
- 実行レベル
 - 0 (電源切断状態) 116
 - 1 (シングルユーザー状態) 116
 - 2 (マルチユーザー状態) 116
 - 3 (NFS を使用できるマルチユーザー状態) 116, 119, 120, 147, 162
 - 6 (リブート状態) 116
 - s または S (シングルユーザー状態) 116, 148, 163
- 調べる 117
- 定義 115
- デフォルトの実行レベル 115
- 自動構成プロセス 305
- 自動ブートチェック機能の変更 562
- 自動ブートチェックの変更 564
- 自動マウント
 - ユーザーのホームディレクトリ 44
- シャットダウン
 - shutdown と init コマンドによる正常なシャットダウン 130
 - サーバー 132
 - スタンドアロンシステム 136
- シャットダウン時間のユーザーへの通知 132
- 重複ブロック 559
- 終了
 - CD にアクセスするプロセス 195
 - フロッピーディスクにアクセスするプロセス 224
- ボリューム管理 203
- 準備
 - バックアップ 621
 - ファイルの復元 633
- 状態フラグ
 - UFS ファイルシステム 457
 - fsck 552

- 初期設定ファイル
 - システム 44
- 調べる
 - 完全バックアップに必要なテープの本数 622
 - ディスクデバイス名 634
 - テープデバイス名 634
 - テープドライブのタイプ 634, 702
 - ファイルシステムのタイプ 466
- シリンダグループ 585
- シンボリックリンク 557
 - CD ディレクトリの表示 193
 - raw デバイスのアクセス 262
 - ファイルシステムのアクセス 262
 - フロッピーディスクのディレクトリの表示 221

す

- スーパーブロック 555, 568, 585
- スケルトンディレクトリ 44, 50
- スタンドアロンシステム 104
 - ソフトウェアパッケージの削除 287
 - ソフトウェアパッケージの追加 279, 281
- 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 282, 283, 286
 - ソフトウェアパッケージの削除 287
- すべてのデバイスの電源を落とす方法 138
- すべてのプロセスの停止 495
- スライス(定義) 347
- スワップパーティション 456, 540
- スワップファイル
 - vfstab へ追加 543
 - 削除 548
 - 作成 545
 - 表示 544
- スワップファイルの削除 548

せ

- 正常なシャットダウン 130
- セキュリティ 269
 - ユーザー ID 番号の再利用 39

そ

- 増分バックアップ 608, 653

その他 (アクセス権の設定) 68
ソフトウェア管理 269
Solaris アップグレードオプション 274
インストール済みソフトウェア情報の表示 273, 277, 284, 286
インストールの検査 277, 282, 284, 286
概要 269
クライアント 273, 274
サーバー 273, 287
ツール 269, 271, 272, 277
定義 271
パッケージの削除 271 - 274, 276, 287, 288
パッケージの追加 269, 271 - 276, 277, 279 - 283, 286
パッケージの定義 271
パッケージの命名規則 273
ソフトウェアパッケージの複数のバージョン 274, 276
ソフトウェアマネージャ 269, 272

た

大規模ファイルシステムオプション 480
対話式
ファイルシステムのチェック 565, 566
復元 637
対話式ブート
IA 164
SPARC 149
ダンプレベル
定義 610
日単位個別バックアップ 610
日単位累積バックアップ 610

ち

チェック
i ノードの形式とタイプ 557
i ノードリストの整合性 556
空き i ノード 556
空きブロック 556
自動ブートの変更 562
ファイルシステム 565
ファイルシステムのサイズ 556
ファイルシステムの対話式のチェック 566
中密度 (DD) フロッピーディスク
フォーマットオプション 208, 209

直接アクセスするディスクコントローラ 339
直接入出力 597

つ

追加

周辺デバイス 312
グループ 46, 77
サーバーとクライアントのサポート 102
実行制御スクリプト 123
スワップを `vfstab` へ追加 543
ディスク 405
ユーザーアカウント 46, 79
ユーザー初期設定ファイル 50
ユーザーをグループに 46, 49, 85
デバイスドライバ 314
「追加」オプション (「編集」メニュー)
Admintool: グループ 77
Admintool: ユーザー 79
通常の i ノード 557

て

提供側スライスの説明 361

停止

CD にアクセスするプロセスを終了させる 195
CD の取り出し 196
ファイルシステムのすべてのプロセス 495
フロッピーディスクにアクセスするプロセスの終了 224

ディスク

SCSI ドライブの自動構成 381
新しいディスク上でのファイルシステムの作成 400, 418
欠陥セクターの修復 384, 387
システム上のディスクの確認 365
システムディスクの接続 393, 406
追加 405
ディスクスライスとディスクラベルの作成 396, 416
ディスクラベルの検査 375
二次ディスクの接続 394, 407
破損したディスクラベルの復元 376
フォーマット 356
フォーマットされているかを調べる 368

- フォーマットする場合 367
- ラベル 373
- ディスクコントローラ 338
- ディスクスライス
 - システム構成の要件 352
 - 使用するスライスの決定 352
 - 情報の表示 370
 - 定義 347
- ディスクスライスの指定 338, 341
- ディスクデバイス名 634
- ディスクのクローン化 671
- ディスクのフォーマットの概要 356
- ディスクベースのファイルシステム 449
- ディスクラベル
 - prtvtoc コマンドによる検査 375
 - 作成 373
 - 説明 357
- 低密度フロッピーディスク
 - フォーマットオプション 208, 209
- ディレクトリ
 - CD 257, 259, 260, 262
 - PATH 環境変数 64 - 66
 - i ノード 557
 - path シェル変数 62, 64
 - /proc 452
 - /tmp 451
 - アクセス制御 36, 67
 - コピー 194, 195, 222 - 224
 - スケルトン 44, 50
 - ファイルシステム間のコピー (cpio) 674
 - フロッピーディスク 257, 260, 262
 - ベースディレクトリ (basedir) 274, 276
 - ホーム 43
 - 未割り当てブロック 560
 - ユーザーアカウント用の所有権の変更 49
- データディレクトリブロック 560
- データブロック 561
- テープ 704
 - tar コマンドによるファイルのリスト 679
 - 記憶容量 603, 700
 - サイズ 603, 700
 - 指定したファイルを取り出す (cpio) 686
 - 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) 683
 - 特性 656
 - ファイルを取り出す (cpio) 685
 - ファイルを取り出す (tar) 680
- 容量 656
- テープ (磁気カートリッジ)
 - たるみを直す 704
- テープデバイス (命名) 341
- テープドライブ
 - 管理 705
 - 最大の SCSI テープドライブ数 701
 - タイプを調べる 702
 - デフォルトの密度 701
 - 復元用にタイプを調べる 634
 - リモートからの復元 642
- テープドライブの管理 705
- テープドライブのタイプ 702
- テープの容量 656
- テーブル
 - マウント 461
- 手作業によるマウント
 - 自動マウントとの比較 188
 - リモート CD 196, 197
 - リモートのフロッピーディスク 226, 227
- デバイス
 - アクセス 335
 - 電源を落とす場合 131
- デバイスドライバ
 - 追加 314
 - 定義 304
- デバイスのインスタンス名 336
- デバイスの電源を落とす場合 131
- デバイス名
 - ディスクデバイス名を調べる 634
 - テープデバイス名を調べる 634
 - バックアップ 700
- デフォルト
 - LOGNAME 環境変数 63
 - PATH 環境変数 64, 66
 - SHELL 環境変数 64
 - SunOS ファイルシステム 455
 - TZ (時間帯) 環境変数 65
 - /tmp (TMPFS) 用ファイルシステム 451
 - グループ 45
 - スプールディレクトリ 281
 - テープドライブの密度 701
 - ファイルのアクセス権 67
 - プリンタ 63
 - マウントオプション 488, 492
 - ユーザーアカウントのデフォルト設定 46
 - ユーザー初期設定 ファイル 58

と

動的再構成 315

取り消し

UFS フロッピーディスクのフォーマット 211

取り出し

bar コマンドで作成したファイルを取り出す 696

指定したファイルをテープから取り出す (cpio) 686

テープからすべてのファイルを取り出す (cpio) 685

テープからファイルを取り出す (tar) 680

ファイルをフロッピーディスクから取り出す (tar) 693

ファイルをリモートテープから取り出す (tar と dd) 689

取り出す

CD 196

PCMCIA メモリーカード 250

フロッピーディスク 225

な

名前

SUNW 接頭辞 273

グループ 45

ソフトウェアパッケージの命名規則 273

ユーザーログイン 36, 37, 49

名前/命名

CD 名 192, 197

フロッピーディスク名 206

に

二次グループ 45, 56, 269

二次ディスク

システムへの接続 395, 408

説明 352

ね

ネームサービス 269

グループ 45, 46

ユーザーアカウント 37, 46, 51

ネットワーク 269

ネットワークベースのファイルシステム 450

は

パーティション (スワップ) 456, 540

媒体の終わりの検出

cpio コマンド 674

ufsdump コマンド 652, 656

バイト数 (i ノードごとの) 593

バス指向コントローラ 340

バス指向ディスクコントローラ 340

パス名 189

パスワード (グループ) 56

パスワードの有効期限 50, 51

パスワードファイル

テープからの復元 642

パスワード (ユーザー)

Admintool: ユーザー 50

LK パスワード 50, 54

NP パスワード 54

暗号化 51

期限切れ 55

設定 42, 50

説明 36, 42, 43

選択 42

注意点 42

変更 42, 46, 50, 54

有効期限 43, 50, 51

ユーザーアカウントを無効に、またはロックする 50, 55

バックアップ

完全バックアップと増分バックアップ、定義 608

準備 621

増分バックアップの記録 653

タイプ 608

テープへの完全 (レベル 0) バックアップ 624

デバイス名 700

ファイルシステムの選択 604

理由 603

バックアップスケジュール 609

ガイドライン 609

サーバー用 615

推奨事項 619

ダンプレベル 610

日単位増分、週単位累積バックアップ 614

日単位累積、週単位増分バックアップ 613
日単位累積、週単位累積バックアップ 611
例 611, 618
パッチ
Sun Service カスタマが利用できるもの 295
WWW によるアクセス 296
ftp によるアクセス 296
一般的な利用可能性 295
インストール 297
インストール済みのものを調べる 294
インストール用 README 294
削除 298
参照先 296
定義 293
パッチ番号 297
ユーティリティ 294
パラメータ (ファイルシステム) 589

ひ

日単位個別バックアップ 610
日単位累積バックアップ 610
ビッグエンディアン方式のビットコード化 205

表

mkfile のオプション 546
ufsdump コマンドのオプション 658
ufsrestore コマンドのオプション 661
ufsrestore の対話式コマンド 664
テープ容量 656
デフォルトの Solaris ファイルシステム 455
媒体の記憶容量 603, 700
ファイルシステム管理コマンド 453

表示

CD の内容 193
CD ユーザー 195
PCMCIA 内容 246
環境変数 61
システム構成情報 306, 308
スワップ空間 544
ディスクスライス情報 370
デバイス情報 309
フロッピーディスクの内容 221
ユーザーマスク 67

ふ

ファイル 269
/etc/default/fs 466
/etc/dfs/fstypes 466
/proc ディレクトリの中 452
アクセス制御 36, 67
数 593
共有 463
コピー (cpio) 674
新規インストールパッケージの属性の検査 285
対話式の復元 637
テープから取り出す (cpio) 685, 686
テープから取り出す (tar) 680
テープからファイルをコピー (pax) 682
テープ上のファイルのリスト (tar) 679
テープ上のファイルのリストを表示 (cpio) 684
テープへコピー (tar) 678
特定のファイルの復元 640
複数のフロッピーディスクにアーカイブする (cpio) 694
フロッピーディスクから取り出す (tar) 693
フロッピーディスク上のファイルのリスト (tar) 692
フロッピーディスクにコピー 690
フロッピーディスクにコピー (tar) 691
ユーザーアカウント用の所有権の変更 49
ファイルシステム
/ 455
4.3 Tahoe 449
BSD Fat Fast 449
DOS 449
/export/home 455
FDFS 452
FIFOFS 452
loopback の作成 475
NAMEFS 452
NFS のマウント 491
/opt 456
/proc 456
PROCFS 452
SPECFS 452
SunOS のデフォルト 455
SWAPFS 452

- TMPFS 450
- UFS 449
- UFS について 457
- UFS の作成 471
- UNIX 449
- /usr 455
- /var 456
- カスタムパラメータ 589
- 仮想 450
- 管理コマンド 453
- キャッシュされた 500
- 共有 463
- サイズのチェック 556
- 修正 570
- 修復 567, 568
- 使用可能にする 478
- シリンダグループの構造 585
- すべてのプロセスの停止 495
- 整合性がなくなる原因 555
- 全体のコピー (dd) 670
- 全体の復元 635, 643
- 大規模 480
- タイプ 448
- タイプを調べる 466
- 対話方式のチェック 565, 566
- チェック 565
- 定義 448
- ディスクベース 449
- テーブル (vfstab) 462
- どれにバックアップするか 604
- ネットワークベース 450
- 破損 552
- バックアップする理由 603
- プロセス 452
- マウント解除 496
- マウントテーブル 461
- マニュアルページ 454
- ファイルシステムテーブル
- 仮想 462
- ファイルシステムの修復 567, 568
- ファイルシステムのすべてのプロセスの停止 495
- ファイルシステムの整合性がなくなる 555
- ファイルシステムのタイプ 448
- ファイルシステムの破損 552
- ファイルシステムのバックアップと復元
- コマンド 602
- コマンドの概要 607
- 定義 602
- ファイルシステムの復元
- 全体 643, 645
- ルートと /usr 646, 649
- ファイルシステムのマウント解除 459, 496, 497
- ファイルシステム破損の原因 552
- ファイルシステム用のカスタムパラメータ 589
- ファイルとディレクトリのアクセス制御 36, 67
- ファイルとファイルシステム
- UFS ファイルシステム 213, 215
- ファイルのアクセス 189
- ファイルマネージャ
- フロッピーディスクのフォーマット 210, 215
- 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) 694
- フィールド
- passwd ファイル 52
- shadow ファイル 54
- グループファイル 56
- フィールド (passwd ファイル) 52
- フィールド (shadow ファイル) 54, 55
- フィールド (グループファイル) 55, 56
- ブート
- PC BIOS 177
- カーネルデバッグ (kadb) の使用 157
- クラッシュダンプの強制とリブート 156, 173
- 実行レベル 3 147, 162
- 実行レベル S 148, 163
- 自動チェックの変更 564
- 自動ブートチェック機能の変更 562
- 対話式 149
- ブートプロセス 176, 182
- 復元のため 153, 167
- ブート PROM の設定 143
- ブートする、IA: システムをネットワーク経由で 166
- ブートする、SPARC システムをネットワーク経由で 151
- ブートタイプの説明 109
- ブートブロック 585
- ブートブロックのインストール
- IA 419

SPARC 402
フォーマット
PCMCIA メモリーカード 236, 241
フロッピーディスク 206 - 212, 215, 217
復元
準備 633
対話式 637
テープドライブのタイプ 634
特定のファイル 635, 640
ファイルシステム全体 643
リモートドライブからの復元 642
不正な . と .. エントリ 561
物理デバイス名 336
フラグメントサイズ 590
不良 i ノード番号 560
不良スーパーブロック 568
不良スーパーブロックの復元 568
不良ファイルシステムの修正 570
不良ブロック数 559
プリンタ (デフォルト) 63
プロセスファイルシステム 452
ブロック
空き 588
間接 559
記憶 588
重複 559
通常ファイル 561
ディレクトリデータ 560
特殊 i ノード 557
ブート 585
論理サイズ 589
ブロックサイズ 470
ブロック数
不良 559
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 338
定義 337
フロッピーディスク
UFS フロッピーディスク 207, 210 - 213,
215, 264
アクセス 189, 257, 262
コマンド 205
情報の移動 222, 224
情報のコピー 222 - 224
取り出す 225
内容の検査 221
名前 206
ファイルのコピー 690

ファイルのリスト表示 (tar) 692
ファイルをコピー (tar) 691
ファイルを取り出す (tar) 693
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) 694
フォーマット 206 - 212, 215, 217
フロッピーディスクが使用中かどうかを
調べる 224
フロッピーディスクにアクセスするプロ
セスの終了 224
ボリューム管理下のディレクトリ 258
マウント 188, 226, 227
マウントポイント 260, 262
リモートアクセス 226 - 228
ロード 218, 221
フロッピーディスクドライブ
アクセスポイント 189
ボリューム管理下のディレクトリ 258
へ
ベースディレクトリ (basedir) 274, 276
ヘッダー形式
別のヘッダー形式でファイルをコピー
(cpio) 695
別名
削除 50
ユーザーログイン名 38
変更 269
ユーザー ID 番号 49
ユーザーアカウント 46, 49
ユーザーアカウント用のディレクトリ所
有権の変更 49
ユーザーアカウント用のファイル所有権
の変更 49
ユーザーパスワード 42, 46, 50, 54
ユーザーログイン名 49
[変更] オプション ([編集] メニュー)
Admintool: グループ 85
Admintool: ユーザー 49
[編集] メニュー
Admintool: グループ 77, 85
Admintool: ユーザー 79
変数 269
環境 61, 67
シェル (ローカル) 62
シェル (ローカル) 変数 65

ほ

- ホットプラグ機能 315
- ボリューム管理 257, 265
 - CD 189, 257, 259, 260, 262, 263
 - UFS の制限 264
 - 再起動 203
 - 終了 203
 - シンボリックリンク 262
 - 設定 203
 - 手作業によるマウントとの比較 188
 - 取り外し可能媒体へのアクセス 257
 - フロッピーディスク 189, 221, 257 - 260, 262
- 利点 188
- ボリューム管理の使用
 - フロッピーディスク 218

ま

- マウント
 - LOFS ファイルシステム 486
 - NFS ファイルシステム 485, 491
 - PCMCIA メモリーカード 251
 - UFS ファイルシステム 485
 - s5fs ファイルシステム 492
 - vfstab ファイル内のすべてのファイル 486
 - 手作業によるマウントとの比較 188
 - デフォルトオプションの使用 488, 492
 - ファイルシステム 459
 - ファイルシステムの自動マウント 463
 - マウントポイント 196, 260, 262
 - ユーザーのホームディレクトリ 44, 80, 82
 - リモート CD 196, 197
 - リモートのフロッピーディスク 226, 227
- マウントテーブル 461
- マウントポイント 460
- マスク 67
- マニュアルページ 454

み

- 密度
 - フロッピーディスクのフォーマット 208
 - 211, 215
- 未割り当て i ノード 557
- 未割り当てディレクトリブロック 560

む

- 無効にする
 - 実行制御スクリプト 124
 - ユーザーアカウント 46, 50, 55

め

- メール別名
 - 削除 50
 - ユーザーログイン名 38
- メールボックス (ユーザー) 50
- メモリー記憶域 (仮想) 456, 540

も

- モニター (PROM) 175

ゆ

- ユーザー ID 番号 38, 39, 49
- ユーザーアカウント 36
 - ID 番号 38, 39, 49
 - ガイドライン 37, 44
 - 管理ツール 46
 - 検索 (コマンド) 46
 - コピー 46
 - コマンド 46
 - 削除 46, 50
 - 情報の格納 37, 51
 - 設定 74
 - 説明 36
 - ソート (コマンド) 46
 - 追加 46, 79
 - デフォルト設定 46
 - ネームサービス 37, 46, 51
 - 変更 46, 49
 - 無効に、またはロックする 46, 50, 55
 - ログイン名 36, 37, 49
- ユーザーアカウントのソート 46
- ユーザーアクセス 36
- ユーザー初期設定ファイル
 - カスタマイズ 44, 50, 58 - 60, 62, 65, 67, 69, 75
 - シェル 58, 60, 69
 - 説明 36, 44, 58
 - デフォルト 58
 - 例 68

- ユーザー登録
 - solregis コマンド 95
 - 説明 94
 - 無効にする 97
 - 問題 95
- 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 44, 50
 - 共有 80, 82
 - サイズ 43
 - 削除 50
 - 所有権の変更 49
 - 説明 36, 43
 - マウント 44, 80, 82
 - ローカル以外からの \$HOME の参照 44
 - ローカル以外からの \$HOME への参照 60
- ユーザーのメールボックス 50
- ユーザーパスワードの有効期限 43
- ユーザーマスク 67
- ユーザーマネージャ、Solstice AdminSuite 46
- ユーザーログイン (疑似) 39
- ユーザーログイン名
 - 説明 36, 37
 - 変更 49
- ユーザーログイン名における下線 () 37

よ

- 呼び名
 - CD 192
 - フロッピーディスク 206

り

- リスト表示
 - テープ上のファイル (cpio) 684
 - テープ上のファイル (tar) 679
 - フロッピーディスク上のファイル (tar) 692
- リセットボタン 171
- リトルエンディアン方式のビットコード化 205

- リモートアクセス
 - CD 196 - 198
 - フロッピーディスク 226 - 228
- リモートドライブ (からの復元) 642
- 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ソフトウェアのインストール 283
 - ソフトウェアのインストール 280, 281
- リモートマウント 80, 82
- 領域最適化タイプ 592
- 履歴環境変数 63

る

- ルート (/) ファイルシステム 455

れ

- レベル 0 バックアップ 610

ろ

ロード

- CD 193
- PCMCIA メモリーカード 244
- フロッピーディスク 218, 221

ログイン名 (ユーザー)

- 説明 36, 37
- 変更 49
- ログ (ダンプの記録) 652
- 論理デバイス名
 - CD-ROM 342
 - 定義 337
 - ディスク 337
 - テープ 341
- 論理ブロックサイズ 589

わ

- ワールド (アクセス権) 68
- 割り当て済み i ノード 557