



# Solaris のシステム管理 (デバイス とファイルシステム)

---

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

Part No: 817-7239-11  
2004 年 9 月

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面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。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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

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

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

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

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

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

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

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

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

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、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: Devices and File Systems*

Part No: 817-6960-11

Revision A



040727@9495



# 目次

---

はじめに 19

- 1 リムーバブルメディアの管理 (概要) 25
  - リムーバブルメディアの管理についての参照先 25
  - リムーバブルメディアの機能と利点 26
  - 自動マウントと手動によるマウントの比較 26
  - ボリューム管理を使って行える操作 27
  
- 2 リムーバブルメディアへのアクセス (手順) 29
  - リムーバブルメディアへのアクセス (作業マップ) 29
  - リムーバブルメディアへのアクセス (概要) 30
    - リムーバブルメディア名の使用 30
    - リムーバブルメディア上のデータにアクセスするためのガイドライン 32
      - ▼ 新しいリムーバブルメディアドライブを追加する方法 33
      - ボリューム管理 (vol1d) の終了と起動 33
      - ▼ リムーバブルメディア上の情報にアクセスする方法 34
      - ▼ リムーバブルメディア上の情報をコピーする方法 35
      - ▼ 音楽用 CD/DVD を再生する方法 35
      - ▼ リムーバブルメディアが使用中かどうかを調べる方法 37
      - ▼ リムーバブルメディアを取り出す方法 37
    - リモートシステム上のリムーバブルメディアへのアクセス (作業マップ) 38
      - ▼ ローカルのメディアを他のシステムで使用可能にする方法 38
      - ▼ リモートシステム上のリムーバブルメディアにアクセスする方法 42

- 3 リムーバブルメディアのフォーマット (手順) 45
  - リムーバブルメディアのフォーマット (作業マップ) 45
  - リムーバブルメディアのフォーマットの概要 46
    - リムーバブルメディアのフォーマットのガイドライン 46
    - リムーバブルメディアのハードウェア面での考慮事項 47
    - ▼ リムーバブルメディアを読み込む方法 49
    - ▼ リムーバブルメディアをフォーマットする方法 (rmformat) 50
    - ▼ ファイルシステムの追加用にリムーバブルメディアをフォーマットする方法 51
    - ▼ リムーバブルメディア上のファイルシステムを検査する方法 53
    - ▼ リムーバブルメディア上の不良ブロックを修復する方法 54
    - リムーバブルメディアに読み取り/書き込み保護とパスワードによる保護を適用する 54
    - ▼ リムーバブルメディアの書き込み保護を有効または無効にする方法 54
    - ▼ Iomega メディアの読み取り/書き込み保護とパスワードを有効または無効にする方法 55
  
- 4 CD への書き込み (手順) 57
  - オーディオ CD やデータ CD の取り扱い 57
    - CD メディアに関するよく使われる用語 58
  - データ CD やオーディオ CD への書き込み 59
    - RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する 60
    - ▼ RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法 60
    - CD デバイスを確認する方法 61
    - ▼ CD メディアをチェックする方法 61
    - データ CD を作成する 62
    - ▼ データ CD 用の ISO 9660 ファイルシステムを作成する方法 62
    - ▼ マルチセッションのデータ CD を作成する方法 63
    - オーディオ CD を作成する 65
    - ▼ オーディオ CD を作成する方法 65
    - ▼ オーディオトラックを CD から抽出する方法 66
    - ▼ CD をコピーする方法 67
    - ▼ CD-RW メディアを消去する方法 68
  
- 5 デバイスの管理 (手順) 69
  - デバイス管理の新機能 69
    - USB デバイスの新機能 70

デバイス管理作業についての参照先	70
デバイスドライバについて	70
デバイスの自動構成	71
自動構成の機能と利点	71
標準サポートされていないデバイスを使用する場合	72
デバイス構成情報の表示	73
driver not attached メッセージ	73
システムデバイスの識別	73
システム構成情報を表示する方法	74
デバイス情報を表示する方法	76
システムへ周辺デバイスを追加する	77
▼ 周辺デバイスを追加する方法	77
▼ デバイスドライバを追加する方法	78
<b>6 デバイスの動的構成 (手順)</b>	<b>81</b>
動的再構成とホットプラグ機能	81
接続点	82
x86: PCI アダプタカードの取り外し	84
cfgadm コマンドによる SCSI ホットプラグ (作業マップ)	85
cfgadm コマンドによる SCSI ホットプラグ	86
▼ SCSI デバイスに関する情報を表示する方法	86
▼ SCSI コントローラの構成を解除する方法	87
▼ SCSI コントローラを構成する方法	87
▼ SCSI デバイスを構成する方法	88
▼ SCSI コントローラを切り離す方法	89
▼ SPARC: SCSI コントローラを接続する方法	90
▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法	90
▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法	91
▼ SPARC: SCSI デバイスを取り外す方法	92
SPARC: SCSI 構成に関する問題の障害追跡	93
▼ 失敗した SCSI 構成解除操作の解決方法	95
cfgadm コマンドによる PCI ホットプラグ (作業マップ)	95
x86: cfgadm コマンドによる PCI ホットプラグ	96
▼ x86: PCI スロット構成情報を表示する方法	96
▼ x86: PCI アダプタカードを取り外す方法	97
▼ x86: PCI アダプタカードを取り付ける方法	98
x86: PCI 構成に関する問題の障害追跡	99
Reconfiguration Coordination Manager (RCM) スクリプトの概要	99

RCM スクリプトについて	100
RCM スクリプトで実行できること	100
RCM スクリプト処理の動作方法	100
RCM スクリプトでの作業	101
アプリケーション開発者 RCM スクリプト (作業マップ)	102
システム管理者 RCM スクリプト (作業マップ)	103
RCM スクリプトに名前を付ける	103
RCM スクリプトのインストールまたは削除	103
▼ RCM スクリプトのインストール方法	104
▼ RCM スクリプトの削除方法	104
▼ RCM スクリプトのテスト方法	105
テープバックアップ用の RCM スクリプトの例	105
<b>7 USB デバイスの使用 (概要)</b>	<b>111</b>
USB デバイスの新機能	111
USB 二重フレームワーク	111
Solaris がサポートしている USB デバイス	112
SPARC: USB 2.0 の機能	114
USB 大容量ストレージデバイス	116
SPARC: USB ドライバの機能拡張	118
USB デバイスの概要	119
よく使用される USB 関連の略語	120
USB バスの説明	120
Solaris 環境における USB について	123
USB キーボードとマウス	123
USB ホストコントローラとルートハブ	124
SPARC: USB 電源管理	125
USB ケーブルに関するガイドライン	126
<b>8 USB デバイスの使用 (手順)</b>	<b>127</b>
Solaris 環境での USB デバイスの管理 (作業マップ)	127
USB 大容量ストレージデバイスの使用 (作業マップ)	128
USB 大容量ストレージデバイスの使用	129
void を使用して USB 大容量ストレージデバイスを使用できるように準備する方 法	130
▼ void を使用しないで USB 大容量ストレージデバイスを使用できるように準備 する方法	131
USB デバイス情報を表示する方法 (prtconf)	132

▼ vol1d を使用しないで USB 大容量ストレージデバイスをフォーマットする方 法	133
▼ vol1d を使用して USB 大容量ストレージデバイスをマウントまたはマウント解 除する方法	135
▼ vol1d を使用しないで USB 大容量ストレージデバイスをマウントまたはマウン ト解除する方法	136
特定の USB ドライバを無効にする	137
▼ 特定の USB ドライバを無効にする方法	137
▼ 使用されていない USB デバイスのリンクを削除する方法	138
USB デバイスのホットプラグ (作業マップ)	139
▼ vol1d を使用して USB 大容量ストレージデバイスを追加する方法	140
▼ vol1d を使用しないで USB 大容量ストレージデバイスを追加する方法	140
▼ vol1d を使用して USB 大容量ストレージデバイスを取り外す方法	141
▼ vol1d を使用しないで USB 大容量ストレージデバイスを取り外す方法	141
▼ USB カメラを追加する方法	142
USB オーディオデバイスの使用 (作業マップ)	143
USB オーディオデバイスの使用	144
複数の USB オーディオデバイスのホットプラグ	145
▼ USB オーディオデバイスを追加する方法	145
▼ システムの主オーディオデバイスを識別する方法	146
主 USB オーディオデバイスを変更する方法	147
USB オーディオデバイスに関する問題の障害追跡	147
cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)	148
cfgadm コマンドを使った USB デバイスのホットプラグ	148
USB バス情報を表示する方法 (cfgadm)	149
▼ USB デバイスの構成を解除する方法	150
▼ USB デバイスの構成方法	151
▼ 論理的に USB デバイスを接続解除する方法	151
▼ 論理的に USB デバイスを接続する方法	151
▼ 論理的に USB デバイスのサブツリーを接続解除する方法	152
▼ USB デバイスのリセット方法	152
▼ 複数の構成を持つ USB デバイスのデフォルト構成を変更する方法	153
<b>9 デバイスへのアクセス (概要)</b>	<b>155</b>
デバイスへのアクセス	155
デバイス情報が作成される方法	155
デバイスの管理方法	156
デバイス名の命名規則	156

論理ディスクデバイス名	157
ディスクサブディレクトリの指定	157
スライスの指定	158
x86: 直接コントローラでアクセスされるディスク	159
SPARC: バス指向コントローラでアクセスされるディスク	159
x86: SCSI コントローラでアクセスされるディスク	160
論理テープデバイス名	160
論理リムーバブルメディアデバイス名	161

<b>10</b>	<b>ディスクの管理 (概要)</b>	<b>163</b>
	Solaris 9 アップデートリリースの新しいディスク管理機能	163
	SPARC: EFI ディスクラベルによるマルチテラバイトディスクのサポート	163
	ディスク管理作業についての参照先	168
	ディスク管理の概要	168
	ディスク関連の用語	169
	ディスクスライスについて	169
	SPARC: ディスクスライス	170
	x86: ディスクスライス	172
	raw データスライスの使用	173
	複数のディスク上のスライス配置	173
	使用するスライスの決定	174
	format ユーティリティ	175
	format ユーティリティを使用する場合	176
	format ユーティリティ使用上のガイドライン	176
	ディスクのフォーマット	178
	ディスクラベルについて	178
	パーティションテーブル	179
	パーティションテーブル情報の表示	179
	ディスクをスライスに分割する	181
	free hog スライスの使用方法	182
<b>11</b>	<b>ディスクの管理 (手順)</b>	<b>183</b>
	ディスクの管理 (作業マップ)	183
	システム上のディスクの確認	184
	▼ システム上のディスクを確認する方法	185
	ディスクのフォーマット	186
	▼ ディスクがフォーマット済みかを調べる方法	187



- ▼ ディスクをフォーマットする方法 187
  - ディスクスライスの表示 189
    - ▼ ディスクスライス情報を表示する方法 189
  - ディスクラベルの作成と検査 191
    - ▼ ディスクラベルを作成する方法 191
    - ▼ ディスクラベルを検査する方法 193
  - 破損したディスクラベルの復元 195
    - ▼ 破損したディスクラベルを復元する方法 195
  - Sun 製品以外のディスクの追加 197
    - format.dat のエントリの作成 198
      - ▼ format.dat のエントリを作成する方法 198
  - SCSI ディスクドライブの自動構成 199
    - ▼ SCSI ドライブを自動構成する方法 200
  - 欠陥セクターの修復 201
    - ▼ 表面解析を使用して欠陥セクターを調べる方法 202
    - ▼ 欠陥セクターを修復する方法 203
  - ディスク管理のヒント 203
    - format セッションのデバッグ 203
    - 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 204
- 12 SPARC: ディスクの追加 (手順) 207**
- SPARC: システムディスクまたは二次ディスクの追加 (作業マップ) 207
    - SPARC: システムディスクまたは二次ディスクの追加 208
      - ▼ SPARC: システムディスクを接続してブートする方法 209
      - ▼ SPARC: 二次ディスクを接続してブートする方法 209
      - ▼ SPARC: ディスクスライスとディスクラベルを作成する方法 211
      - ▼ SPARC: UFS ファイルシステムを作成する方法 215
      - ▼ 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 216
- 13 x86: ディスクの追加 (手順) 217**
- x86: システムディスクまたは二次ディスクの追加 (作業マップ) 217
  - x86: システムディスクまたは二次ディスクの追加 218
    - ▼ x86: システムディスクを接続してブートする方法 219
    - ▼ x86: 二次ディスクを接続してブートする方法 219
    - x86: fdisk パーティションの作成上のガイドライン 220
      - ▼ x86: Solaris fdisk パーティションを作成する方法 221
      - ▼ x86: ディスクスライスとディスクラベルを作成する方法 227

- ▼ x86: ファイルシステムを作成する方法 228
- ▼ x86: システムディスクにブートブロックをインストールする方法 229

<b>14</b>	<b>format</b>	<b>ユーティリティ (参照情報)</b>	<b>231</b>
	format	ユーティリティを使用する上での推奨事項および要件	231
	format	のメニューとコマンドの説明	232
		partition	メニュー 234
		x86: fdisk	メニュー 235
		analyze	メニュー 236
		defect	メニュー 237
	format.dat	ファイル	238
		format.dat	ファイルの内容 239
		format.dat	ファイルの構文 239
		format.dat	ファイル中のキーワード 239
		パーティションまたはスライステーブル (format.dat)	242
		format	ユーティリティの代替データファイルを指定する 242
	format	コマンドへの入力規則	243
		format	コマンドへ番号を指定する 243
		format	コマンドへブロック番号を指定する 243
		format	のコマンド名を指定する 244
		format	コマンドへディスク名を指定する 245
	format	ユーティリティのヘルプを利用する	245
<b>15</b>		<b>ファイルシステムの管理 (概要)</b>	<b>247</b>
	Solaris 9 Update	リリースの新しいファイルシステム機能	247
		デフォルトで有効な UFS ロギング	247
		SPARC: マルチテラバイトの UFS ファイルシステムのサポート	249
		ファイルシステム管理作業についての参照先	255
		ファイルシステムの概要	255
		ファイルシステムのタイプ	256
		ディスクベースのファイルシステム	256
		ネットワークベースのファイルシステム	257
		仮想ファイルシステム	258
		拡張ファイル属性	261
		ファイルシステム管理用のコマンド	262
		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	263
		汎用コマンドと専用コマンドのマニュアルページ	263

	デフォルトの Solaris ファイルシステム	264
	スワップ空間	265
	UFS ファイルシステム	265
	UFS ファイルシステムの計画	266
	UFS ログイン	267
	UFS スナップショット	268
	UFS 直接入出力	268
	ファイルシステムのマウントとマウント解除	268
	マウントされたファイルシステムテーブル	270
	仮想ファイルシステムテーブル	271
	NFS 環境	272
	自動マウントまたは AutoFS	272
	ファイルシステムのタイプを調べる	273
	ファイルシステムのタイプを調べる方法	273
<b>16</b>	<b>UFS、TMPFS、LOFS ファイルシステムの作成 (手順)</b>	<b>275</b>
	UFS ファイルシステムの作成	275
	newfs コマンドのデフォルトのパラメータ	276
	▼ UFS ファイルシステムを作成する方法	277
	一時ファイルシステム (TMPFS) の作成	278
	▼ TMPFS ファイルシステムを作成する方法	279
	ループバックファイルシステム (LOFS) の作成	280
	▼ LOFS ファイルシステムを作成する方法	280
<b>17</b>	<b>ファイルシステムのマウントとマウント解除 (手順)</b>	<b>283</b>
	ファイルシステムのマウントの概要	283
	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	284
	汎用マウントオプション	285
	/etc/vfstab ファイルのフィールドの説明	287
	ファイルシステムのマウント	288
	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	289
	▼ /etc/vfstab ファイルにエントリを追加する方法	289
	▼ 1つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)	291
	▼ UFS ファイルシステムのマウント方法 (mount コマンド)	292
	▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)	293
	▼ NFS ファイルシステムのマウント方法 (mount コマンド)	294

▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)	295
ファイルシステムのマウント解除	296
ファイルシステムをマウント解除する場合の前提条件	297
ファイルシステムのマウント解除を確認する方法	297
▼ 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	297
▼ 1つのファイルシステムをマウント解除する方法	298
<b>18 CacheFS ファイルシステムの使用 (手順)</b>	<b>301</b>
CacheFS ファイルシステムの概観 (作業マップ)	301
CacheFS ファイルシステムの概要	302
CacheFS ファイルシステムの機能	302
CacheFS ファイルシステムの構造と動作	303
CacheFS ファイルシステムの作成とマウント (作業マップ)	304
▼ キャッシュを作成する方法	305
ファイルシステムをキャッシュにマウントする	305
▼ CacheFS ファイルシステムのマウント方法 (mount)	306
▼ CacheFS ファイルシステムのマウント方法 (/etc/vfstab)	308
▼ CacheFS ファイルシステムのマウント方法 (AutoFS)	309
CacheFS ファイルシステムの保守 (作業マップ)	310
CacheFS ファイルシステムの保守	310
CacheFS ファイルシステムの変更	311
▼ CacheFS ファイルシステムに関する情報を表示する方法	311
CacheFS ファイルシステムの整合性チェック	312
▼ 必要に応じてキャッシュの整合性チェックを指定する方法	312
▼ CacheFS ファイルシステムを削除する方法	313
▼ CacheFS ファイルシステムの整合性をチェックする方法	314
CacheFS ファイルシステムのバックアップ (作業マップ)	315
CacheFS ファイルシステムのバックアップ	316
キャッシュにファイルをバックする方法	316
バックされたファイルの情報を表示する方法	317
バックアップリストの使用	318
バックアップリストを作成する方法	319
バックアップリストを使ってファイルをキャッシュにバックする方法	319
キャッシュからファイルまたはバックアップリストのバックアップを解除する	320
キャッシュからファイルまたはバックアップリストのバックアップを解除する方法	320
cachefspack エラーの障害追跡	321

	CacheFS の統計情報の収集 (作業マップ)	325
	CacheFS の統計情報の収集	325
	▼ CacheFS ログインを設定する方法	327
	CacheFS ログファイルの場所を調べる方法	327
	CacheFS ログインを停止する方法	328
	▼ 作業セット (キャッシュ) のサイズを表示する方法	328
	CacheFS の統計情報の表示	329
	CacheFS 統計情報を表示する方法	329
<b>19</b>	<b>追加スワップ空間の構成 (手順)</b>	<b>331</b>
	スワップ空間について	331
	スワップ空間と仮想メモリー	332
	スワップ空間と TMPFS ファイルシステム	332
	ダンプデバイスとしてのスワップ空間	333
	スワップ空間と動的再構成	333
	スワップ空間の追加が必要かどうかを調べる方法	334
	スワップ関連のエラーメッセージ	334
	TMPFS 関連のエラーメッセージ	334
	スワップ空間の割り当て方法	335
	/etc/vfstab ファイル	335
	スワップ空間の計画	336
	スワップリソースの監視	336
	スワップ空間の追加	338
	スワップファイルの作成	338
	▼ スワップファイルを作成して使用可能にする方法	339
	スワップファイルの削除	340
	▼ 不要になったスワップ空間を削除する方法	340
<b>20</b>	<b>UFS ファイルシステムの整合性チェック (手順)</b>	<b>343</b>
	ファイルシステムの整合性	344
	ファイルシステムの状態はどのように記録されるか	344
	fsck コマンドでチェックして修復される内容	346
	不整合が発生する原因	346
	整合性がチェックされる UFS 構成要素	347
	fsck 要約メッセージ	352
	UFS ファイルシステムを対話式でチェックして修復する	352
	▼ 代替ブートデバイスからルート (/) ファイルシステムまたは /usr ファイルシステムをチェックする方法	353

▼ ルート (/) 以外、または /usr 以外のファイルシステムをチェックする方 法	355
UFS ファイルシステムの修復	357
▼ UFS ファイルシステムを修復する方法	357
fsck コマンドで修復できない UFS ファイルシステムの修正	358
不正なスーパーブロックの復元	359
▼ 不正なスーパーブロックを復元する方法	359
fsck コマンドの構文とオプション	361
<b>21 UFS ファイルシステム (参照情報)</b>	<b>363</b>
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	363
プラットフォームに依存するディレクトリ	371
UFS ファイルシステムのシリンダグループの構造	372
ブートブロック	372
スーパーブロック	372
i ノード	373
データブロック	374
空きブロック	374
カスタムファイルシステムパラメータ	375
論理ブロックサイズ	375
フラグメントサイズ	376
最小空き容量	377
回転待ち	377
最適化のタイプ	377
i ノード数 (ファイルの数)	378
UFS ファイルとファイルシステムの最大サイズ	378
UFS サブディレクトリの最大数	378
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	379
newfs コマンドの構文、オプション、引数	379
汎用 mkfs コマンド	382
<b>22 ファイルシステムのバックアップと復元 (概要)</b>	<b>383</b>
バックアップと復元についての参照先	383
ファイルシステムのバックアップと復元とは	384
ファイルシステムをバックアップする理由	385
バックアップを作成するファイルシステムの計画	385
バックアップタイプの選択	388
テープデバイスの選択	388

	ファイルシステムのバックアップおよび復元の概要 (作業マップ)	389
	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	390
	バックアップ頻度	390
	バックアップの用語と定義	391
	バックアップスケジュールに関するその他の推奨事項	391
	ダンプレベルを使用して増分バックアップを作成する	393
	バックアップスケジュールの例	394
	例 — 日単位累積、週単位累積バックアップ	394
	例 — 日単位累積、週単位増分バックアップ	395
	例 — 日単位増分、週単位累積バックアップ	396
	例—サーバーの月単位バックアップスケジュール	397
<b>23</b>	<b>ファイルとファイルシステムのバックアップ (手順)</b>	<b>401</b>
	ファイルとファイルシステムのバックアップ (作業マップ)	401
	ファイルシステムバックアップの実行準備	402
	▼ ファイルシステム名を検索する方法	402
	▼ 完全バックアップに必要なテープ数を判別する	403
	ファイルシステムのバックアップ	403
	▼ ファイルシステムのバックアップをテープに作成する方法	404
<b>24</b>	<b>UFS スナップショットの使用 (手順)</b>	<b>411</b>
	UFS スナップショットの使用 (作業マップ)	411
	UFS スナップショットの概要	412
	なぜ UFS スナップショットを使用するか	412
	UFS スナップショットのパフォーマンス上の問題	413
	UFS スナップショットの作成と削除	413
	▼ UFS スナップショットを作成する方法	414
	▼ UFS スナップショットの情報を表示する方法	415
	UFS スナップショットの削除	416
	▼ UFS スナップショットを削除する方法	416
	UFS スナップショットのバックアップ	417
	▼ UFS スナップショットの完全バックアップを作成する方法 (ufsdump)	417
	▼ UFS スナップショットの増分バックアップの作成方法 (ufsdump)	418
	▼ UFS スナップショットのバックアップ方法 (tar)	418
	UFS スナップショットのバックアップからのデータの復元	419

- 25 ファイルとファイルシステムの復元 (手順) 421
  - ファイルおよびファイルシステムのバックアップを復元する (作業マップ) 421
  - ファイルとファイルシステムを復元するための準備 422
    - ファイルシステム名の確認 423
    - 必要なテープデバイスのタイプの決定 423
    - テープデバイス名の決定 423
  - ファイルシステムの復元 423
    - ▼ 使用するテープを決定する方法 424
    - ▼ 対話式でファイルを復元する方法 425
    - ▼ 特定のファイルを復元する方法 427
    - ▼ ファイルシステム全体を復元する方法 429
    - ▼ ルート (/) と /usr を復元する方法 432
  
- 26 UFS バックアップおよび復元コマンド (参照情報) 435
  - ufsdump コマンドの機能 435
    - デバイス特性の判断 435
    - メディアの終わりの検出 436
  - ufsdump コマンドを使用したデータのコピー 436
    - /etc/dumpdates ファイルの目的 436
    - バックアップデバイス (*dump-file*) 引数 437
    - バックアップを作成するファイルを指定する 439
    - テープの性質を指定する 440
    - ufsdump の制限 440
  - ufsdump コマンドのオプションと引数 440
    - ufsdump のデフォルトオプション 440
    - ufsdump コマンドのオプション 441
  - ufsdump とセキュリティに関する注意事項 443
  - ufsrestore コマンドのオプションと引数 443
  
- 27 UFS ファイルとファイルシステムのコピー (手順) 449
  - ファイルシステムをコピーするためのコマンド 450
  - ファイルシステムをディスク間でコピーする 451
    - ファイルシステムのリテラルコピーを作成する 451
      - ▼ ディスクをコピーする方法 (dd) 452
  - 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 454
    - ▼ 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) 455
  - ファイルとファイルシステムをテープにコピーする 456



- tar を使用してファイルをテープにコピーする 457
  - ▼ ファイルをテープにコピーする方法 (tar) 458
  - ▼ テープ上のファイルのリストを表示する方法 (tar) 459
  - ▼ テープからファイルを取り出す方法 (tar) 459
- pax コマンドを使用してファイルをテープにコピーする 460
  - ▼ ファイルをテープにコピーする方法 (pax) 460
- cpio コマンドを使用してファイルをテープにコピーする 461
  - ▼ 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) 461
  - ▼ テープ上のファイルのリストを表示する方法 (cpio) 462
  - ▼ テープからすべてのファイルを取り出す方法 (cpio) 463
  - ▼ テープから特定のファイルを取り出す方法 (cpio) 464
- ファイルをリモートテープデバイスにコピーする 465
  - ▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd) 465
  - ▼ ファイルをリモートテープデバイスから取り出す方法 467
- 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 468
  - 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 468
  - ▼ 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) 468
  - ▼ 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) 469
  - ▼ ファイルをフロッピーディスクから取り出す方法 (tar) 470
  - 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 471

## 28 テープドライブの管理 (手順) 473

- 使用するメディアの選択 473
- バックアップデバイス名 474
  - テープドライブの巻き戻しオプションを指定する 475
  - テープドライブに別の密度を指定する 476
- テープドライブの状態を表示する 476
  - ▼ テープドライブの状態を表示する方法 476
- 磁気テープカートリッジの取り扱い 477
  - 磁気テープカートリッジのたるみを直す方法 477
  - 磁気テープカートリッジを巻き戻す方法 478
- ドライブの管理とメディア処理のガイドライン 478

索引 481



## はじめに

---

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

本書では、次のことを前提としています。

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

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

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

---

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

---

---

注 - このマニュアル内で引用する第三者の Web サイトの可用性について Sun は責任を負いません。こうしたサイトやリソース上の、またはこれらを通じて利用可能な、コンテンツ、広告、製品、その他の素材について、Sun は推奨しているわけではなく、Sun はいかなる責任も負いません。こうしたサイトやリソース上の、またはこれらを経由して利用可能な、コンテンツ、製品、サービスを利用または信頼したことに伴って発生した (あるいは発生したと主張される) いかなる損害や損失についても、Sun は一切の責任を負いません。

---

---

## 対象読者

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

---

## Solaris システム管理マニュアルセットの構成

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

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

マニュアル名	内容
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティ、BART、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、Solaris Secure Shell、および SEAM

## Sun のオンラインマニュアル

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

## 表記上の規則

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

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%
<b>AaBbCc123</b>	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% <b>su</b> 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。

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
\	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% <b>grep</b> `^#define \  XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

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

■ C シェルのスーパーユーザー

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

■ Bourne シェルおよび Korn シェル

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

■ Bourne シェルおよび Korn シェルのスーパーユーザー

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

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

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

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

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

## 一般規則

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

- このマニュアル中の手順を実行したり、例 (コマンド入力、コードなど) を使用する場合には、二重引用符 ("), 左一重引用符 ('), 右一重引用符 (') をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。

- /sbin、/usr/sbin、/usr/bin、/etc ディレクトリにあるコマンドについては、このマニュアルでは絶対パス名で表記していません。ただし、それ以外のあまり一般的でないディレクトリにあるコマンドについては、このマニュアル中の例では絶対パスで表記します。
- このマニュアル中の例は、SunOS ソフトウェアが標準的にインストールされていることを前提としています。つまり、バイナリ互換パッケージがインストールされていることや /usr/ucb が検索パスに設定されていることは、前提としていません。



---

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

---





## 第 1 章

---

# リムーバブルメディアの管理 (概要)

---

この章では、Solaris 環境でリムーバブルメディアを管理するための一般的なガイドラインについて説明します。

この章の内容は次のとおりです。

- 25 ページの「リムーバブルメディアの管理についての参照先」
- 26 ページの「リムーバブルメディアの機能と利点」
- 26 ページの「自動マウントと手動によるマウントの比較」
- 27 ページの「ボリューム管理を使って行える操作」

---

## リムーバブルメディアの管理についての参照先

リムーバブルメディアを管理する手順については、次を参照してください。

リムーバブルメディアの管理作業	参照先
リムーバブルメディアへのアクセス	第 2 章
リムーバブルメディアのフォーマット	第 3 章
データ CD とオーディオ CD の作成	第 4 章

共通デスクトップ環境のファイルマネージャでリムーバブルメディアを使用する方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』を参照してください。

---

## リムーバブルメディアの機能と利点

Solaris 環境には、ユーザーとソフトウェア開発者用に、リムーバブルメディアを扱うための標準インタフェースが用意されています。ボリューム管理と呼ばれるこのインタフェースには、主に次の3つの利点があります。

- リムーバブルメディアを自動的にマウントすることによって、操作を簡単にします。手動によるマウントと自動マウントの比較については、次の節を参照してください。
- スーパーユーザーでなくても、リムーバブルメディアにアクセスできるようになります。
- ネットワーク上の他のシステムがローカルシステム上のリムーバブルメディアに自動的にアクセスできるようになります。詳細については、[第2章](#)を参照してください。

---

## 自動マウントと手動によるマウントの比較

次の表は、リムーバブルメディアの手動によるマウント (ボリューム管理を使用しない場合) と自動マウント (ボリューム管理を使用する場合) に関する手順を比較したものです。

表 1-1 手動によるマウントと自動マウントの比較

手順	手動によるマウント	自動マウント
1	メディアを挿入する	メディアを挿入する
2	スーパーユーザーになる	フロッピーディスクの場合は、 <code>volcheck</code> コマンドを使用する
3	メディアデバイスの位置を確認する	ボリュームマネージャ ( <code>voladm</code> ) は、リムーバブルメディアを手動でマウントするのに必要な作業のほとんどを自動的に実行する
4	マウントポイントを作成する	
5	マウントポイントが現在のディレクトリではないことを確認する	

表 1-1 手動によるマウントと自動マウントの比較 (続き)

手順	手動によるマウント	自動マウント
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	メディア上のファイルを操作する	メディア上のファイルを操作する
9	スーパーユーザーになる	
10	メディアデバイスのマウントを解除する	
11	メディアを取り出す	メディアを取り出す
12	スーパーユーザーを終了する	

## ボリューム管理を使って行える操作

ボリューム管理を使用すると、手動によるマウントの場合と同様にリムーバブルメディアにアクセスできますが、その操作ははるかに容易になり、スーパーユーザーになる必要もありません。リムーバブルメディアは、操作しやすいように、覚えやすい位置にマウントされます。

表 1-2 ボリュームマネージャによって管理されたリムーバブルメディア上のデータにアクセスする方法

アクセス	操作	ファイルを検索する場所
最初のフロッピーディスク上のファイル	フロッピーディスクを挿入して、volcheck と入力する	/floppy
最初のリムーバブルハードディスク上のファイル	リムーバブルハードディスクを挿入して、volcheck と入力する	/rmdisk/jaz0 または /rmdisk/zip0
最初の CD 上のファイル	CD を挿入して、数秒間待つ	/cdrom/volume-name
最初の DVD 上のファイル	DVD を挿入して、数秒間待つ	/dvd/volume-name
最初の PCMCIA 上のファイル	PCMCIA を挿入して、数秒間待つ	/pcmem/pcmem0

システムに複数の種類のリムーバブルメディアデバイスがある場合は、そのアクセスポイントについて、次の表を参照してください。

表 1-3 リムーバブルメディアにアクセスする場所

メディアデバイス	ファイルシステムにアクセスするためのパス	raw データにアクセスするための場所
最初のフロッピーディスクドライブ	/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
最初のリムーバブルハードディスク	/rmdisk/jaz0	/vol/dev/aliases/jaz0
	/rmdisk/zip0	/vol/dev/aliases/zip0
最初の PCMCIA ドライブ	/pcmem/pcmem0	/vol/dev/aliases/pcmem0

## 第 2 章

# リムーバブルメディアへのアクセス (手順)

この章では、Solaris 環境でコマンド行からリムーバブルメディアにアクセスする方法について説明します。

リムーバブルメディアへのアクセスに関連する手順については、次の節を参照してください。

- 29 ページの「リムーバブルメディアへのアクセス (作業マップ)」
- 38 ページの「リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)」

リムーバブルメディアの概要については、第 1 章を参照してください。

## リムーバブルメディアへのアクセス (作業マップ)

作業	説明	参照先
1. (省略可能) リムーバブルメディアドライブを追加する	必要に応じて、リムーバブルメディアドライブをシステムに追加する	33 ページの「新しいリムーバブルメディアドライブを追加する方法」
2. (省略可能) リムーバブルメディアにアクセスするときにボリューム管理 (vold) を使用するかどうかを決める	デフォルトでは、ボリューム管理 (vold) は実行される。リムーバブルメディアにアクセスするときにボリューム管理を使用するかどうかを決める	33 ページの「ボリューム管理 (vold) の終了と起動」

作業	説明	参照先
3. リムーバブルメディアにアクセスする	ボリューム管理を使用して/使用しないで、各種のリムーバブルメディアにアクセスする	34 ページの「リムーバブルメディア上の情報にアクセスする方法」
4. (省略可能) ファイルまたはディレクトリをコピーする	ファイルまたはディレクトリを、ファイルシステムの別の場所からコピーするように、リムーバブルメディアからコピーする	35 ページの「リムーバブルメディア上の情報をコピーする方法」
5. (省略可能) 音楽用 CD/DVD を再生するようにシステムを設定する	音楽用 CD/DVD を再生するようにシステムを設定できるが、再生には Sun 以外のソフトウェアが必要になる	35 ページの「音楽用 CD/DVD を再生する方法」
6. メディアが使用中であるかどうかを調べる	メディアを取り出す前に、それが使用中であるかどうかを調べる	37 ページの「リムーバブルメディアが使用中かどうかを調べる方法」
7. メディアを取り出す	作業が終了したら、ドライブからメディアを取り出す	37 ページの「リムーバブルメディアを取り出す方法」

## リムーバブルメディアへのアクセス (概要)

リムーバブルメディア上の情報にアクセスするには、ボリュームマネージャを使用する方法と使用しない方法があります。CDE のファイルマネージャを使用してリムーバブルメディア上の情報にアクセスする方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の「ファイル・マネージャでのリムーバブル・メディアの使い方」を参照してください。

Solaris 8 6/00 リリースから、ボリュームマネージャ (vold) ですべてのリムーバブルメディアデバイスを動的に管理できるようになりました。これに伴い、`/dev/rdsk/cntndnsn` や `/dev/dsk/cntndnsn` などのデバイス名でのリムーバブルメディアへのアクセスは、できなくなります。

## リムーバブルメディア名の使用

リムーバブルメディアには、さまざまな名前前でアクセスできます。次の表に、ボリューム管理を使用して/使用しないでアクセスできるさまざまなメディア名を示します。

表 2-1 リムーバブルメディア名

メディア	ボリューム管理デバイスの名前	ボリューム管理デバイスの別名	デバイス名
最初のフロッピーディスクドライブ	/floppy	/vol/dev/aliases/floppy0	/dev/rdiskette /vol/dev/rdiskette0/ <i>volume-name</i>
最初、2番目、3番目のCD-ROMまたはDVD-ROMドライブ	/cdrom0 /cdrom1 /cdrom2	/vol/dev/aliases/cdrom0 /vol/dev/aliases/cdrom1 /vol/dev/aliases/cdrom2	/vol/dev/rdsk/cntn[dn]/ <i>volume-name</i>
最初、2番目、3番目のJaz (ジャズ) ドライブ	/rmdisk/jaz0 /rmdisk/jaz1 /rmdisk/jaz2	/vol/dev/aliases/jaz0 /vol/dev/aliases/jaz1 /vol/dev/aliases/jaz2	/vol/dev/rdsk/cntndn/ <i>volume-name</i>
最初、2番目、3番目のZipドライブ	/rmdisk/zip0 /rmdisk/zip1 /rmdisk/zip2	/vol/dev/aliases/zip0 /vol/dev/aliases/zip1 /vol/dev/aliases/zip2	/vol/dev/rdsk/cntndn/ <i>volume-name</i>
最初、2番目、3番目のPCMCIAドライブ	/pcmem/pcmem0 /pcmem/pcmem1 /pcmem/pcmem2	/vol/dev/aliases/pcmem0 /vol/dev/aliases/pcmem1 /vol/dev/aliases/pcmem2	/vol/dev/rdsk/cntndn/ <i>volume-name</i>

特定の Solaris コマンドで使用するリムーバブルメディア名を確認する場合は、次の表を使用してください。

Solaris コマンド	デバイス名	使用例
ls, more, vi	/floppy /cdrom /rmdisk/zip0 /rmdisk/jaz0 /pcmem/pcmem0	ls /floppy/myfiles/ more /cdrom/myfiles/filea
fsck, newfs, mkfs	/vol/dev/aliases/floppy0 /vol/dev/rdsk/cntndn	newfs /vol/dev/aliases/floppy0 mkfs -F udfs /vol/dev/rdsk/cntndn

## リムーバブルメディア上のデータにアクセスするためのガイドライン

ほとんどの CD と DVD は、ISO 9660 標準でフォーマットされています。このフォーマットには移植性があるため、ほとんどの CD と DVD をボリューム管理によってマウントできます。ただし、UFS ファイルシステムを持つ CD と DVD にはアーキテクチャ間の移植性がないため、設計されたときのアーキテクチャで使用する必要があります。

たとえば、SPARC システム用にフォーマットされた UFS ファイルシステムを持つ CD と DVD は x86 システムでは認識されません。同様に、x86 システム用にフォーマットされた UFS CD は、ボリューム管理によって SPARC システム上にマウントすることはできません。同じ制限が、フロッピーディスクにもあてはまります。実際には、同じビット構造を共有するアーキテクチャもあります。このため、場合によっては、あるアーキテクチャに固有の UFS フォーマットが別のアーキテクチャによって認識されることもありますが、UFS ファイルシステム構造はこのような互換性を保証するように設計されたものではありません。

さまざまなフォーマットに対応するために、CD または DVD は、実際にはハードディスクのパーティションに似たスライスに分割されます。9660 部分は移植可能ですが、UFS 部分はアーキテクチャに固有です。CD または DVD のマウントで問題が生じた場合、特にそれがインストール用 CD または DVD の場合は、その UFS ファイルシステムが、使用しているシステムのアーキテクチャに適しているかどうかを CD または DVD のラベルで確認してください。

## Jaz ドライブおよび Zip ドライブへのアクセス

次の状況に応じて、Jaz ドライブや Zip ドライブへのアクセス方法が以前の Solaris リリースから変更されたかどうかを判断できます。

- 以前の Solaris 8 6/00 リリースから Solaris 9 リリースへアップグレードする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることができます。
- 新規に Solaris 9 リリースをインストールする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることはできません。

以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスする場合は、次の手順に従ってください。

1. /etc/vold.conf ファイル内の次の行でテキストの初めに # マークを挿入し、コメント扱いにします。

```
# use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

2. システムをリブートします。



## ▼ 新しいリムーバブルメディアドライブを追加する方法

新しいリムーバブルメディアドライブを追加する場合は、`/reconfigure` ファイルを作成し、システムをリブートして、ボリューム管理に新しいメディアドライブを認識させる必要があります。

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

```
# touch /reconfigure
```
  3. システムを実行レベル **0** にします。

```
# init 0
```
  4. システムの電源を切ります。
  5. 新しいリムーバブルメディアドライブを接続します。  
詳細な手順については、該当するハードウェアのマニュアルを参照してください。
  6. システムの電源を入れます。  
システムが自動的にマルチユーザーモードになります。

## ボリューム管理 (vold) の終了と起動

場合によっては、ボリューム管理を使用しないで、メディアを管理した方がよいことがあります。この節では、ボリューム管理をいったん終了し、後で再起動する方法について説明します。

## ▼ ボリューム管理 (vold) を終了させる方法

- 手順
1. メディアが使用中でないことを確認します。  
メディアを使用中のすべてのユーザーを確認できた保証がない場合は、[37 ページの「リムーバブルメディアが使用中かどうかを調べる方法」](#)の手順に従って `fuser` コマンドを使用してください。
  2. スーパーユーザーになります。
  3. `volmgt stop` コマンドを入力します。

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

## ▼ ボリューム管理 (vold) を再起動する方法

- 手順
1. スーパーユーザーになります。
  2. **volmgt start** コマンドを入力します。

```
# /etc/init.d/volmgt start
volume management starting.
```

## ▼ リムーバブルメディア上の情報にアクセスする方法

- 手順
1. メディアを挿入します。  
数秒後にメディアがマウントされます。
  2. ドライブにメディアが入っているかどうかを確認します。

```
% volcheck
コマンド行インタフェースを使用し、適切なデバイス名を指定して情報にアクセス
します。デバイス名については、表 2-1 を参照してください。
```

3. メディアの内容をリスト表示します。

```
% ls /media
```

### 例 2-1 リムーバブルメディア上の情報にアクセスする

次のように入力して、フロッピーディスク上の情報にアクセスします。

```
$ volcheck
$ ls /floppy
myfile
```

次のように入力して、Jaz ドライブ上の情報にアクセスします。

```
$ volcheck
$ ls /rmdisk
jaz0/          jaz1/
```

次のように入力して、CD-ROM 上の情報にアクセスします。

```
$ volcheck
$ ls /cdrom
cdrom0@          solaris_9_sparc
```

次のように入力して、CD-ROM 上のシンボリックリンクを表示します。

```
$ ls -lL /cdrom/cdrom0
total 24
dr-xr-xr-x  2 root      sys          2048 Dec  3 11:54 s0/
```

```
drwxr-xr-x 18 root    root      512 Dec  3 13:09 s1/
drwxr-xr-x  2 root    root      512 Dec  3 13:10 s2/
drwxr-xr-x  2 root    root      512 Dec  3 13:10 s3/
drwxr-xr-x  2 root    root      512 Dec  3 13:10 s4/
drwxr-xr-x  2 root    root      512 Dec  3 13:10 s5/
```

次のように入力して、PCMCIA メモリーカード上の情報にアクセスします。

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

## ▼ リムーバブルメディア上の情報をコピーする方法

リムーバブルメディア上のファイルやディレクトリへのアクセスは、他のファイルシステムの場合とまったく同じように行えます。ただし、所有権とアクセス権については重大な制限があります。

たとえば、あるユーザーが、CD 上のファイルを自分のファイルシステムにコピーした場合、そのユーザーはファイルの所有者になりますが、書き込み権は与えられません (CD 上のファイルには書き込み権がないため)。アクセス権は各自で変更しなければなりません。

手順 1. メディアがマウントされていることを確認します。

```
$ ls /media
```

ls コマンドは、マウントされたメディアの内容を表示します。内容が表示されない場合は、34 ページの「リムーバブルメディア上の情報にアクセスする方法」を参照してください。

2. (省略可能) ファイルまたはディレクトリをコピーします。

たとえば、CD の場合は次のように入力します。

```
$ cp /cdrom/sol_9_1202_sparc/s0/Solaris_9/Tools/add_install_client .
$ ls -l
-rwxr-xr-x  1 pmorph  gelfs    59586 Jan 16  2004 add_install_client*
```

たとえば、PCMCIA メモリーカードの場合は次のように入力します。

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

## ▼ 音楽用 CD/DVD を再生する方法

Solaris リリースを実行しているシステムに接続されたリムーバブルメディアドライブから音楽用のメディアを再生する場合は、xmcd などのパブリックドメインソフトウェアにアクセスする必要があります。xmcd は、次の場所から入手できます。

- <http://www.ibiblio.org/tkan/xmcd>

このサイトには、xmcdd ソフトウェアの更新情報が頻繁に追加され、Sun Blade™ システムなどの Sun 製の新しいハードウェアで再生するための新バージョンの xmcdd も含まれています。

- [http://www.sun.com/software/solaris/freeware/pkggs\\_download.html](http://www.sun.com/software/solaris/freeware/pkggs_download.html)

xmcdd ソフトウェアを CDDA (CD Digital Audio) サポートと併用して音楽用のメディアを再生するときは、次のことに注意してください。

- Sun Blade™ システムではバージョン 3.1 以上の xmcdd を使用してください (このバージョンには CDDA サポート機能があるため)。CDDA は Sun Blade システムで CD の応答待ちをするために必要です。
- xmcdd を起動し、オプションボタン (ハンマーとねじ回しが表示されたボタン) をクリックして、「CDDA playback」をクリックすることで CDDA を使用可能にしてください。
- CDDA が使用可能になると、オーディオはオーディオデバイスに送られるよう指示されるため、ヘッドホンや外部スピーカはメディアドライブそのものではなくオーディオデバイスに接続する必要があります。
- CDDA は他のマシンでも使用可能にできます。Sun Blade システムでメディアを再生するためには、CDDA を使用可能にする必要があります。

次の問題にも考慮してください。

- CD-ROM とオーディオデバイスが内部接続されていないシステムで xmcdd を標準の再生機能と組み合わせて使用する場合は、ヘッドホンを CD-ROM ドライブのヘッドホンポートに挿入する必要があります。
- CD-ROM とオーディオデバイスが内部接続されているシステムで xmcdd を標準の再生機能と組み合わせて使用する場合は、次のどちらの方法も可能です。
  1. ヘッドホンを CD-ROM ドライブのヘッドホンポートに挿入する。
  2. ヘッドホンをオーディオデバイスのヘッドホンポートに挿入する。

ただし、2 番目を選択した場合は、次の作業を行う必要があります。

- 内蔵 CD を入力デバイスとして選択する。
- モニターボリュームが 0 以外の値になっていることを確認する。

これらの作業はどちらも、sdtaudiocontrol のレコードパネルから実行することができます。

xmcdd ソフトウェアをインストールすると、音楽用 CD を CD-ROM ドライブに挿入し、xmcdd のコントロールパネルを起動するだけで再生できます。

- 手順
1. **xmcdd** ソフトウェアをインストールします。
  2. メディアをメディアドライブに挿入します。
  3. **xmcdd** コマンドを起動します。

```
% ./xmcdd &
```

## ▼ リムーバブルメディアが使用中かどうかを調べる方法

手順 1. スーパーユーザーになります。

2. メディアにアクセスしているプロセスを特定します。

```
# fuser -u [-k] /media
```

-u メディアを使用しているプロセスなどを表示する

-k メディアにアクセスしているプロセスを強制終了する

fuser コマンドの使用方法については、fuser(1M) のマニュアルページを参照してください。

3. (省略可能) メディアにアクセスしているプロセスを強制終了します。

```
# fuser -u -k /media
```



---

注意 - メディアにアクセスしているプロセスの強制終了は、緊急の場合にのみ行います。

---

4. プロセスが終了していることを確認します。

```
# pgrep process-ID
```

### 例 2-2 メディアが使用中かどうかを調べる

次の例は、プロセス26230c (所有者 pmorph) が /cdrom/sol\_9\_1202\_sparc/s0/Solaris\_9/Tools/ ディレクトリにアクセスしていることを示しています。

```
# fuser -u /cdrom/sol_9_1202_sparc/s0/Solaris_9/Tools/  
/cdrom/sol_9_1202_sparc/s0/Solaris_9/Tools/: 7258c (pmorph)
```

## ▼ リムーバブルメディアを取り出す方法

手順 1. メディアが使用中でないことを確認します。

シェルまたはアプリケーションがメディア上のファイルまたはディレクトリのいずれかにアクセスしている場合、メディアは「使用中」であることを忘れないでください。CD を使用しているシェルやアプリケーションなどをすべて検出したかどうか分からない (デスクトップツールの背後に隠れているシェルがアクセスしている可能性がある) 場合は、37 ページの「リムーバブルメディアが使用中かどうかを調

べる方法」の手順に従って `fuser` コマンドを使用してください。

2. メディアを取り出します。

```
# eject media
```

たとえば、CD の場合は次のように入力します。

```
# eject cdrom
```

たとえば、PCMCIA メモリーカードの場合は次のように入力します。

```
# eject pcmem0
```

---

## リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)

次の表に、リモートシステム上のリムーバブルメディアへのアクセスに必要な作業を示します。

作業	説明	参照先
1. ローカルのメディアをリモートシステムで使用できるようにする	必要に応じて、リムーバブルメディアドライブをシステムに追加する	38 ページの「ローカルのメディアを他のシステムで使用可能にする方法」
2. リモートシステム上のリムーバブルメディアにアクセスする	メディアをドライブに挿入する	34 ページの「リムーバブルメディア上の情報にアクセスする方法」

### ▼ ローカルのメディアを他のシステムで使用可能にする方法

システムのメディアドライブを共有するようにシステムを設定すると、そのドライブに読み込まれているメディアが他のシステムでも使用できるようになります。(これは音楽用の CD には当てはまりません。) メディアドライブを共有すると、42 ページの「リモートシステム上のリムーバブルメディアにアクセスする方法」の手順に従ってそのドライブをマウントするだけで、ドライブにロードされているメディアを他のシステムでも使用できます。

- 手順
1. スーパーユーザーになります。
  2. NFS デーモン (`nfsd`) が実行されているかどうかを調べます。

```
# ps -ef | grep nfsd
root 14533    1 17 10:46:55 ?        0:00 /usr/lib/nfs/nfsd -a 16
root 14656   289  7 14:06:02 pts/3 0:00 grep nfsd
```

デーモンが実行されている場合、`/usr/lib/nfs/nfsd` の行は、上記のように表示されます。デーモンが実行されていない場合は、`grep nfsd` の行だけが表示されます。

3. `nfsd` の状態を確認し、次のいずれかの手順を選択します。

- a. `nfsd` が実行されている場合は、**手順 8** に進む。
- b. `nfsd` が実行されていない場合は、**手順 4** に進む。

4. `nfsd` が共有するためのダミーディレクトリを作成します。

```
# mkdir /dummy-dir
```

`dummy-dir` マウントポイントは、任意のディレクトリ名にすることができます。たとえば、`dummy` にできます。このディレクトリには、ファイルは含まれません。これは、NFS デーモンを「呼び起こして」、共有されたメディアドライブを認識させることだけを目的としています。

5. 次のエントリを `/etc/dfs/dfstab` ファイルに追加します。

```
share -F nfs -o ro [-d comment] /dummy-dir
```

NFS デーモンを起動すると、このエントリ「呼び起こし」が参照され、共有されたメディアドライブが認識されます。コメント (`-d` が前に付く) はオプションです。

6. NFS デーモンを起動します。

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

7. NFS デーモンが実際に実行されていることを確認します。

```
# ps -ef | grep nfsd
root 14533    1 17 10:46:55 ?        0:00 /usr/lib/nfs/nfsd -a 16
root 14656   289  7 14:06:02 pts/3 0:00 /grep nfsd
```

8. 現在ドライブにあるメディアを取り出します。

```
# eject media
```

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

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

10. 次の行を `/etc/rmmount.conf` ファイルに追加します。

```
# File System Sharing
share media*
```

上記の行によって、システムの CD-ROM ドライブにロードされる CD が共有されます。ただし、`share (1M)` のマニュアルページに説明されているように、特定の CD (複数も可) に共有範囲を限定することができます。

11. `/etc/rmmount.conf` ファイルから書き込み権を削除します。

```
# chmod 444 /etc/rmmount.conf
```

この手順によって、ファイルはそのデフォルトのアクセス権に戻ります。

12. メディアを読み込みます。

ここで読み込むメディアはすべて、他のシステムで使用できるようになります。必ずドライブのランプの点滅が消えるまで待って、この作業を確認するようにしてください。

メディアにアクセスするには、リモートユーザーは、[42 ページの「リモートシステム上のリムーバブルメディアにアクセスする方法」](#)の手順に従って、名前によりそのメディアをマウントする必要があります。

13. `share` コマンドを使用して、メディアが実際に他のシステムで使用できるかどうかを確認します。

メディアが使用可能な場合は、その共有設定が表示されます。共有される `dummy` ディレクトリも表示されます。

```
# share
- /dummy ro "dummy dir to wake up NFS daemon"
- /cdrom/sol_9_1202_sparc/s5 ro ""
- /cdrom/sol_9_1202_sparc/s4 ro ""
- /cdrom/sol_9_1202_sparc/s3 ro ""
- /cdrom/sol_9_1202_sparc/s2 ro ""
- /cdrom/sol_9_1202_sparc/s1 ro ""
- /cdrom/sol_9_1202_sparc/s0 ro ""
```

### 例 2-3 ローカル CD を他のシステムで使用可能にする

次の例は、ローカルの CD をネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```
# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118   1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
  share -F nfs -o ro /dummy
# eject cdrom0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount.conf
(次の行をファイルシステム共有セクションに追加する)
  share cdrom*
# chmod 444 /etc/rmmount.conf
(CD をロードする)
# share
- /dummy ro ""
- /cdrom/sol_9_1202_sparc/s5 ro ""
- /cdrom/sol_9_1202_sparc/s4 ro ""
- /cdrom/sol_9_1202_sparc/s3 ro ""
- /cdrom/sol_9_1202_sparc/s2 ro ""
```



```
- /cdrom/sol_9_1202_sparc/s1 ro ""
- /cdrom/sol_9_1202_sparc/s0 ro ""
```

#### 例 2-4 ローカルのフロッピーディスクを他のシステムで使用可能にする

次の例は、ローカルのフロッピーディスクをネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```
# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118    1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
share -F nfs -o ro /dummy
# eject floppy0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount.conf
(次の行をファイルシステム共有セクションに追加する)
share floppy*
# chmod 444 /etc/rmmount.conf
(フロッピーディスクをロードする)
# volcheck -v
media was found
# share
- /dummy ro ""
- /floppy/myfiles rw ""
```

#### 例 2-5 ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする

次の例は、ローカルの PCMCIA メモリーカードをネットワーク上の他のシステムでも使用できるようにする方法を示しています。

```
# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118    1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
share -F nfs -o ro /dummy
# eject pcmem0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount.conf
(次の行をファイルシステム共有セクションに追加する)
share floppy*
# chmod 444 /etc/rmmount.conf
(PCMCIA メモリーカードをロードする)
# volcheck -v
media was found
# share
- /dummy ro ""
- /pcmem/myfiles rw ""
```

## ▼ リモートシステム上のリムーバブルメディアにアクセスする方法

38 ページの「ローカルのメディアを他のシステムで使用可能にする方法」の手順に従って、リモートシステムがそのシステム上のメディアを共有している場合は、手動でそのメディアをファイルシステムにマウントすれば、リモートシステム上のメディアにアクセスできます。

手順 1. マウントポイントとして使用する既存のディレクトリを指定するか、作成します。

```
$ mkdir directory
```

*directory* は、他のシステムのメディアのマウントポイントとして作成するディレクトリの名前です。

2. マウントするメディアの名前を確認します。

```
$ showmount -e system-name
```

たとえば、次のようになります。

```
export list for starbug:
/dummy (everyone)
/cdrom/sol_9_1202_sparc/s5 (everyone)
/cdrom/sol_9_1202_sparc/s4 (everyone)
/cdrom/sol_9_1202_sparc/s3 (everyone)
/cdrom/sol_9_1202_sparc/s2 (everyone)
/cdrom/sol_9_1202_sparc/s1 (everyone)
/cdrom/sol_9_1202_sparc/s0 (everyone)
```

スーパーユーザー権限で、メディアをマウントします。

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

*system-name*           マウントするメディアを持つシステムの名前

*media-name*           マウントするメディアの名前

*local-mount-point*   リモートのメディアのマウント先のローカルディレクトリ

3. スーパーユーザーをログアウトします。

4. メディアがマウントされていることを確認します。

```
$ ls /media
```

### 例 2-6 他のシステム上の CD にアクセスする

次の例は、sol\_9\_1202\_sparc という CD をリモートシステム starbug からローカルシステムの /mnt ディレクトリにマウントする方法を示しています。

```
$ showmount -e starbug
export list for starbug:
/dummy (everyone)
/cdrom/sol_9_1202_sparc/s5 (everyone)
```

```

/cdrom/sol_9_1202_sparc/s4 (everyone)
/cdrom/sol_9_1202_sparc/s3 (everyone)
/cdrom/sol_9_1202_sparc/s2 (everyone)
/cdrom/sol_9_1202_sparc/s1 (everyone)
/cdrom/sol_9_1202_sparc/s0 (everyone)
$ su
Password: password
# mount -F nfs -o ro starbug:/cdrom/sol_9_1202_sparc/s0 /mnt
# exit
$ ls /mnt
Copyright Solaris_9

```

### 例 2-7 他のシステム上のフロッピーディスクにアクセスする

次の例は、myfiles というフロッピーディスクをリモートシステム mars からローカルシステムの /floppy ディレクトリにマウントする方法を示しています。

```

$ cd /net/mars
$ ls /floppy
floppy0      myfiles
$ su
Password: password
# mount -F nfs mars:/floppy/myfiles /floppy
# exit
$ ls /floppy
myfiles

```

### 例 2-8 他のシステム上の PCMCIA メモリーカードにアクセスする

次の例は、myfiles という PCMCIA メモリーカードをリモートシステム mars からローカルシステムの /pcmem ディレクトリにマウントする方法を示しています。

```

$ cd /net/mars
$ ls /pcmem
pcmem0      myfiles
$ su
Password: password
# mount -F nfs mars:/pcmem/myfiles /pcmem
# exit
$ ls /pcmem
myfiles

```



## 第 3 章

# リムーバブルメディアのフォーマット (手順)

この章では、Solaris 環境でコマンド行からリムーバブルメディアをフォーマットする方法について説明します。

リムーバブルメディアのフォーマットに関する手順については、45 ページの「リムーバブルメディアのフォーマット (作業マップ)」を参照してください。

リムーバブルメディアの概要については、第 1 章を参照してください。

# リムーバブルメディアのフォーマット (作業マップ)

作業	説明	参照先
1. フォーマットされていないメディアのロード	メディアをドライブに挿入して、 <code>volcheck</code> コマンドを実行する	49 ページの「リムーバブルメディアを読み込む方法」
2. メディアのフォーマット	リムーバブルメディアをフォーマットする	50 ページの「リムーバブルメディアをフォーマットする方法 ( <code>rmformat</code> )」
3. (省略可能) UFS ファイルシステムの追加	ファイル転送にフロッピーディスクを使用する場合は、UFS ファイルシステムを追加する	51 ページの「ファイルシステムの追加用にリムーバブルメディアをフォーマットする方法」
4. (省略可能) メディアの検査	メディア上のファイルシステムの整合性を検査する	53 ページの「リムーバブルメディア上のファイルシステムを検査する方法」

作業	説明	参照先
5. (省略可能) メディア上の不良ブロックの修復	必要に応じて、メディア上に不良ブロックがあれば修復する	54 ページの「リムーバブルメディア上の不良ブロックを修復する方法」
6. (省略可能) 読み取り/書き込み保護とパスワードによる保護の適用	必要に応じて、読み取り/書き込み保護やパスワードによる保護をメディアに適用する	54 ページの「リムーバブルメディアの書き込み保護を有効または無効にする方法」

## リムーバブルメディアのフォーマットの概要

rmformat コマンドは、スーパーユーザー以外で使用できるユーティリティであり、書き込み可能なリムーバブルメディアのフォーマットや保護に使用できます。

rmformat コマンドには、次の3つのフォーマットオプションがあります。

- quick — このオプションは、トラックを検証せずに、あるいは、検証するトラックを制限して、リムーバブルメディアをフォーマットします。
- long — このオプションは、リムーバブルメディアを完全にフォーマットします。このオプションを使用してドライブ自身によるメディア全体の検証を行うデバイスもあります。
- force — このオプションは、ユーザーへの確認なしに、リムーバブルメディアを完全にフォーマットします。パスワードによる保護機能を備えたメディアでは、このオプションはフォーマットを行う前にパスワードをクリアします。この機能はパスワードを忘れてしまったときに便利です。パスワードによる保護機能を備えていないメディアでは、long オプションのフォーマットが行われます。

## リムーバブルメディアのフォーマットのガイドライン

リムーバブルメディアをフォーマットするときは、次のことに注意してください。

- ファイルマネージャのウィンドウを閉じて、ファイルマネージャを終了してください。

ファイルマネージャは、フォーマットされていないメディアを挿入すると、フォーマットウィンドウを自動的に表示します。このウィンドウが表示されないようにするために、ファイルマネージャを終了してください。ファイルマネージャを開いたままにしておきたい場合は、フォーマットウィンドウが表示されてから、そのフォーマットウィンドウを終了してください。

- ボリュームマネージャ (vold) はファイルシステムを自動的にマウントするため、メディアに既存のファイルシステムが含まれている場合は、メディアをフォーマットする前にマウント解除する必要があります。

## リムーバブルメディアのハードウェア面での考慮事項

この節では、リムーバブルメディアのハードウェアの面で考慮すべき事項について説明します。

### フロッピーディスクのハードウェア面での考慮事項

フロッピーディスクをフォーマットするときは、次のことに注意してください。

- フロッピーディスク名について、表 2-1 を参照してください。
- 名前の付いていない (「ラベル」がない) フロッピーディスクには、noname というデフォルト名が割り当てられます。

Solaris システムは、Solaris システム用と DOS システム用にフロッピーディスクをフォーマットできます。ただし、ハードウェアプラットフォームによっていくつかの制限があります。次の表に、これらの制限がまとめてあります。

プラットフォームの種類	フロッピーディスクのフォーマット仕様
SPARC システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
	UDFS 用
x86 システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
	UDFS 用

UFS 用にフォーマットされたフロッピーディスクは、それらがフォーマットされたハードウェアプラットフォームに制限されます。つまり、SPARC システムでフォーマットされた UFS フロッピーディスクは、x86 システム上の UFS には使用できません。同様に、x86 システム上でフォーマットされたフロッピーディスクは SPARC システム上の UFS には使用できません。これは、SPARC と x86 とでは UFS フォーマットが異なるためです。SPARC はリトルエンディアンによるビットコーディング、x86 はビッグエンディアンによるビットコーディングを採用しています。

SunOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、SunOS ファイルシステムをサポートするための構造からなります。DOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、MS-DOS または NEC-DOS のどちらか

のファイルシステムをサポートする構造からなります。フロッピーディスクを準備するために必要な手順は、ファイルシステムによって異なります。したがって、フロッピーディスクをフォーマットする前には、どの作業が必要かを決めてください。詳細については、45 ページの「リムーバブルメディアのフォーマット (作業マップ)」を参照してください。

Solaris システム (SPARC または x86 のどちらか) では、次の密度でフロッピーディスクをフォーマットできます。

フロッピーディスクのサイズ	フロッピーディスクの密度	容量
3.5 インチ	高密度 (HD)	1.44M バイト
3.5 インチ	倍密度 (DD)	720K バイト

デフォルトで、フロッピーディスクドライブは、それに近い密度にフロッピーディスクをフォーマットします。つまり、デフォルトでは、1.44M バイトのドライブは、フロッピーディスクが実際に 1.44M バイトのフロッピーディスクかどうかに関係なく、特に指示しないかぎり、そのフロッピーディスクを 1.44M バイト用にフォーマットしようとしています。言い換えれば、フロッピーディスクもドライブも、その容量以下にフォーマットすることは可能です。

## PCMCIA メモリーカードのハードウェア面での考慮事項

Solaris システムは、Solaris システム用と DOS システム用に PCMCIA メモリーカードをフォーマットできます。ただし、ハードウェアプラットフォームによっていくつかの制限があります。次の表に、これらの制限がまとめてあります。

プラットフォームの種類	PCMCIA メモリーカードのフォーマット仕様
SPARC システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用
x86 システム	UFS 用
	MS-DOS または NEC-DOS (PCFS) 用

UFS 用にフォーマットされた PCMCIA メモリーカードは、それらがフォーマットされたハードウェアプラットフォームに制限されます。つまり、SPARC システムでフォーマットされた UFS PCMCIA メモリーカードは、x86 システム上の UFS には使用できません。同様に、x86 システムでフォーマットされた PCMCIA メモリーカードは SPARC システムでは使用できません。これは、SPARC と x86 とでは UFS フォーマットが異なるためです。

UFS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、UFS ファイルシステムをサポートする構造からなります。DOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、MS-DOS または NEC-DOS のどちらかのファイルシ



システムをサポートする構造からなります。PCMCIA メモリーカードを準備するために必要な手順はファイルシステムによって異なります。したがって、PCMCIA メモリーカードをフォーマットする前には、どの作業が必要かを決めてください。

## ▼ リムーバブルメディアを読み込む方法

手順 1. メディアを挿入します。

2. メディアがフォーマットされていることを確認します。

フォーマットされているかどうか分からない場合は、メディアを挿入し、[手順 3](#)の説明に従って、コンソールの状態メッセージを確認してください。メディアをフォーマットする必要がある場合は、[50 ページの「リムーバブルメディアをフォーマットする方法 \(rmformat\)」](#)を参照してください。

3. ボリューム管理に通知します。

```
$ volcheck -v media was found
```

次の 2 つの状態メッセージのいずれかが表示されます。

```
media was found
```

ボリューム管理がメディアを検出し、[表 2-1](#) に記述されたディレクトリにそのメディアをマウントしようとする。

メディアが正しくフォーマットされている場合は、コンソールにエラーメッセージが表示されない。

メディアがフォーマットされていない場合でも「media was found」メッセージは表示されるが、次のようなエラーメッセージがコンソールに表示される。

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

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

```
fd0: bad format
```

メディアをフォーマットしてからでないと、ボリューム管理はそれをマウントできない。詳細は、[第 3 章](#)を参照。

```
no media was found
```

ボリューム管理は、メディアを検出できなかった。メディアが正しく挿入されていることを確認して、`volcheck` をもう一度実行する。うまくいかない場合は、メディアをチェックする。損傷の可能性がある。メディアを手動でマウントしてみることもできる。

4. メディアの内容を表示して、メディアがマウントされていることを確認します。  
たとえば、フロッピーディスクの場合は次のように入力します。

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

前に説明したように、floppy0 はフロッピーディスクの実際の名前へのシンボリックリンクです。この場合、myfiles が実際の名前です。正しくフォーマットされていて、名前がない場合は、unnamed\_floppy と呼ばれます。

/floppy ディレクトリの下に何も表示されない場合は、フロッピーディスクがマウントされていないか、正しくフォーマットされていないかのどちらかです。これを調べるには、mount コマンドを実行して、次のような /floppy で始まる行を探してください (通常は、リストの最後にあります)。

```
/floppy/name on /vol/dev/diskette0/name
```

このような行が表示されない場合、フロッピーディスクはマウントされていません。コンソールウィンドウにエラーメッセージが表示されていないかどうか確認してください。

## ▼ リムーバブルメディアをフォーマットする方法 (rmformat)

rmformat コマンドを使用すると、メディアをフォーマットすることができます。このコマンドは、デフォルトで、メディア上にパーティション 0 とパーティション 2 (メディア全体) という 2 つのパーティションを作成します。

- 手順 1. ボリュームマネージャが動作していることを確認します。動作していれば、デバイス名のニックネームを使用できます。

```
$ ps -ef | grep vold
root 212 1 0 Nov 03 ? 0:01 /usr/sbin/vold
```

vold の起動方法については、34 ページの「ボリューム管理 (vold) を再起動する方法」を参照してください。メディアのデバイス名の確認方法については、30 ページの「リムーバブルメディア名の使用」を参照してください。

2. リムーバブルメディアをフォーマットします。

```
$ rmformat -F [ quick | long | force ] device-name
```

rmformat のフォーマットオプションについては、前出の節を参照してください。

rmformat コマンドの出力によって不良ブロックが見つかった場合は、54 ページの「リムーバブルメディア上の不良ブロックを修復する方法」の指示に従ってブロックを修復してください。

3. (省略可能) リムーバブルメディアに、Solaris 環境で使用する 8 文字のラベルを付けます。

```
$ rmformat -b label device-name
```

DOS ラベルの作成方法については、mkfs\_pcfs(1M) のマニュアルページを参照してください。

### 例 3-1 リムーバブルメディアをフォーマットする

次の例は、フロッピーディスクのフォーマット方法を示しています。

```
$ rmformat -F quick /dev/rdiskette
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

.....

次の例は、Zip ドライブのフォーマット方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/zip0
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

.....

## ▼ ファイルシステムの追加用にリムーバブルメディアをフォーマットする方法

手順 1. リムーバブルメディアをフォーマットします。

```
$ rmformat -F quick device-name
```

2. (省略可能) 代替の Solaris パーティションテーブルを作成します。

```
$ rmformat -s slice-file device-name
```

スライスファイルの例は次のようになります。

```
slices: 0 = 0, 30MB, "wm", "home" :
        1 = 30MB, 51MB :
        2 = 0, 94MB, "wm", "backup" :
        6 = 81MB, 13MB
```

3. スーパーユーザーになります。

4. 適切なファイルシステムの種類を決定し、次のいずれかの作業を選択します。

a. UFS ファイルシステムを作成する

```
# newfs device-name
```

b. UDFS ファイルシステムを作成する

```
# mkfs -F udfs device-name
```

### 例 3-2 UFS ファイルシステム用にフロッピーディスクをフォーマットする

次の例は、フロッピーディスクをフォーマットし、そのフロッピーディスク上に UFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/floppy0
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
$ su
# /usr/sbin/newfs /vol/dev/aliases/floppy0
newfs: construct a new file system /dev/rdiskette: (y/n)? y
/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors
      1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
      32, 640, 1184, 1792, 2336,
#
```

### 例 3-3 UFS ファイルシステム用に PCMCIA メモリーカードをフォーマットする

次の例は、PCMCIA メモリーカードをフォーマットし、そのカード上に UFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /vol/dev/aliases/pcm0
$ su
# /usr/sbin/newfs -v /vol/dev/aliases/pcm0
newfs: construct a new file system /vol/dev/aliases/pcm0: (y/n)? y
.
.
.
#
```

### 例 3-4 PCFS ファイルシステム用にリムーバブルメディアをフォーマットする

次の例は、代替 fdisk パーティションを作成する方法を示しています。

```
$ rmformat -F quick /dev/rdisk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
$ su
# fdisk /dev/rdisk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdisk/c0t4d0s2:c: (y/n)? y
#
```

次の例は、fdisk パーティションを作成せずに、PCFS ファイルシステムを作成する方法を示しています。

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
```

```
$ su
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
#
```

## ▼ リムーバブルメディア上のファイルシステムを検査する方法

- 手順
1. スーパーユーザーになります。
  2. ネームサービスを確認し、次のいずれかの作業を選択します。
    - a. UFS ファイルシステムを検査する  

```
# fsck -F ufs device-name
```
    - b. UDFS ファイルシステムを検査する  

```
# fsck -F udfs device-name
```
    - c. PCFS ファイルシステムを検査する  

```
# fsck -F pcfs device-name
```

### 例 3-5 リムーバブルメディア上の PCFS ファイルシステムを検査する

次の例は、メディア上の PCFS ファイルシステムの整合性を検査する方法を示しています。

```
# fsck -F pcfs /dev/rdisk/c0t4d0s2
** /dev/rdisk/c0t4d0s2
** ファイルシステムのメタデータを走査する
** メタデータの不一致があれば訂正する
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

## ▼ リムーバブルメディア上の不良ブロックを修復する方法

ドライブが不良ブロック管理をサポートしている場合にのみ、検証中に見つかった不良セクタを `rmformat` コマンドで検証、解析、および修復できます。ほとんどのフロッピーディスクや PCMCIA メモリカードは不良ブロック管理をサポートしていません。

ドライブが不良ブロック管理をサポートしている場合、不良ブロックを修復するための最大の努力が行われます。それでも不良ブロックを修復できなかった場合、修復に失敗したことを示すメッセージが表示されます。

- 手順 1. リムーバブルメディア上の不良ブロックを修復します。

```
$ rmformat -c block-numbers device-name
```

`block-numbers` には、前の `rmformat` セッションで獲得したブロック番号を 10 進数、8 進数、または 16 進数形式で指定します。

2. リムーバブルメディアを検証します。

```
$ rmformat -V read device-name
```

## リムーバブルメディアに読み取り/書き込み保護とパスワードによる保護を適用する

Iomega メディア (Zip ドライブや Jaz ドライブなど) には、読み取り保護または書き込み保護を適用し、パスワードを設定することができます。

## ▼ リムーバブルメディアの書き込み保護を有効または無効にする方法

- 手順 1. 書き込み保護を有効にするか無効にするかを決定し、次のいずれかの作業を選択します。

- a. 書き込み保護を有効にする

```
$ rmformat -w enable device-name
```

- b. 書き込み保護を無効にする

```
$ rmformat -w disable device-name
```

2. リムーバブルメディアの書き込み保護が有効または無効になっていることを確認します。

```
$ rmformat -p device-name
```

## ▼ Iomega メディアの読み取り/書き込み保護とパスワードを有効または無効にする方法

パスワードによる保護機能をサポートしている Iomega メディアには、最大 32 文字のパスワードを適用できます。Iomega メディアに対して読み取り保護または書き込み保護を設定するときは、必ずパスワードを使用する必要があります。このとき、パスワードを提供するように促すプロンプトが表示されます。

パスワード機能をサポートしていないリムーバブルメディア上でパスワードを適用しようとする、警告メッセージが表示されます。

- 手順 1. 読み取り保護または書き込み保護とパスワード保護を有効または無効のどちらにするかを決定します。

- a. 読み取り保護または書き込み保護を有効にする

```
$ rmformat -W enable device-name
Please enter password (最大 32 文字): xxx
Please reenter password:
```

```
$ rmformat -R enable device-name
Please enter password (最大 32 文字): xxx
Please reenter password:
```

- b. 読み取り保護または書き込み保護を無効にし、パスワードを削除する

```
$ rmformat -W disable device-name
Please enter password (最大 32 文字): xxx
```

```
$ rmformat -R disable device-name
Please enter password (最大 32 文字): xxx
```

2. リムーバブルメディアの読み取り保護または書き込み保護が有効または無効になっていることを確認します。

```
$ rmformat -p device-name
```

### 例 3-6 読み取り/書き込み保護を有効または無効にする

次の例は、Zip ドライブに対して書き込み保護を有効にし、パスワードを設定する方法を示しています。

```
$ rmformat -W enable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
Please reenter password: xxx
```

次の例は、Zip ドライブに対して書き込み保護を無効にし、パスワードを削除する方法を示しています。

```
$ rmformat -W disable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
```

次の例は、Zip ドライブに対して読み取り保護を有効にし、パスワードを設定する方法を示しています。

```
rmformat -R enable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
Please reenter password: xxx
```

次の例は、Zip ドライブに対して読み取り保護を無効にし、パスワードを削除する方法を示しています。

```
$ rmformat -R disable /vol/dev/aliases/zip0
Please enter password (最大 32 文字): xxx
```



## 第 4 章

---

# CD への書き込み (手順)

---

この章では、`cdrw` コマンドを使用して、データ CD やオーディオ CD を作成およびコピーする手順について説明します。

- 60 ページの「RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法」
- 61 ページの「CD デバイスを確認する方法」
- 61 ページの「CD メディアをチェックする方法」
- 62 ページの「データ CD 用の ISO 9660 ファイルシステムを作成する方法」
- 63 ページの「マルチセッションのデータ CD を作成する方法」
- 65 ページの「オーディオ CD を作成する方法」
- 66 ページの「オーディオトラックを CD から抽出する方法」
- 67 ページの「CD をコピーする方法」
- 68 ページの「CD-RW メディアを消去する方法」

---

## オーディオ CD やデータ CD の取り扱い

`cdrw` コマンドを使用すると、Rock Ridge 拡張または Joliet 拡張を備えた ISO 9660 フォーマットで、CD-R や CD-RW のメディアデバイス上に CD ファイルシステムを書き込むことができます。

`cdrw` コマンドを使用すると、次の作業を行えます。

- データ CD の作成
- オーディオ CD の作成
- オーディオ CD からのオーディオデータの抽出
- CD のコピー
- CD-RW メディアの消去

`cdrw` コマンドは、次のリリースで利用可能です。

- Solaris 8 Operating Environment 1/01 の Software Supplement の CD

- Solaris 9 およびそれ以降のリリース (Solaris 環境の一部として組み込まれている)

推奨される CD-R または CD-RW デバイスについて

は、[http://www.sun.com/io\\_technologies/ihvindex.html](http://www.sun.com/io_technologies/ihvindex.html) をご覧ください。

## CD メディアに関するよく使われる用語

CD メディアについて言及するときによく使われる用語は、次のとおりです。

用語	説明
CD-R	CD 読み取りメディア。1 度だけ書き込みができ、その後は読み取り専用となる。
CD-RW	書き換え可能な CD メディア。書き込みと消去が可能。CD-RW デバイスだけが CD-RW メディアを読み取れる。
ISO 9660	ISO (Industry Standards Organization の略)。コンピュータの標準記憶フォーマットを設定する組織。  ISO 9660 ファイルシステムは、標準の CD-ROM ファイルシステムであり、主だったコンピュータプラットフォームで同じ CD-ROM を読み取れる。この標準は 1988 年に発行され、ハイシエラ (ネバダ州のハイシエラホテルにちなんで名づけられた) という業界団体によって作成された。CD-ROM ドライブを備えたほぼすべてのコンピュータが ISO 9660 ファイルシステムからファイルを読み取れる。
Joliet 拡張	Windows™ ファイルシステム情報を追加する。
Rock Ridge 拡張	UNIX™ ファイルシステム情報を追加する。(Rock Ridge は映画「Blazing Saddles」に出てくる町にちなんで名づけられた。)  注 - これらの拡張は排他的ではない。したがって、mkisofs コマンドに -R と -j の両オプションを指定して両方のシステムとの互換性を確保できる。(詳細については mkisofs (1M) のマニュアルページを参照。)
MMC 準拠のレコード	Multi Media Command の略。これらのレコーダが共通のコマンドセットに準拠していること意味する。ある MMC 準拠レコーダに書き込めるプログラムは、他のすべての MMC 準拠レコーダにも書き込むことができる。

用語	説明
Red Book CDDA	Compact Disc Digital Audio の略。デジタルオーディオをコンパクトディスクに格納するための業界標準方式。「Red Book」形式とも呼ばれる。この公式の業界仕様では、1 つまたは複数のオーディオファイルが 44.1 kHz のサンプリングレートで 16 ビットのステレオサウンドにサンプリングされることが要求される。

CD メディアを扱うときによく使われる用語は、次のとおりです。

用語	説明
ブランキング	CD-RW メディアからデータを消去する処理
mkisofs	CD に書き込むための ISO ファイルシステムを生成するためのコマンド
セッション	リードイン/リードアウト情報を持つ完全なトラック
トラック	完全なデータまたはオーディオの単位

## データ CD やオーディオ CD への書き込み

CD への書き込み処理は、途中で中断されることなく、データストリームが一定に保たれている必要があります。cdrw -s オプションを使用してメディアへの書き込みをシミュレートし、システムが CD への書き込みに十分適した速度でデータを送れるかどうかを確認してみてください。

次の場合には、書き込みエラーが発生することがあります。

- メディアがドライブの速度に対応できない場合。たとえば、メディアの中には 2x または 4x の速度しか保証されていないものもあります。
- システムが書き込み処理に支障をきたすほど多数の大きなプロセスを実行している場合
- リモートシステム上のイメージがある場合、ネットワークの混雑によってイメージ読み取りに遅延が生じる可能性がある場合
- CD 間でコピーするときに、送り側ドライブの速度が受け側ドライブよりも遅い可能性がある場合

上記の問題が発生した場合は、cdrw -p オプションを使用して、デバイスの書き込み速度を遅くしてください。

たとえば、4x の速度での書き込みをシミュレートします。

```
$ cdrw -iS -p 4 image.iso
```

また、`cdrw -C` オプションを使用すると、80 分の CD をコピーするのに、規定のメディア容量が使われます。このオプションを指定しないと、`cdrw` コマンドは、オーディオ CD のコピーに 74 分というデフォルト値を使用します。

詳細については、`cdrw`(1) のマニュアルページを参照してください。

## RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する

デフォルトでは、すべてのユーザーが Solaris 9 リリースで起動するリムーバブルメディアにアクセスできます。ただし、役割によるアクセス制御 (RBAC) で役割を設定して、リムーバブルメディアへのユーザーアクセスを制限することができます。リムーバブルメディアへのアクセスを制限するには、限られたユーザーに役割を割り当てます。

役割の使用方法については、『Solaris のシステム管理 (セキュリティサービス)』の第 5 章「役割によるアクセス制御 (概要)」を参照してください。

### ▼ RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **Solaris** 管理コンソールを起動します。

```
$ /usr/sadm/bin/smc &
```

コンソールの起動方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

3. デバイス管理の権利が含まれている役割を設定します。

詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 6 章「役割によるアクセス制御 (手順)」を参照してください。

4. `cdrw` コマンドを使用する必要があるユーザーを、新しく作成した役割に追加します。

5. `/etc/security/policy.conf` ファイルの次の行をコメントにします。

```
AUTHS_GRANTED=solaris.device.cdrw
```

この手順を実行しないと、デバイス管理役割のメンバーだけでなく、すべてのユーザーが引き続き `cdrw` コマンドを利用できます。

このファイルを変更した後は、デバイス管理役割のメンバーだけが `cdrw` コマンドを使用できるようになります。メンバー以外のユーザーがこのコマンドを使おうとすると、アクセスが拒否され、次のメッセージが表示されます。

```
Authorization failed, Cannot access disks.
```

## CD デバイスを確認する方法

システム上の CD デバイスを確認する場合は、`cdrw -l` コマンドを使用します。

```
$ cdrw -l
Looking for CD devices...
  Node          | Connected Device          | Device type
-----+-----+-----
 cdrom0        | YAMAHA CRW8424S          | 1.0d | CD Reader/Writer
```

特定の CD デバイスを使用する場合は、`-d` オプションを指定します。たとえば、次のようになります。

```
$ cdrw -a filename.wav -d cdrom2
```

メディアが空であるか、または既存の目次があるかどうかを確認する場合は、`cdrw -M` コマンドを使用します。

```
$ cdrw -M

Device : YAMAHA CRW8424S
Firmware : Rev. 1.0d (06/10/99)
Media is blank
%
```

## ▼ CD メディアをチェックする方法

`cdrw` コマンドは、`vold` を使用している状態でも使用していない状態でも機能します。ただし、`vold` デーモンを終了/起動する場合は、スーパーユーザーまたは役割のアクセス権が必要となります。

- 手順
1. **CD** を **CD-RW** デバイスに挿入します。  
CD は、デバイスが読み取ることができるものであれば、どんな種類の CD でもかまいません。
  2. デバイスのリストを表示して、その **CD-RW** ドライブが正しく接続されていることを確認します。

```
$ cdrw -l
Looking for CD devices...
  Node          | Connected Device          | Device type
-----+-----+-----
```

3. (省略可能) そのドライブがリスト内がない場合、システムがそのデバイスを認識するように再構成ブートを行う必要があることがあります。

```
# touch /reconfigure
# init 6
```

あるいは、次のコマンドを実行して、システムをリブートしないで CD-RW デバイスを追加できます。

```
# drvconfig
# disks
```

次に、`vold` を再起動します。

```
# /etc/init.d/vold stop
# /etc/init.d/vold start
```

## データ CD を作成する

まず `mkisofs` コマンドを使用してファイルとファイル情報を CD で使用されるハイシエラ形式に変換し、データを準備します。

### ▼ データ CD 用の ISO 9660 ファイルシステムを作成する方法

手順 1. 空の CD を CD-RW デバイ스에挿入します。

2. 新しい CD で ISO 9660 ファイルシステムを作成します。

```
$ mkisofs -r /pathname > cd-file-system
```

`-r` Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする

`/pathname` ISO 9660 ファイルシステムの作成に使われるパス名を指定する

`> cd-file-system` CD に書き込むファイルシステムの名前を指定する

3. CD のファイルシステムを CD にコピーします。

```
$ cdrw -i cd-file-system
```

`-i cd-file-system` データ CD を作成するためのイメージファイルを示す

#### 例 4-1 データ CD 用の ISO 9660 ファイルシステムを作成する

次の例は、データ CD 用の ISO 9660 ファイルシステムを作成する方法を示しています。

```
$ mkisofs -r /home/dubs/ufs_dir > ufs_cd
Total extents actually written = 56
Total translation table size: 0
Total rockridge attributes bytes: 329
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 8000
56 extents written (0 Mb)
```

次に、CD のファイルシステムを CD にコピーします。たとえば、次のようになります。

```
$ cdrw -i ufs_cd
Initializing device...done.
Writing track 1...done.
Finalizing (数分かかる)...done.
```

## ▼ マルチセッションのデータ CD を作成する方法

この手順では、複数のセッションを CD に書き込む方法について説明します。また、infoA と infoB の各ディレクトリを CD にコピーする例も示します。

手順 1. 最初の CD セッション用のファイルシステムを作成します。

```
$ mkisofs -o infoA -r -V my_infoA /data/infoA
Total translation table size: 0
Total rockridge attributes bytes: 24507
Total directory bytes: 34816
Path table size(bytes): 98
Max brk space used 2e000
8929 extents written (17 Mb)
```

-o infoA      ISO ファイルシステムの名前 (この場合は *infoA*) を指定する  
-r            Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする  
-V my\_infoA   vold がマウントポイントとして使用するボリュームラベル (この場合は *my\_infoA*) を指定する  
/data/infoA   作成する ISO イメージディレクトリを指定する

手順 2. 最初のセッションの ISO ファイルシステムを CD にコピーします。

```
$ cdrw -iO infoA
Initializing device...done.
Writing track 1...done.
done.
Finalizing (数分かかる)...done.
```

-i infoA      CD に書き込むイメージファイルの名前 (*infoA*) を指定する

-o 書き込むために CD を開いたままにしておく

3. CD が排出された後で、再度挿入します。

4. 次の書き込みセッションに含める CD メディアのパス名を確認します。

```
$ eject -n
.
.
.
cdrom0 -> /vol/dev/rdisk/c2t4d0/my_infoA
/vol/dev/... パス名を書き留めてください。
```

5. 次のセッションを書き込む CD 上の次に書き込み可能なアドレスを確認します。

```
% cdrw -M /cdrom
Device : YAMAHA CRW8424S
Firmware : Rev. 1.0d (06/10/99)
```

Track No.	Type	Start address
1	Audio	0
2	Audio	33057
3	Data	60887
4	Data	68087
5	Data	75287
Leadout	Data	84218

Last session start address: 75287

Next writable address: 91118

Next writable address: 出力に記述されているアドレスを書き留めて、次のセッションの書き込み時にこのアドレスを使用できるようにします。

6. 次の CD セッション用の ISO ファイルシステムを作成し、CD に書き込みます。

```
$ mkisofs -o infoB -r -C 0,91118 -M /vol/dev/rdisk/c2t4d0/my_infoA
/data/infoB
Total translation table size: 0
Total rockridge attributes bytes: 16602
Total directory bytes: 22528
Path table size(bytes): 86
Max brk space used 20000
97196 extents written (189 Mb)
```

-o infoB

ISO ファイルシステムの名前 (この場合は infoA) を指定する

-r

Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットする

-C 0,91118

最初のセッションの開始アドレスと、次に書き込み可能なアドレスを指定する



```
-M /vol/dev/rdisk/c2t4d0/my_infoA マージする既存の ISO イメージのパスを示す
/data/infoB 作成する ISO イメージディレクトリを指定する
```

## オーディオ CD を作成する

`cdrw` コマンドを使用すると、個々のオーディオトラックまたは `.au` と `.wav` ファイルからオーディオ CD を作成できます。

次のオーディオ形式がサポートされています。

形式	説明
<code>sun</code>	Red Book CDDA 形式のデータが入る <code>Sun .au</code> ファイル
<code>wav</code>	Red Book CDDA 形式のデータが入る RIFF ( <code>.wav</code> ) ファイル
<code>cda</code>	raw CD オーディオデータ (「リトルエンディアン」バイト順序により、44.1 kHz のサンプリングレートでサンプリングされた 16 ビットの PCM ステレオ) が入る <code>.cda</code> ファイル
<code>aur</code>	<code>.aur</code> ファイル。「ビッグエンディアン」バイト順序による raw CD データが入る

オーディオ形式を指定しなかった場合、`cdrw` コマンドはファイル拡張子に基づいてオーディオファイルの形式を判断しようとします。ファイル拡張子の太文字と小文字は区別されません。

### ▼ オーディオ CD を作成する方法

この手順では、オーディオファイルを CD にコピーする方法について説明します。

- 手順
1. 空の **CD** を **CD-RW** デバイスに挿入します。
  2. オーディオファイルが入っているディレクトリに移動します。
 

```
$ cd /myaudiodir
```
  3. オーディオファイルを **CD** にコピーします。
 

```
$ cdrw -a track1.wav track2.wav track3.wav
```

`-a` オプションによってオーディオ CD が作成されます。

## 例 4-2 オーディオ CD を作成する

次の例は、オーディオ CD を作成する方法を示しています。

```
$ cdwr -a bark.wav chirp.au meow.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (数分かかる)...done.
```

次の例は、マルチセッションのオーディオ CD を作成する方法を示しています。最初のセッションの書き込みが終わると、CD が排出されます。次の書き込みセッションの前に CD を再度挿入します。

```
$ cdwr -a0 groucho.wav chico.au harpo.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (数分かかる)...done.
<CD を再挿入する>
$ cdwr -a zeppo.au
Initializing device...done.
Writing track 1...done.
done.
Finalizing (数分かかる)...done.
```

## ▼ オーディオトラックを CD から抽出する方法

オーディオトラックを CD から抽出して、新しい CD にコピーする場合は、次の手順に従います。

`cdwr -T` オプションを使ってオーディオファイル形式を指定しなかった場合、`cdwr` コマンドはファイル名拡張子を使ってオーディオファイル形式を判断します。たとえば、`cdwr` コマンドは、このファイルが `.wav` ファイルであることを検知します。

```
$ cdwr -x 1 testme.wav
```

- 手順
1. オーディオ CD を CD-RW デバイスに挿入します。
  2. オーディオトラックを抽出します。

```
$ cdwr -x -T audio-type 1 audio-file
```

-x                   オーディオ CD からオーディオデータを抽出する

-T *audio-type*       抽出されるオーディオファイルの形式を指定する。サポートされているオーディオ形式は、sun、wav、cda、aur

- 抽出したトラックを新しい CD にコピーします。

```
$ cdrw -a audio-file
```

#### 例 4-3 オーディオトラックを CD から抽出してオーディオ CD を作成する方法

次の例は、オーディオ CD から最初のトラックを抽出し、そのファイルに `song1.wav` という名前を付ける方法を示しています。

```
$ cdrw -x -T wav 1 song1.wav
Extracting audio from track 1...done.
```

次の例は、オーディオ CD にトラックをコピーする方法を示しています。

```
$ cdrw -a song1.wav
Initializing device...done.
Writing track 1...done.
Finalizing (数分かかる)...done.
```

## ▼ CD をコピーする方法

この手順では、まずすべてのトラックをオーディオ CD から抽出してディレクトリに入れ、次にそれらのトラックをすべて空の CD にコピーする方法について説明します。

---

注 - デフォルトでは、`cdrw` コマンドは CD を `/tmp` ディレクトリにコピーします。コピーを行うには、最高 700M バイトの空き領域が必要です。CD をコピーするのに必要な空き領域が `/tmp` ディレクトリに不足している場合は、`-m` オプションを使って代替ディレクトリを指定します。

---

- 手順
- オーディオ CD を CD-RW デバイスに挿入します。
  - オーディオ CD からトラックを抽出します。

```
$ mkdir music_dir
$ cdrw -c -m music_dir
```

トラックごとに `Extracting audio ...` メッセージが表示されます。  
すべてのトラックが抽出されると、CD が排出されます。

- 空の CD を挿入して、**Return** キーを押します。  
トラックの抽出が終わると、オーディオ CD が排出され、空の CD を挿入するよう指示するプロンプトが表示されます。

#### 例 4-4 CD をコピーする

次の例は、CD 間でコピーする方法を示しています。この作業を行うには、CD-RW デバイスが 2 台必要です。

```
$ cdrw -c -s cdrom0 -d cdrom1
```

### ▼ CD-RW メディアを消去する方法

CD を書き換える前に、既存の CD-RW データを消去する必要があります。

手順 ● 次のいずれかの手順を選択して、メディア全体を消去するか、CD 上の最後のセッションだけを消去します。

a. 最後のセッションだけを消去する

```
$ cdrw -d cdrom0 -b session
```

-b session オプションを使って最後のセッションだけを消去する場合は、-b all オプションを使ってメディア全体を消去する場合に比べて短い時間で済みます。cdrw コマンドを使用して、1 セッションだけでデータ CD またはオーディオ CD を作成した場合でも、-b session オプションを使用できません。

b. メディア全体を消去する

```
$ cdrw -d cdrom0 -b all
```

## 第 5 章

---

# デバイスの管理 (手順)

---

この章では、Solaris 環境のディスク、CD-ROM、テープデバイスなどの周辺デバイスを管理する方法について説明します。

この章の内容は次のとおりです。

- 70 ページの「デバイス管理作業についての参照先」
- 70 ページの「デバイスドライバについて」
- 71 ページの「デバイスの自動構成」
- 73 ページの「デバイス構成情報の表示」

この章で説明する手順は次のとおりです。

- 74 ページの「システム構成情報を表示する方法」
- 76 ページの「デバイス情報を表示する方法」
- 78 ページの「デバイスドライバを追加する方法」
- 77 ページの「周辺デバイスを追加する方法」

周辺デバイスへのアクセスについては、第 9 章を参照してください。

Solaris 環境のデバイス管理には、通常、システムでの周辺デバイスの追加と削除、デバイスをサポートするための Sun 以外のデバイスドライバの追加、システム構成情報の表示が含まれます。

---

## デバイス管理の新機能

ここでは、デバイス管理の新機能について説明します。

## USB デバイスの新機能

USB デバイスの新機能の詳細については、111 ページの「USB デバイスの新機能」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

次の表に、デバイスをホットプラグしたり、シリアルデバイス (プリンタやモデムなど) や周辺デバイス (ディスク、CD-ROM、テープデバイスなど) を追加したりする手順を説明している参照先を示します。

表 5-1 デバイスを追加する場合の参照先

デバイス管理作業	参照先
ホットプラグ不可ディスクの追加	第 12 章または第 13 章
SCSI または PCI デバイスのホットプラグ	86 ページの「cfgadm コマンドによる SCSI ホットプラグ」または 96 ページの「x86: cfgadm コマンドによる PCI ホットプラグ」
USB デバイスのホットプラグ	139 ページの「USB デバイスのホットプラグ (作業マップ)」
CD-ROM またはテープデバイスの追加	77 ページの「周辺デバイスを追加する方法」
モデムの追加	『Solaris のシステム管理 (上級編)』の第 10 章「端末とモデムの管理 (概要)」
プリンタの追加	『Solaris のシステム管理 (上級編)』の第 2 章「印刷サービスの管理 (概要)」

---

## デバイスドライバについて

コンピュータは通常、広範囲の周辺デバイスと大量記憶デバイスを使用します。たとえば、各システムには、SCSI ディスクドライブ、キーボードとマウス、磁気バックアップメディアなどがあります。これ以外に一般に使用されるデバイスには、CD-ROM ドライブ、プリンタとプロッタ、ライトペン、タッチセンサー式画面、デジタルタイザ、タブレットとスタイラスのペアがあります。

Solaris ソフトウェアは、これらのデバイスと直接には通信を行いません。各タイプのデバイスに異なるデータ形式、プロトコル、および転送速度が必要になります。

「デバイスドライバ」は、オペレーティングシステムが特定のハードウェアと通信できるようにする低レベルのプログラムです。このドライバは、そのハードウェアに対するオペレーティングシステムの「インタプリタ」として機能します。

---

## デバイスの自動構成

プラットフォーム固有の構成要素を備えた汎用コアと、一連のモジュールからなるカーネルは、Solaris 環境で自動的に構成されます。

カーネルモジュールとは、システムで固有の作業を実行するために使用されるハードウェアまたはソフトウェアの構成要素のことです。「ロード可能」なカーネルモジュールの例としては、デバイスのアクセス時にロードされるデバイスドライバがあげられます。

プラットフォームに依存しないカーネルは `/kernel/genunix` です。プラットフォーム固有の構成要素は、`/platform/`uname -m`/kernel/unix` です。

カーネルモジュールについては、次の表で説明します。

表 5-2 カーネルモジュール

保存場所	内容
<code>/platform/`uname -m`/kernel</code>	プラットフォーム固有のカーネル構成要素
<code>/kernel</code>	システムのブートに必要なすべてのプラットフォームに共通のカーネル構成要素
<code>/usr/kernel</code>	特定の命令セット内にあるすべてのプラットフォームに共通のカーネル構成要素

システムは、ブート時にどのデバイスが接続されているかを判断します。さらに、カーネルは、それ自体を動的に構成して、必要なモジュールだけをメモリーにロードします。ディスクデバイスやテープデバイスなどのデバイスが初めてアクセスされると、対応するデバイスドライバがロードされます。このプロセスは、「自動構成」と呼ばれます。これは、すべてのカーネルモジュールが、必要に応じて自動的にロードされるためです。

`/etc/system` ファイルを修正することによって、カーネルモジュールがロードされる方法をカスタマイズできます。このファイルを修正する方法については、`system` (4) のマニュアルページを参照してください。

## 自動構成の機能と利点

自動構成の利点は次のとおりです。

- モジュールが必要に応じてロードされるため、主メモリーをより効率的に使用できる。
- 新しいデバイスがシステムに追加されるときに、カーネルを再構成する必要がない。
- カーネルを再構成しないでドライバをロード、テストして、システムをリブートすることができる。

自動構成プロセスは、システム管理者が新しいデバイス (およびドライバ) をシステムに追加するときに使用されます。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。

コマンド	マニュアルページ	説明
<code>prtconf</code>	<code>prtconf (1M)</code>	メモリーの総量、システムのデバイス階層によって記述されたデバイス構成を含む、システム構成情報を表示します。このコマンドによる出力は、システムのタイプによって異なります。
<code>sysdef</code>	<code>sysdef (1M)</code>	システムハードウェア、疑似デバイス、ロード可能なモジュール、および指定のカーネルパラメータを含む、デバイス構成情報を表示します。
<code>dmesg</code>	<code>dmesg (1M)</code>	最後のリ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。

システムのデバイスの識別に使用されるデバイス名については、156 ページの「[デバイス名の命名規則](#)」を参照してください。

## driver not attached メッセージ

次のドライバ関連メッセージが、`prtconf` コマンドと `sysdef` コマンドによって表示されることがあります。

```
device, instance #number (driver not attached)
```

このメッセージは、このデバイスのドライバが使用できないことをいつも示すわけではありません。このメッセージは、ノードにデバイスがないか、あるいはデバイスが使用中ではないために、デバイスインスタンスに「現在」接続されているドライバがないことを示します。ドライバは、デバイスがアクセスされると自動的にロードされ、デバイスが使用されなくなると自動的にアンロードされます。

## システムデバイスの識別

`prtconf` および `sysdef` コマンドの出力から、システムに接続されているディスク、テープ、および CD-ROM デバイスを識別できます。デバイスインスタンスの出力の横に `driver not attached` メッセージが表示されます。これらのデバイスは、何らかのシステムプロセスによって常に監視されているため、「`driver not attached`」メッセージは通常、そのデバイスインスタンスにデバイスがないことを示す良い標識になります。

たとえば、次の `prtconf` 出力は、instance #3 と instance #6 のデバイスを識別しています。これは、最初の SCSI ホストアダプタ (`esp`, instance #0) のターゲット 3 のディスクデバイスと、ターゲット 6 の CD-ROM デバイスを示しています。

```
$ /usr/sbin/prtconf
.
.
.

esp, instance #0
    sd (driver not attached)
    st (driver not attached)
    sd, instance #0 (driver not attached)
    sd, instance #1 (driver not attached)
    sd, instance #2 (driver not attached)
    sd, instance #3
    sd, instance #4 (driver not attached)
    sd, instance #5 (driver not attached)
    sd, instance #6
.
.
.
```

次のコマンドを使うと、システムに接続されたデバイスのみを表示できます。

```
$ prtconf | grep -v not
```

デバイス情報は、`sysdef` 出力からも得られます。

## システム構成情報を表示する方法

システム構成情報を表示するには、`prtconf` コマンドを使用してください。

```
# /usr/sbin/prtconf
```

疑似デバイス、ロード可能なモジュール、および指定のカーネルパラメータを含むシステム構成情報を表示するには、`sysdef` コマンドを使用してください。

```
# /usr/sbin/sysdef
```

### 例 — システム構成情報を表示する

SPARC システムでは、次の `prtconf` 出力が表示されます。

```
# prtconf
System Configuration: Sun Microsystems sun4u
Memory size: 128 Megabytes
System Peripherals (Software Nodes):
SUNW,Ultra-5_10
    packages (driver not attached)
```

```

terminal-emulator (driver not attached)
deblocker (driver not attached)
obp-tftp (driver not attached)
disk-label (driver not attached)
SUNW,builtin-drivers (driver not attached)
sun-keyboard (driver not attached)
ufs-file-system (driver not attached)
chosen (driver not attached)
openprom (driver not attached)
  client-services (driver not attached)
options, instance #0
aliases (driver not attached)
memory (driver not attached)
virtual-memory (driver not attached)
pci, instance #0
  pci, instance #0
    ebus, instance #0
      auxio (driver not attached)
      power, instance #0
      SUNW,pll (driver not attached)
      se, instance #0
      su, instance #0
      su, instance #1
      ecpp (driver not attached)
      fdthree, instance #0
.
.
.

```

x86 システムからは、次の sysdef 出力が表示されます。

```

# sysdef
* Hostid
*
  29f10b4d
*
* i86pc Configuration
*
*
* Devices
*
+boot (driver not attached)
memory (driver not attached)
aliases (driver not attached)
chosen (driver not attached)
i86pc-memory (driver not attached)
i86pc-mmio (driver not attached)
openprom (driver not attached)
options, instance #0
packages (driver not attached)
delayed-writes (driver not attached)
itu-props (driver not attached)
isa, instance #0
  motherboard (driver not attached)
  pnpADP,1542, instance #0

```

```

    asy, instance #0
    asy, instance #1
    lp, instance #0 (driver not attached)
    fdc, instance #0
        fd, instance #0
        fd, instance #1 (driver not attached)
    kd (driver not attached)
    kdmouse (driver not attached)
.
.
.

```

## デバイス情報を表示する方法

デバイス情報は、`dmesg` コマンドを使用して表示してください。

```
# /usr/sbin/dmesg
```

この `dmesg` 出力は、システムコンソール上のメッセージとして表示され、最後のリブート以降に接続されたデバイスを表示します。

### 例 — デバイス情報を表示する

SPARC システムからは、次の `dmesg` 出力が表示されます。

```

Apr  2 13:26:19 venus genunix: [ID 540533 kern.notice] SunOS Release 5.9 Version Generic ...
Apr  2 13:26:19 venus genunix: [ID 943905 kern.notice] Copyright 1983-2003...
Apr  2 13:26:19 venus genunix: [ID 678236 kern.info] Ethernet address ...
Apr  2 13:26:19 venus unix: [ID 389951 kern.info] mem = 65536K (0x4000000)
Apr  2 13:26:19 venus unix: [ID 930857 kern.info] avail mem = 57688064
Apr  2 13:26:19 venus rootnex: [ID 466748 kern.info] root nexus =
Sun Ultra 1 SBus (UltraSPARC 167MHz)

```

x86 システムからは、次の `dmesg` 出力が表示されます。

```

# dmesg
Dec 17 16:32:10 naboo unix: [ID 930857 kern.info] avail mem = 1037565952
Dec 17 16:32:10 naboo rootnex: [ID 466748 kern.info] root nexus = i86pc
Dec 17 16:32:10 naboo unix: [ID 406534 kern.info] ACPI detected: 2 13 0 0
Dec 17 16:32:10 naboo rootnex: [ID 349649 kern.info] pci1 at root: isa 0x0
Dec 17 16:32:10 naboo genunix: [ID 936769 kern.info] pci1 is /pci@1,0
Dec 17 16:32:10 naboo pcplusmp: [ID 637496 kern.info] pcplusmp: ...
Dec 17 16:32:10 naboo pci: [ID 370704 kern.info] PCI-device: ...
Dec 17 16:32:10 naboo genunix: [ID 936769 kern.info] cadp1601 ...
Dec 17 16:32:13 naboo scsi: [ID 193665 kern.info] sd5 at cadp1601: ...
.
.
.

```

---

## システムへ周辺デバイスを追加する

新しい (ホットプラグイン不可の) 周辺デバイスを追加する場合、通常、次の作業が必要になります。

- システムのシャットダウン
- システムへのデバイスの接続
- システムのリブート

システムにホットプラグインできない次のデバイスを追加する場合は、[77 ページ](#)の「[周辺デバイスを追加する方法](#)」の手順に従ってください。

- CD-ROM
- 二次ディスクドライブ
- テープドライブ
- SBUS カード

場合によっては、新しいデバイスをサポートするために、Sun 以外のデバイスドライバを追加しなければなりません。

ホットプラグインデバイスについては、[第 6 章](#)を参照してください。

### ▼ 周辺デバイスを追加する方法

- 手順
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、[78 ページ](#)の「[デバイスドライバを追加する方法](#)」の手順 **2** と **3** を実行します。
  3. `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

この `/reconfigure` ファイルがあると、Solaris ソフトウェアは、次にシステムに電源を入れたときまたはブートしたときに、新しくインストールされたデバイスがないかどうかをチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -g30 -y
```

-i0 システムを `init 0` 状態に戻す。システムの電源を落としてデバイスの追加、削除を行うのに適した状態になる。

-g30 システムを 30 秒以内にシャットダウンする。デフォルト値は 60 秒。

-y ユーザーの介入なしに、システムのシャットダウンを続ける。このオプションを指定しないと、シャットダウンプロセスを続けるかどうか、プ

ロンプトでたずねられる。

5. システムがシャットダウンしたら、次のいずれかを選択して電源を落とします。
  - a. **SPARC** プラットフォームでは、**ok** プロンプトが表示されたら電源を落としても安全です。
  - b. **x86** プラットフォームでは、**type any key to continue** プロンプトが表示されたら電源を落としても安全です。電源スイッチの位置については、各システムに添付のハードウェアマニュアルを参照してください。
6. すべての外部デバイスの電源を落とします。  
周辺デバイスの電源スイッチの位置については、各自の周辺デバイスに添付のハードウェアマニュアルを参照してください。
7. 周辺デバイスをインストールして、追加するデバイスのターゲット番号がシステム上の他のデバイスとは異なることを確認します。  
ターゲット番号を選択するために、ディスクの裏側にある小さいスイッチを見つけてください。  
デバイスの設置と接続については、周辺デバイスに添付のハードウェアマニュアルを参照してください。
8. システムの電源を入れます。  
システムがブートされてマルチユーザーモードになり、ログインプロンプトが表示されます。
9. 周辺デバイスにアクセスし、そのデバイスが追加されたことを確認してください。  
デバイスにアクセスする方法については、[第9章](#)を参照してください。

## ▼ デバイスドライバを追加する方法

この手順では、デバイスがすでにシステムに追加されていることを前提としています。追加されていない場合は、[72 ページ](#)の「標準サポートされていないデバイスを使用する場合」を参照してください。

- 手順
1. スーパーユーザーになります。
  2. テープ、フロッピーディスク、または **CD-ROM** をドライブに入れます。
  3. ドライバをインストールします。

```
# pkgadd -d device package-name
```

-d device            パッケージを含むデバイスのパス名を指定する。

*package-name* デバイスドライバを含むパッケージ名を指定する。

4. パッケージが正常に追加されたことを確認します。

```
# pkgchk package-name
#
パッケージが正しくインストールされている場合は、何も表示されません。
```

#### 例 5-1 デバイスドライバを追加する

次の例では、XYZdrv というパッケージをインストールして確認します。

```
# pkgadd XYZdrv
(ライセンス関連のメッセージが表示される)
.
.
.
Installing XYZ Company driver as <XYZdrv>
.
.
.
Installation of <XYZdrv> was successful.
# pkgchk XYZdrv
#
```





## 第 6 章

---

# デバイスの動的構成 (手順)

---

この章では、Solaris 環境でデバイスを動的に構成する手順について説明します。Solaris 環境でシステム構成要素がホットプラグ機能をサポートする場合、システムが動作しているときにも、デバイスを追加、削除、または交換することができます。システムの構成要素がホットプラグをサポートしていない場合は、システムをリブートしてデバイスを再構成します。

デバイスを動的に構成する手順については、次の項目を参照してください。

- 85 ページの「[cfgadm コマンドによる SCSI ホットプラグ \(作業マップ\)](#)」
- 95 ページの「[cfgadm コマンドによる PCI ホットプラグ \(作業マップ\)](#)」
- 102 ページの「[アプリケーション開発者 RCM スクリプト \(作業マップ\)](#)」
- 103 ページの「[システム管理者 RCM スクリプト \(作業マップ\)](#)」

cfgadm コマンドを使用して USB デバイスをホットプラグする方法については、[148 ページの「cfgadm コマンドを使った USB デバイスのホットプラグ」](#)を参照してください。

デバイスへのアクセス方法については、[第 9 章](#)を参照してください。

---

## 動的再構成とホットプラグ機能

「ホットプラグ機能」とは、システムの動作中に、システム構成要素を物理的に取り付け、取り外し、または交換できる機能のことです。「動的再構成」とは、システム構成要素をホットプラグできる機能のことです。また動的再構成は、システムリソースをシステムから物理的に取り外さなくても (なんらかの方法で) システムリソース (ハードウェアとソフトウェアの両方) をシステム内で移動したり、無効にできる機能のことです。

cfgadm コマンドを使用すると、次のデバイスをホットプラグできます。

- SPARC および x86 プラットフォームの USB デバイス

- SPARC および x86 プラットフォームの SCSI デバイス
- x86 プラットフォームの PCI デバイス

cfgadm コマンドには、次のような機能があります。

- システム構成要素の状態の表示
- システム構成要素の検査
- システム構成要素の構成の変更
- 構成ヘルプメッセージの表示

cfgadm コマンドでシステム構成要素を再構成する利点は、システムが動作しているときでも、システム構成要素を取り付け、取り外し、または交換できることです。さらに、cfgadm コマンドでは、システム構成要素を取り付け、取り外し、または交換するために必要な手順が示されます。

SCSI 構成要素をホットプラグする手順については、[cfgadm\(1M\)](#) のマニュアルページと [86 ページ](#) の「[cfgadm コマンドによる SCSI ホットプラグ](#)」を参照してください。x86 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、[96 ページ](#) の「[x86: cfgadm コマンドによる PCI ホットプラグ](#)」を参照してください。

---

注 – すべての SCSI と PCI のコントローラが cfgadm コマンドによるホットプラグ機能をサポートしているわけではありません。

---

Sun が提供する高可用性の一部として、動的再構成は他の階層化製品 (代替パス指定やフェイルオーバーソフトウェアなど) とともに使用することをお勧めします。両方の製品ともデバイス障害に対する耐性を提供します。

高可用性ソフトウェアがなくても、障害が発生したデバイスを交換できます。この場合、適切なアプリケーションを手動で停止し、重要でないファイルシステムのマウントを手動で解除し、デバイスを取り付けまたは取り外します。

---

注 – 企業レベルのシステムなど、特定のハードウェア構成に対するデバイスのホットプラグについては、お使いのハードウェア構成のマニュアルを参照してください。

---

## 接続点

cfgadm コマンドは接続点についての情報を表示します。「接続点」とは、動的再構成を行うことができるシステム内の特定の場所のことです。

接続点は、次の要素から構成されています。

- 占有装置 (*occupant*): システムに構成できるハードウェア構成要素のことです。
- 受容体 (*receptacle*): 占有装置を受け入れる場所のことです。

接続点は、論理と物理の両方の接続点 ID (Ap\_Id) で表現されます。物理 Ap\_Id は接続点の物理的なパス名です。論理 Ap\_Id は物理 Ap\_Id に代わるユーザーに理解しやすい ID です。Ap\_Id の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

通常、SCSI HBA (Host Bus Adapter)、つまり、SCSI コントローラの論理 Ap\_Id はコントローラ番号 (c0 など) で表現されます。

コントローラ番号が SCSI HBA に割り当てられていない場合、内部的に生成された固有の識別子が提供されます。たとえば、SCSI コントローラの固有な識別子は次のようになります。

```
fas1:scsi
```

通常、SCSI デバイスの論理 Ap\_Id は次のようになります。

```
HBA-logical-apid::device-identifier
```

次の例において、c0 は SCSI HBA の論理 Ap\_Id です。

```
c0::dsk/c0t3d0
```

通常、デバイス識別子は /dev ディレクトリ内にある論理デバイス名から導き出されます。たとえば、論理デバイス名が /dev/rmt/1 のテープデバイスの論理 Ap\_Id は次のようになります。

```
c0::rmt/1
```

SCSI デバイスの論理 Ap\_Id を /dev ディレクトリ内にある論理デバイス名から導き出すことができない場合、内部的に生成された固有の識別子が提供されます。たとえば、/dev/rmt/1 テープデバイスの識別子は、次のようになります。

```
c0::st4
```

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

`cfgadm` コマンドはすべてのリソースと動的再構成の操作を、一般的な状態 (`configured`、`unconfigured` など) や操作 (`connect`、`configure`、`unconfigure` など) を示す用語で表現します。一般的な状態や操作の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

次に、SCSI HBA 接続点の受容体と占有装置の状態を説明します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI HBA には利用できない	configured	1 つまたは複数のデバイスがバス上で構成されている

受容体の状態	説明	占有装置の状態	説明
disconnected	バスは休止している	unconfigured	デバイスは構成されていない
connected	バスはアクティブである		

次に、SCSI デバイス接続点の受容体と占有装置の状態を説明します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI デバイスには利用できない	configured	デバイスが構成されている
disconnected	バスは休止している	unconfigured	デバイスは構成されていない
connected	バスはアクティブである		

SCSI 接続点の状態は特別なハードウェアによって示されない限り未知です。SCSI 構成要素の構成情報を表示する手順については、[86 ページの「SCSI デバイスに関する情報を表示する方法」](#)を参照してください。

## x86: PCI アダプタカードの取り外し

デバイスドライバがホットプラグ機能をサポートしている場合、必須でないシステムリソースとして機能している PCI アダプタカードは取り外すことができます。重要なシステムリソースとして機能している PCI アダプタカードは取り外すことができません。PCI アダプタカードが取り外し可能であるためには、次の条件が必要です。

- デバイスドライバはホットプラグ機能をサポートしていなければなりません。
- 重要なリソースには代替バスでアクセスできなければなりません。

たとえば、システムにインストールされているイーサネットカードが 1 つしかない場合、ネットワーク接続を切断せずにこのイーサネットカードを取り外すことは不可能です。このような環境でネットワーク接続をアクティブに保ったままイーサネットカードを取り外すには、別の階層化ソフトウェアサポートが必要です。

## x86: PCI アダプタカードの取り付け

PCI アダプタカードをシステムに取り付けるには、次の条件が必要です。

- スロットが利用できなければなりません。
- デバイスドライバが当該アダプタカードのホットプラグ機能をサポートしていなければなりません。

PCI アダプタカードの取り付けまたは取り外しの手順については、96 ページの「x86: cfgadm コマンドによる PCI ホットプラグ」を参照してください。

## cfgadm コマンドによる SCSI ホットプラグ (作業マップ)

作業	説明	参照先
1. SCSI デバイスについての情報の表示	SCSI コントローラおよびデバイスについての情報を表示する	86 ページの「SCSI デバイスに関する情報を表示する方法」
2. SCSI コントローラの構成解除	SCSI コントローラの構成を解除する	87 ページの「SCSI コントローラの構成を解除する方法」
3. SCSI コントローラの構成	構成を解除された SCSI コントローラを構成する	87 ページの「SCSI コントローラを構成する方法」
4. SCSI デバイスの構成	SCSI デバイスを構成する	88 ページの「SCSI デバイスを構成する方法」
5. SCSI コントローラを切り離す	特定の SCSI コントローラを切り離す	89 ページの「SCSI コントローラを切り離す方法」
6. SCSI コントローラの接続	切り離された特定の SCSI コントローラを接続する	90 ページの「SPARC: SCSI コントローラを接続する方法」
7. SCSI デバイスを SCSI バスに取り付ける	特定の SCSI デバイスを SCSI バスに取り付ける	90 ページの「SPARC: SCSI デバイスを SCSI バスに取り付ける方法」
8. SCSI コントローラ上の同一デバイスと交換	SCSI バス上のデバイスを、同じタイプの別のデバイスに交換する	91 ページの「SPARC: SCSI コントローラ上の同一デバイスを交換する方法」
9. SCSI デバイスを取り外す	SCSI デバイスをシステムから取り外す	92 ページの「SPARC: SCSI デバイスを取り外す方法」
10. SCSI 構成に関する問題の障害追跡	失敗した SCSI 構成解除操作を解決する	95 ページの「失敗した SCSI 構成解除操作の解決方法」

---

# cfgadm コマンドによる SCSI ホットプラグ

この節では、cfgadm コマンドを使用してさまざまな SCSI ホットプラグ処理を実行する方法について説明します。

この節で説明する手順では、特定のデバイスを使用して、cfgadm コマンドで SCSI 構成要素をホットプラグする例を示します。cfgadm コマンドで提供されるデバイス情報や表示されるデバイス情報は、システム構成によって異なります。

## ▼ SCSI デバイスに関する情報を表示する方法

次の手順では、SCSI コントローラ c0 と c1、およびコントローラに接続されたデバイスを例として使用して、cfgadm コマンドを使用して表示可能なデバイス構成情報を示します。

---

注 - SCSI デバイスが cfgadm コマンドでサポートされていない場合、その SCSI デバイスは cfgadm コマンドの出力には表示されません。

---

手順 1. スーパーユーザーになります。

2. システムの接続点についての情報を表示します。

```
# cfgadm -l
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
```

この例では、c0 と c1 は 2 つの SCSI コントローラを表しています。

3. システムの **SCSI** コントローラとこれらに接続されているデバイスについての情報を表示します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::disk/c0t0d0  disk         connected   configured  unknown
c0::rmt/0       tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::disk/c1t3d0  disk         connected   configured  unknown
c1::disk/c1t4d0  unavailable  connected   unconfigured unknown
```

---

注 - `cfgadm -l` コマンドは、SCSI デバイスではなく、SCSI HBA についての情報を表示します。ディスクやテープなどの SCSI デバイスについての情報を表示するには、`cfgadm -al` コマンドを使用してください。

---

## ▼ SCSI コントローラの構成を解除する方法

次の手順では、SCSI コントローラ `c1` を使用して、SCSI コントローラの構成を解除する例を示します。

手順 1. スーパーユーザーになります。

2. SCSI コントローラの構成を解除します。

```
# cfgadm -c unconfigure c1
```

3. SCSI コントローラの構成が解除されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected   configured unknown
c0::dsk/c0t0d0       disk                connected   configured unknown
c0::rmt/0             tape                connected   configured unknown
c1                   scsi-bus            connected   unconfigured unknown
```

`c1` の `Occupant` の列に `unconfigured` と表示されていることに注目してください。これは、SCSI バスに占有装置が構成されていないことを示します。

構成解除処理が失敗した場合は、95 ページの「失敗した SCSI 構成解除操作の解決方法」を参照してください。

## ▼ SCSI コントローラを構成する方法

次の手順では、SCSI コントローラ `c1` を使用して、SCSI コントローラを構成する例を示します。

手順 1. スーパーユーザーになります。

2. SCSI コントローラを構成します。

```
# cfgadm -c configure c1
```

3. SCSI コントローラが構成されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected   configured unknown
```

c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	unavailable	connected	unconfigured	unknown

前述の構成を解除する例では、SCSI バス上のすべてのデバイスを削除しました。この例では、すべてのデバイスをシステムに構成し直します。

## ▼ SCSI デバイスを構成する方法

次の手順では、SCSI ディスク c1t4d0 を使用して、SCSI デバイスを構成する例を示します。

- 手順
1. スーパーユーザーになります。
  2. 構成するデバイスを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 unavailable  connected   unconfigured unknown
```

3. SCSI デバイスを構成します。

```
# cfgadm -c configure c1::dsk/c1t4d0
```

4. SCSI デバイスが構成されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 disk         connected   configured  unknown
```



## ▼ SCSI コントローラを切り離す方法



---

注意 - SCSI デバイスを切り離すときは十分に注意してください。特に、ルート (/)、usr、var、swap パーティションなどの重要なファイルシステムが入っているディスクのコントローラを扱うときは十分に注意を払ってください。動的再構成ソフトウェアは、システムがハングする原因をすべて発見できるわけではありません。この手順は、十分注意して実行してください。

---

次の手順では、SCSI コントローラ c1 を使用して、SCSI デバイスを切り離す例を示します。

- 手順
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 バスが切り離されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             unavailable  disconnected  configured  unknown
c1::dsk/c1t10d0 unavailable  disconnected  configured  unknown
c1::dsk/c1t4d0 unavailable  disconnected  configured  unknown
```

コントローラとそれに接続されていたすべてのデバイスがシステムから切り離されました。

## ▼ SPARC: SCSI コントローラを接続する方法

次の手順は、SCSI コントローラ `c1` を使用して、SCSI コントローラを接続する例を示します。

- 手順 1. スーパーユーザーになります。
2. デバイスを接続する前に、デバイスが切り離されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::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。

---

- 手順 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
```

a. Continue (yes/no)? というプロンプトに **y** と入力して、次に進みます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

b. デバイスを接続して、電源を入れます。

c. Enter y if operation is complete or n to abort (yes/no)? というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. デバイスが取り付けられていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected   configured unknown
c0::dsk/c0t0d0       disk                 connected   configured unknown
c0::rmt/0            tape                 connected   configured unknown
c1                   scsi-bus            connected   configured unknown
c1::dsk/c1t3d0       disk                 connected   configured unknown
c1::dsk/c1t4d0       disk                 connected   configured unknown
```

コントローラ c1 に新しいディスクが取り付けられました。

## ▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法

次の手順では、SCSI ディスク c1t4d0 を使用して、SCSI コントローラ上の同一デバイスを交換する例を示します。

- 手順
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
```

- a. **Continue (yes/no)?** というプロンプトに **y** と入力して、次に進みます。  
ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

- b. デバイスの電源を切ってから、そのデバイスを取り外します。
- c. 交換用のデバイスを取り付けます。そして、取り付けたデバイスの電源を入れます。  
交換用のデバイスは取り外したデバイスと同じタイプであり、同じアドレス (ターゲットと論理ユニット番号) でなければなりません。

- d. **Enter y if operation is complete or n to abort (yes/no)?** というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. デバイスが交換されていることを確認します。

```
# cfgadm -al
Ap_Id          Type          Receptacle  Occupant    Condition
c0             scsi-bus     connected   configured  unknown
c0::dsk/c0t0d0 disk         connected   configured  unknown
c0::rmt/0      tape         connected   configured  unknown
c1             scsi-bus     connected   configured  unknown
c1::dsk/c1t3d0 disk         connected   configured  unknown
c1::dsk/c1t4d0 disk         connected   configured  unknown
```

## ▼ SPARC: SCSI デバイスを取り外す方法

次に、SCSI ディスク c1t4d0 を使用して、SCSI コントローラ上のデバイスを取り外す例を示します。

- 手順 1. スーパーユーザーになります。

2. 現在の SCSI 構成を確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected   configured unknown
c0::dsk/c0t0d0       disk                connected   configured unknown
c0::rmt/0            tape                connected   configured unknown
c1                   scsi-bus            connected   configured unknown
c1::dsk/c1t3d0       disk                connected   configured unknown
c1::dsk/c1t4d0       disk                connected   configured unknown
```

3. SCSI デバイスをシステムから取り外します。

```
# cfgadm -x remove_device c1::dsk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
```

a. Continue (yes/no)? というプロンプトに **y** と入力して、次に進みます。

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

b. デバイスの電源を切ってから、そのデバイスを取り外します。

c. Enter y if operation is complete or n to abort (yes/no)? というプロンプトに **y** と入力します。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. デバイスがシステムから取り外されていることを確認します。

```
# cfgadm -al
Ap_Id                Type                Receptacle  Occupant  Condition
c0                   scsi-bus            connected   configured unknown
c0::dsk/c0t0d0       disk                connected   configured unknown
c0::rmt/0            tape                connected   configured unknown
c1                   scsi-bus            connected   configured unknown
c1::dsk/c1t3d0       disk                connected   configured unknown
```

## SPARC: SCSI 構成に関する問題の障害追跡

この節では、SCSI 構成に関する問題の障害追跡を行うために、エラーメッセージとその解決策について説明します。SCSI 構成に関する問題の障害追跡の詳細については、`cfgadm(1M)` のマニュアルページを参照してください。

### エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource                Information
-----
```

```
/dev/dsk/c1t0d0s0  mounted filesystem "/file-system"
```

#### 原因

ファイルシステムがマウントされているデバイスを取り外しまたは交換しようとしてしました。

#### 解決法

エラーメッセージのリストにあるファイルシステムのマウントを解除してから、もう一度 `cfgadm` コマンドを実行します。

`cfgadm` コマンドを使用して、スワップデバイス、専用のダンプデバイスなどのシステムリソースを取り外すと、システムリソースがアクティブな場合、次のようなエラーメッセージが表示されます。

#### エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource           Information
-----
/dev/dsk/device-name  swap area
```

#### 原因

1 つまたは複数の構成されているスワップ領域を削除または置き換えようとしてしました。

#### 解決法

指定されたデバイス上のスワップ領域の構成を解除してから、再度 `cfgadm` を実行します。

#### エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource           Information
-----
/dev/dsk/device-name  dump device (swap)
```

#### 原因

スワップ領域上に構成されているダンプデバイスを取り外そうとしたか、置き換えようとしてしました。

#### 解決法

スワップ領域に構成されているダンプデバイスの構成を解除してから、再度 `cfgadm` を実行します。

#### エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
  device path
  Resource           Information
-----
/dev/dsk/device-name  dump device (dedicated)
```

#### 原因

専用ダンプデバイスを取り外そうとしたか、置き換えようとしてしました。

#### 解決法

専用ダンプデバイスの構成を解除し、`cfgadm` 処理を再実行します。

### ▼ 失敗した SCSI 構成解除操作の解決方法

1 つ以上のターゲットデバイスが使用中である場合、および SCSI 構成解除操作が失敗した場合、次の手順を使用します。この手順を使用しないと、将来、このコントローラおよびターゲットデバイスの動的再構成操作が失敗し、`dr in progress` メッセージが表示されます。

- 手順
1. スーパーユーザーになります (まだなっていない場合)。
  2. 次のコマンドを入力してコントローラを再構成します。

```
# cfgadm -c configure device-name
```

---

## cfgadm コマンドによる PCI ホットプラグ (作業マップ)

作業	説明	参照先
1. PCI スロット構成情報の表示	システムの PCI ホットプラグ可能なデバイスとスロットの状態を表示する	96 ページの「x86: PCI スロット構成情報を表示する方法」
2. PCI アダプタカードを取り外す	カードを構成解除してからスロットから電源を外し、システムからカードを取り外す	97 ページの「x86: PCI アダプタカードを取り外す方法」
3. PCI アダプタカードを取り付ける	アダプタカードをホットプラグ可能なスロットに挿入する。スロットを電源に接続し、カードを構成する	98 ページの「x86: PCI アダプタカードを取り付ける方法」
4. PCI 構成に関する問題の障害追跡	PCI 構成障害に対処するために、エラーメッセージと解決策を確認する	99 ページの「x86: PCI 構成に関する問題の障害追跡」

---

## x86: cfgadm コマンドによる PCI ホットプラグ

この節では、x86 システム上で PCI アダプタカードをホットプラグする手順について説明します。

次の例では、簡潔にするため、PCI 接続点だけを表示しています。画面に表示される接続点はシステムによって異なります。

### ▼ x86: PCI スロット構成情報を表示する方法

cfgadm コマンドは、システム上の PCI ホットプラグ可能なデバイスとスロットの状態を表示します。詳細については、cfgadm(1M) のマニュアルページを参照してください。

- 手順
1. スーパーユーザーになります。
  2. PCI スロット構成情報を表示します。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        connected   configured ok
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

具体的な PCI デバイス情報を表示します。

```
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                Type                Information
pci1:hpc0_slot0     unknown            Slot 7
pci1:hpc0_slot1     unknown            Slot 8
pci1:hpc0_slot2     unknown            Slot 9
pci1:hpc0_slot3     ethernet/hp        Slot 10
pci1:hpc0_slot4     unknown            Slot 11
```

論理 Ap\_Id の pci1:hpc0\_slot0 は、ホットプラグ可能なスロット Slot 7 の論理 Ap\_Id です。構成要素 hpc0 はこのスロットのホットプラグ可能なアダプタカードを示し、pci1 は PCI バスのインスタンスを示します。Type フィールドは、スロット中の PCI アダプタカードのタイプを示します。



## ▼ x86: PCI アダプタカードを取り外す方法

手順 1. スーパーユーザーになります。

2. PCI アダプタカードが入っているスロットを確認します。

```
# cfgadm
Ap_Id                Type          Receptacle  Occupant    Condition
pci1:hpc0_slot0     unknown      empty        unconfigured unknown
pci1:hpc0_slot1     unknown      empty        unconfigured unknown
pci1:hpc0_slot2     unknown      empty        unconfigured unknown
pci1:hpc0_slot3     ethernet/hp  connected    configured   ok
pci1:hpc0_slot4     unknown      empty        unconfigured unknown
```

3. デバイスを開いているアプリケーションを停止します。

たとえば、イーサネットカードの場合、ifconfig コマンドでインタフェースを無効にしてから、カードを引き抜きます。

4. デバイスの構成を解除します。

```
# cfgadm -c unconfigure pci1:hpc0_slot3
```

5. デバイスの構成が解除されていることを確認します。

```
# cfgadm
Ap_Id                Type          Receptacle  Occupant    Condition
pci1:hpc0_slot0     unknown      empty        unconfigured unknown
pci1:hpc0_slot1     unknown      empty        unconfigured unknown
pci1:hpc0_slot2     unknown      empty        unconfigured unknown
pci1:hpc0_slot3     ethernet/hp  connected    unconfigured unknown
pci1:hpc0_slot4     unknown      empty        unconfigured unknown
```

6. スロットへの電源を切り離します。

```
# cfgadm -c disconnect pci1:hpc0_slot3
```

7. デバイスが切り離されていることを確認します。

```
# cfgadm
Ap_Id                Type          Receptacle  Occupant    Condition
pci1:hpc0_slot0     unknown      empty        unconfigured unknown
pci1:hpc0_slot1     unknown      empty        unconfigured unknown
pci1:hpc0_slot2     unknown      empty        unconfigured unknown
pci1:hpc0_slot3     ethernet/hp  disconnected  unconfigured unknown
pci1:hpc0_slot4     unknown      empty        unconfigured unknown
```

8. スロットのラッチを開いて PCI アダプタカードを取り外します。

## ▼ x86: PCI アダプタカードを取り付ける方法

- 手順
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 **PCI** アダプタカードを挿入します。
  4. **PCI** アダプタカードを挿入した後に、どのスロットに **PCI** アダプタカードが入っているかを確認します。ラッチを閉じます。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        disconnected unconfigured unknown
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

5. スロットへの電源を接続します。

```
# cfgadm -c connect pci1:hpc0_slot3
```

6. 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
```

7. **PCI** アダプタカードを構成します。

```
# cfgadm -c configure pci1:hpc0_slot3
```

8. スロット中の **PCI** アダプタカードの構成を確認します。

```
# cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
pci1:hpc0_slot0     unknown            empty       unconfigured unknown
pci1:hpc0_slot1     unknown            empty       unconfigured unknown
pci1:hpc0_slot2     unknown            empty       unconfigured unknown
pci1:hpc0_slot3     ethernet/hp        connected   configured  unknown
pci1:hpc0_slot4     unknown            empty       unconfigured unknown
```

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、デバイスがイーサネットカードの場合、ifconfig コマンドでインタフェースを設定します。

## x86: PCI 構成に関する問題の障害追跡

エラーメッセージ

```
cfgadm: Configuration operation invalid: invalid transition
```

原因

無効な移行を行いました。

解決法

cfgadm -c コマンドが適切に発行されているかどうかを確認します。cfgadm コマンドで現在の受容体と占有装置の状態を確認し、Ap\_id が正しいことを確認します。

エラーメッセージ

```
cfgadm: Attachment point not found
```

原因

指定した接続点は見つかりません。

解決法

接続点が正しいかどうかを確認します。cfgadm コマンドを使用して、利用可能な接続点のリストを表示します。また、物理パスを調べて、接続点が現在も残っているかどうかを確認します。

---

注 - cfgadm コマンド以外にも、ホットプラグ処理中に便利なコマンドがいくつかあります。prtconf コマンドは、Solaris がハードウェアを認識するかどうかを表示します。ハードウェアの挿入後に prtconf コマンドを使用して、ハードウェアが認識されているかどうかを確認します。構成後に prtconf -D コマンドを使用して、新たにインストールしたハードウェアデバイスにドライバが設定されているかどうかを確認します。

---

## Reconfiguration Coordination Manager (RCM) スクリプトの概要

Reconfiguration Coordination Manager (RCM) は、システムコンポーネントの動的な除去を管理するフレームワークです。RCM を使用すると、システムリソースを順番に登録および解放することができます。

新しい RCM スクリプト機能を使用すると、アプリケーションを停止したり、動的な再構成の間にアプリケーションからデバイスを手際良く解放したりする独自のスクリプトを記述できます。スクリプトによって登録されたリソースに要求が影響を与える場合、RCM フレームワークは再構成要求に応じてスクリプトを自動的に起動します。

リソースを動的に除去する場合は、アプリケーションからリソースを手動で解放しておく必要があります。あるいは、`-f` オプションを指定して `cfgadm` コマンドを使用して再構成オペレーションを強制することも可能でした。ただし、このオプションはアプリケーションを認識不能な状態のままにする可能性があります。また、アプリケーションからリソースを手動で解放すると、一般にエラーが発生します。

RCM スクリプト機能を使うと、動的再構成処理を簡単かつ効果的に実行できます。RCM スクリプトを作成すると、次の操作を実行できます。

- 動的にデバイスを取り外したときにデバイスが自動的に解放される。デバイスがアプリケーションによって起動した場合は、この処理によって、デバイスも終了する
- システムからデバイスを動的に取り外すときに、サイト固有のタスクを実行する

## RCM スクリプトについて

RCM スクリプトは、次のとおりです。

- 実行可能シェルスクリプト (Perl, `sh`, `csh`, または `ksh`) または RCM デーモンが実行するバイナリプログラム。推奨言語は、Perl
- スクリプトファイル所有者のユーザー ID を使用することにより、自分のアドレス領域で実行されるスクリプト
- `cfgadm` コマンドを使ってシステムリソースを動的に再構成するときに、RCM デーモンによって実行されるスクリプト

## RCM スクリプトで実行できること

RCM スクリプトを使用した場合、デバイスを動的に取り外すと、デバイスがアプリケーションから解放されます。デバイスが開いている場合には、RCM スクリプトによって閉じられます。

たとえば、テープバックアップアプリケーションで RCM スクリプトを使用して、テープドライブを終了させたり、テープバックアップアプリケーションをシャットダウンしたりすることができます。

## RCM スクリプト処理の動作方法

次のようにしてスクリプトを起動します。

```
$ script-name command [args ...]
```

スクリプトにより、次の基本的な手順が実行されます。

1. コマンド行引数から RCM コマンドを取得する
2. コマンドを実行する

3. 結果を名前と値のペアで `stdout` に記述する
4. 適切な終了ステータスで終了する

RCM デーモンは、スクリプトのインスタンスを同時に 1 つ実行します。たとえば、RCM デーモンは、スクリプトの実行中には、そのスクリプトが終了するまで同じスクリプトを実行しません。

## RCM スクリプトコマンド

次の RCM コマンドを RCM スクリプトに含める必要があります。

- `scriptinfo` - スクリプト情報を収集する
- `register` - リソースに処理対象を登録する
- `resourceinfo` - リソース情報を収集する

次の RCM コマンドの一部またはすべてを RCM スクリプトに含めることができます。

- `queryremove` - リソースが解放されたかどうかを問い合わせる
- `preremove` - リソースを解放する
- `postremove` - リソースの削除後に通知する
- `undoremove` - `preremove` で実行された動作を元に戻す

これらの RCM コマンドの詳細については、`rcmscript(4)` のマニュアルページを参照してください。

## RCM スクリプト処理環境

デバイスを動的に取り外すと、RCM デーモンにより次のコマンドが実行されます。

- スクリプトで識別されたリソースリスト (デバイス名) を収集するための、スクリプトの `register` コマンドが実行される
- スクリプトの登録されたリソースが動的な取り外し操作によって影響を受ける場合、リソースを取り外す前にスクリプトの `queryremove/preremove` コマンドが実行される
- 取り外し操作が成功した場合に、スクリプトの `postremove` コマンドが実行される。ただし、取り外し操作に失敗した場合、RCM デーモンによりスクリプトの `undoremove` コマンドが実行される

## RCM スクリプトでの作業

次の節では、アプリケーション開発者およびシステム管理者のために RCM スクリプト作業について説明します。

## アプリケーション開発者 RCM スクリプト (作業マップ)

次の作業マップでは、RCM スクリプトを作成するアプリケーション開発者の作業について説明します。

作業	説明	参照先
1. アプリケーションが使用するリソースを特定する	アプリケーションが使用するリソース (デバイス名) を特定する。このデバイスは動的に取り外される可能性がある	cfgadm (1M)
2. リソースを解放するコマンドを特定する	アプリケーションからリソースを完全に解放するようにアプリケーションに通知するコマンドを特定する	アプリケーションのマニュアル
3. リソースを取り外した後に使用するコマンドを特定する	リソースを取り外したことをアプリケーションに通知するコマンドを含める	rcmscript (4)
4. リソースの取り外しに失敗した場合のコマンドを特定する	使用可能なリソースについてアプリケーションに通知するコマンドを含める	rcmscript (4)
5. RCM スクリプトを記述する	前の作業で特定した情報に基づいて RCM スクリプトを記述する	105 ページの「テープバックアップ用の RCM スクリプトの例」
6. RCM スクリプトをインストールする	適切なスクリプトディレクトリにスクリプトを追加する	104 ページの「RCM スクリプトのインストール方法」
7. RCM スクリプトをテストする	手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストする	105 ページの「RCM スクリプトのテスト方法」

## システム管理者 RCM スクリプト (作業マップ)

ここでは、サイトをカスタマイズするために RCM スクリプトを作成するシステム管理者の作業について説明します。

作業	説明	参照先
1. 動的に移動するリソースを特定する	cfgadm -l コマンドを使って移動する可能性があるリソース (デバイス名) を特定する	cfgadm (1M)
2. 停止するアプリケーションを特定する	アプリケーションを完全に停止させるコマンドを特定する	アプリケーションのマニュアル
3. リソースの取り外し前および取り外し後のコマンドを特定する	リソースを取り外す前後の動作を特定する	rcmscript (4)
4. RCM スクリプトを記述する	前の作業で特定した情報に基づいて RCM スクリプトを記述する	105 ページの「テープバックアップ用の RCM スクリプトの例」
5. RCM スクリプトをインストールする	適切なスクリプトディレクトリにスクリプトを追加する	104 ページの「RCM スクリプトのインストール方法」
6. RCM スクリプトをテストする	手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストする	105 ページの「RCM スクリプトのテスト方法」

## RCM スクリプトに名前を付ける

次の規則に従って、スクリプトに *vendor*、*service* という名前を付ける必要があります。

*vendor* スクリプトを提供するベンダーのストックシンボル、またはベンダーを識別する固有な名

*service* スクリプトが表すサービス名

## RCM スクリプトのインストールまたは削除

RCM スクリプトのインストールまたは削除を行うには、スーパーユーザー (root) の権限が必要です。この表を使用して、RCM スクリプトをインストールするディレクトリを判断してください。

表 6-1 RCM スクリプトディレクトリ

ディレクトリの位置	スクリプトタイプ
/etc/rcm/scripts	特定のシステム用のスクリプト
/usr/platform/`uname -i`/lib/rcm/scripts	特定のハードウェア実装用のスクリプト
/usr/platform/`uname -m`/lib/rcm/scripts	特定のハードウェアクラス用のスクリプト
/usr/lib/rcm/scripts	任意のハードウェア用のスクリプト

## ▼ RCM スクリプトのインストール方法

- 手順
1. スーパーユーザーになります。
  2. スクリプトを表 6-1 で説明されている適切なディレクトリにコピーします。  
たとえば、次のようになります。

```
# cp SUNW,sample.pl /usr/lib/rcm/scripts
```

3. スクリプトのユーザー ID およびグループ ID を希望の値に変更します。

```
# chown user:group /usr/lib/rcm/scripts/SUNW,sample.pl
```

4. SIGHUP を RCM デーモンに送信します。

```
# pkill -HUP -x -u root rcm_daemon
```

## ▼ RCM スクリプトの削除方法

- 手順
1. スーパーユーザーになります。
  2. RCM スクリプトディレクトリからスクリプトを削除します。  
たとえば、次のようになります。

```
# rm /usr/lib/rcm/scripts/SUNW,sample.pl
```

3. SIGHUP を RCM デーモンに送信します。

```
# pkill -HUP -x -u root rcm_daemon
```



## ▼ RCM スクリプトのテスト方法

- 手順
1. スクリプトを実行する前にコマンド行シェルに **RCM\_ENV\_FORCE** などの環境変数を設定します。  
たとえば、Korn シェルで次のように設定します。  

```
$ export RCM_ENV_FORCE=TRUE
```
  2. コマンド行から手動でスクリプトコマンドを実行してスクリプトをテストします。  
たとえば、次のようになります。  

```
$ script-name scriptinfo
$ script-name register
$ script-name preremove resource-name
$ script-name postremove resource-name
```
  3. スクリプトの各 **RCM** スクリプトコマンドにより、適切な出力結果が **stdout** に印刷されるかどうかを確認します。
  4. 適切なスクリプトディレクトリにスクリプトをインストールします。  
詳細については、104 ページの「RCM スクリプトのインストール方法」を参照してください。
  5. 動的な削除操作を実行してスクリプトをテストします。  
たとえば、スクリプトによってデバイス `/dev/dsk/c1t0d0s0` が登録されたとします。次のコマンドを実行してください。  

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```



---

注意 – 上記のコマンドは、システムの状態を変化させたり、システム障害を招くおそれもあるため、これらのコマンドを十分理解しておくことは大切です。

---

## テープバックアップ用の RCM スクリプトの例

ここでは、RCM スクリプトを使用したテープバックアップの例を示します。

## テープバックアップ用の RCM スクリプトの役割

テープバックアップ用の RCM スクリプトは、次の手順を実行します。

1. RCM コマンドのディスパッチテーブルを設定します。
2. 指定した RCM コマンドに対応するディスパッチルーチン呼び出し、未実装の RCM コマンドのステータス 2 で終了させます。

3. `scriptinfo` セクションを設定します。

```
rcm_script_func_info=Tape backup appl script for DR
```

4. すべてのテープドライブのデバイス名を `stdout` に印刷して、すべてのテープドライブをシステムに登録します。

```
rcm_resource_name=/dev/rmt/$f
```

エラーが発生した場合、エラー情報を `stdout` に印刷します。

```
rcm_failure_reason=$errmsg
```

5. テープデバイスのリソース情報を設定します。

```
rcm_resource_usage_info=Backup Tape Unit Number $unit
```

6. バックアップアプリケーションがそのデバイスを使用しているかどうか確認して、`preremove` 情報を設定します。バックアップアプリケーションがそのデバイスを使用していない場合、動的再構成操作が続行されます。バックアップアプリケーションがそのデバイスを使用している場合、スクリプトにより `RCM_ENV_FORCE` が検査されます。`RCM_ENV_FORCE` が `FALSE` に設定されている場合、スクリプトにより動的再構成操作が拒否され、次のメッセージが印刷されます。

```
rcm_failure_reason=tape backup in progress pid=...
```

`RCM_ENV_FORCE` が `TRUE` に設定されている場合、バックアップアプリケーションが停止し、再構成操作が続行されます。

## テープバックアップ再構成シナリオの結果

RCM スクリプトを使わずに `cfgadm` コマンドを使ってテープドライブを取り外した場合、次のような結果になります。

- バックアップアプリケーションがテープドライブを使用していない場合に `cfgadm` コマンドを使用すると、操作は正常に実行されます。
- バックアップアプリケーションがテープドライブを使用中に `cfgadm` コマンドを使用すると、操作が失敗します。

RCM スクリプトと `cfgadm` コマンドを使ってテープドライブを取り外した場合、次のような結果になります。

- バックアップアプリケーションがテープドライブを使用していない場合に `cfgadm` コマンドを使用すると、操作は正常に実行されます。
- バックアップアプリケーションがテープドライブを使用しているときに、`-f` オプションを指定せずに `cfgadm` コマンドを使用すると、次のようなメッセージが表示され、操作が失敗します。

```
tape backup in progress pid=...
```

- バックアップアプリケーションがテープドライブを使用しているときに、`-f` オプションを指定して `cfgadm` コマンドを使用すると、スクリプトによってバックアップアプリケーションが停止され、`cfgadm` 操作が正常に実行されます。

## 例 — テープバックアップ用の RCM スクリプト

```
#!/usr/bin/perl -w
#
# サイト用にカスタマイズされた RCM スクリプトの例
#
# RCM_ENV_FORCE が FALSE の場合、
# RCM はテープドライブがバックアップのために
# 使用されているときは、ドライブを解放できない
#
# RCM_ENV_FORCE が TRUE の場合、
# DR はテープドライブをバックアップのために
# 使用しているバックアップアプリケーションを終了してテープドライブを
# 取り外すことができる
#

use strict;

my ($cmd, %dispatch);
$cmd = shift(@ARGV);
# RCM コマンドのテーブルをディスパッチする
%dispatch = (
    "scriptinfo"    =>    \&do_scriptinfo,
    "register"      =>    \&do_register,
    "resourceinfo" =>    \&do_resourceinfo,
    "queryremove"  =>    \&do_preremove,
    "preremove"    =>    \&do_preremove
);

if (defined($dispatch{$cmd})) {
    &{$dispatch{$cmd}};
} else {
    exit (2);
}

sub do_scriptinfo
{
    print "rcm_script_version=1\n";
    print "rcm_script_func_info=Tape backup appl script for DR\n";
    exit (0);
}

sub do_register
{
    my ($dir, $f, $errmsg);

    $dir = opendir(RMT, "/dev/rmt");
    if (!$dir) {
        $errmsg = "Unable to open /dev/rmt directory: $!";
        print "rcm_failure_reason=$errmsg\n";
        exit (1);
    }

    while ($f = readdir(RMT)) {
```

```

        # 非表示のファイルや同一デバイスの複数の名前を無視する
        if (($f !~ /^\.\/) && ($f =~ /^[0-9]+$/)) {
            print "rcm_resource_name=/dev/rmt/$f\n";
        }
    }

    closedir(RMT);
    exit (0);
}

sub do_resourceinfo
{
    my ($rsrc, $unit);

    $rsrc = shift(@ARGV);
    if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
        $unit = $1;
        print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
        exit (0);
    } else {
        print "rcm_failure_reason=Unknown tape device!\n";
        exit (1);
    }
}

sub do_preremove
{
    my ($rsrc);

    $rsrc = shift(@ARGV);

    # このリソースをバックアップアプリケーションが
    # 使用しているかどうかをチェックする
    # if ($rsrc 上でバックアップアプリケーションが動作していない場合) {
    #     DR を続行させる
    #     exit (0);
    # }
    #
    # RCM_ENV_FORCE が FALSE の場合、処理を拒否する
    # RCM_ENV_FORCE が TRUE の場合、
    # バックアップアプリケーションを終了して
    # DR が処理を続行できるようにする
    #
    if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
        if ($cmd eq 'preremove') {
            # テープバックアップアプリケーションを強制終了する
        }
        exit (0);
    } else {
        #
        # テープバックアップアプリケーションによって
        # デバイスが使用されていたため、テープドライブを
        # 解放できなかったことを示す
        #
        print "rcm_failure_reason=tape backup in progress pid=...\n"
    }
}

```

```
;
    exit (3);
}
```



## 第 7 章

---

# USB デバイスの使用 (概要)

---

この章では、Solaris 環境における USB (Universal Serial Bus) デバイスの概要を説明します。

この章の内容は次のとおりです。

- 111 ページの「USB デバイスの新機能」
- 119 ページの「USB デバイスの概要」
- 123 ページの「Solaris 環境における USB について」

Solaris 環境における USB デバイスの使用手順については、第 8 章を参照してください。

動的再構成およびホットプラグについての一般的な情報については、第 6 章を参照してください。

USB プリンタを構成する方法については、『Solaris のシステム管理 (上級編)』の「印刷の新機能」を参照してください。

---

## USB デバイスの新機能

ここでは、この Solaris リリースで拡張された USB デバイスの機能について説明します。

### USB 二重フレームワーク

Solaris 9 12/03 リリースで導入された USB A フレームワークは、USB 1.1 デバイス用に開発されたものです。USB A 1.0 と呼ばれる新しいフレームワークは、USB 2.0 デバイスの高度な要件を満たすために開発されました。このフレームワークでは、USB 1.1 デバイスも使用できます。この Solaris リリースは、「二重フレームワーク」という機

能で、両方のフレームワークに対応しています。二重フレームワークは、従来のフレームワークから新しいフレームワークに簡単に移行するための機能です。従来の USB A フレームワークは、システムの USB 1.1 ポートに接続されたデバイスに対応し、新しい USB A 1.0 フレームワークは、システムの USB 2.0 ポートに接続されたデバイスに対応しています。Sun のマザーボードのポートはすべて USB 1.1 ポートですが、ほとんどの PCI カードのポートは USB 2.0 に対応しています。

USB 二重フレームワークの詳細については、<http://www.sun.com/desktop/whitepapers.html> を参照してください。

## USB フレームワークの互換性に関する問題

一方の USB フレームワーク用に作成されたドライバは、もう一方の USB フレームワークでは機能しません。Sun が提供している USB ドライバのほとんどには、それぞれのフレームワークに対応するバージョンがあります。

USB デバイスをポートに接続しようとする、互換性の問題が発生することがあります。この問題は、ドライバがフレームワークと互換性がないために、そのデバイス用のドライバをフレームワークが認識しないために発生します。フレームワークが接続しようとしたドライバがそのフレームワークと互換性のないドライバである場合には、次のようなメッセージがコンソールに表示されます。

```
The driver for device binding name is not for USB1.0
```

このメッセージは、Sun 以外のドライバ用のデバイスが USB A 1.0 フレームワークと互換性があり、そのデバイスが従来の USB A フレームワークがサポートしているポートに接続されている場合などに表示されます。USB A 1.0 フレームワークはそのデバイスを認識して適切なドライバに関連付けようとはしますが、接続先のフレームワークと互換性がないためにそのドライバは拒否されます。

USB フレームワークの構成を確認する方法については、132 ページの「USB デバイス情報を表示する方法 (prtconf)」を参照してください。

## Solaris がサポートしている USB デバイス

次の表に、Solaris がサポートしている USB 1.1 デバイスおよび USB 2.0 デバイスを示します。

	Solaris 8 HW* リリース	Solaris 9 リリース
<b>USB 1.1</b>	SPARC と x86	SPARC と x86
USB 1.1 オーディオデバイス	USB 2.0 ハブではサポートされない	USB 2.0 ハブではサポートされない



	Solaris 8 HW* リリース	Solaris 9 リリース
<b>USB 2.0</b>	SPARC	SPARC と x86 (Solaris 9 4/04)
USB 2.0 オーディオデバイス	サポートされない	サポートされない
USB 2.0 ストレージデバイス	USB 2.0 ハブでサポートされる	USB 2.0 ハブでサポートされる (Solaris 9 4/04)

\*Solaris 8 リリースではなく、Solaris 8 HW 5/03 以降の Solaris 8 HW リリースです。Solaris 8 HW 5/03 リリースの USB 二重フレームワークのパッチ番号は、109896 です。

注 - Solaris 9 9/04 リリースでのみ、USB 2.0 ポートに接続されている USB 2.0 ハブ上で USB 1.1 デバイスが動作します。

次の表は、Sun SPARC ハードウェアがサポートしている USB の一覧です。

システムの種類	Solaris リリース	サポートしている USB デバイスと速度
Sun Blade 100、150、1000、および 2000	Solaris 9 4/04 リリースより前の Solaris 9 リリース、および Solaris HW 5/03 リリースより前の Solaris 8 リリース	12 MB/秒のすべての USB デバイス
Sun Blade 100、150、1000、および 2000	Solaris 9 4/04、および Solaris 8 HW 5/03	USB 1.1 デバイス、12 MB/秒 (任意の USB ポートに接続した場合) USB 2.0 デバイス、12 MB/秒 (マザーボードのポートに接続した場合) USB 2.0 デバイス、480 MB/秒 (拡張 PCI USB 2.0 カードのポートに接続した場合)
Sun Blade 1500 および 2500	Solaris 9 4/04 および Solaris 8 HW 5/03	USB 1.1 デバイス、12 MB/秒 (任意の USB ポートに接続した場合) USB 2.0 デバイス、12 MB/秒 (マザーボードのポートに接続した場合) USB 2.0 デバイス、480 MB/秒 (PCI コンボカードのポートに接続した場合)

システムの種類	Solaris リリース	サポートしている USB デバイスと速度
その他の Sun SPARC PCI プラットフォーム	Solaris 9 4/04 および Solaris 8 HW 5/03	USB 1.1 デバイス、12 MB/秒  USB 2.0 デバイス、480 MB/秒 (拡張 PCI USB 2.0 カードのポートに接続し た場合)

Solaris リリースで検証済みの PCI カードについては、次のサイトを参照してください。[http://www.sun.com/io\\_technologies/USB.html](http://www.sun.com/io_technologies/USB.html)

USB デバイスをサポートする Sun のプラットフォームは、次のとおりです。

- USB 1.1 をサポートする OHCI ホストコントローラを搭載した次の SPARC システムは、ロースピードおよびフルスピードのデバイスをサポートします。
  - Solaris 8 または 9 リリースが動作する Sun Blade™ システム
  - Solaris 9 リリースが動作する Netra™ X 1/T1 および一部の Sun Fire™ システム
- Sun Blade 1500 や 2500 システムなど、OHCI および EHCI ホストコントローラを搭載した SPARC システムは、ハイスピードの USB 2.0 デバイス、およびロースピードとフルスピードの USB 1.1 デバイスをサポートします。Solaris 8 HW 5/03 リリースまたは Solaris 9 4/04 リリースが動作する PCI ベースの sun4u システムも含まれます (USB 2.0 PCI カードが搭載されている場合)。
- Solaris 8 または Solaris 9 の x86 版で動作し、UHCI ホストコントローラを搭載した x86 ベースのシステムは、USB 1.1 をサポートします。

USB サポートの追加情報については、119 ページの「USB デバイスの概要」を参照してください。

## SPARC: USB 2.0 の機能

この Solaris リリースでは、次の USB 2.0 機能がサポートされます。

- パフォーマンスの向上 – USB 2.0 コントローラに接続されたデバイスの場合、データのスループットが USB 1.1 デバイスと比較して最大で 40 倍速に向上します。

DVD やハードディスクなどの高速大容量ストレージデバイスにアクセスするときに、高速の USB プロトコルを利用できます。
- 互換性 – 1.0 および 1.1 デバイスおよびドライバとの下位互換性があるので、同じケーブル、コネクタ、およびソフトウェアインタフェースを使用できます。

USB デバイスおよび USB 用語については、119 ページの「USB デバイスの概要」を参照してください。

## USB 2.0 デバイスの機能および互換性の問題

USB 2.0 デバイスは、USB 2.0 仕様に準拠した高速デバイスです。USB 2.0 仕様については、<http://www.usb.org> を参照してください。

この Solaris リリースの SPARC システムおよび x86 システムでは、たとえば、次の USB デバイスがサポートされます。

- 大容量ストレージデバイス – CD-RW、ハードディスク、DVD、デジタルカメラ、Zip、フロッピーディスク、およびテープドライブ
- キーボード、マウス、スピーカ、およびマイク
- オーディオデバイス

この Solaris リリースで検証済みの USB デバイスをすべて確認するには、以下のサイトを参照してください。

[http://www.sun.com/io\\_technologies/USB.html](http://www.sun.com/io_technologies/USB.html)

それ以外のストレージデバイスでも、`scsa2usb.conf` ファイルを変更すれば使用できることがあります。詳細については、`scsa2usb(7D)` のマニュアルページを参照してください。

Solaris USB 2.0 デバイスでは、次の機能がサポートされます。

- USB バス速度が、12 Mbps から 480 Mbps に向上しています。このため、USB 2.0 仕様をサポートするデバイスを USB 2.0 ポートに接続すると、対応する USB 1.1 デバイスに比べて速度が大幅に向上します。

USB 2.0 ポートは、SPARC システムでは次のように定義されています。

- USB 2.0 PCI カード上のポート
- USB 2.0 ポートに接続された USB 2.0 ハブ上のポート
- USB 2.0 は、PCI ベースのすべての SPARC プラットフォーム上で Solaris Ready 認証済みです。USB 2.0 ポートを利用するには、USB 2.0 PCI カードが必要です。Solaris リリースで検証済みの USB 2.0 PCI カードについては、[http://www.sun.com/io\\_technologies/USB.html](http://www.sun.com/io_technologies/USB.html) を参照してください。
- 同じシステムで USB 1.1 と USB 2.0 デバイスを併用する場合でも、USB 1.1 デバイスは、従来と同様に動作します。
- USB 2.0 デバイスは USB 1.x ポートでも動作しますが、USB 2.0 ポートに接続するとパフォーマンスが大幅に向上します。
- ほとんどの USB 2.0 ホストコントローラには、1つのハイスピード EHCI (Enhanced Host Controller) と、1つ以上のロースピードまたはフルスピードの OHCI (OpenHCI Host Controller) が組み込まれたコントローラが用意されています。USB 2.0 ポートに接続されたデバイスは、USB 2.0 をサポートするかどうかに応じて、EHCI または OHCI コントローラに動的に割り当てられます。

---

注 - USB 2.0 PCI カード上のポートに接続された USB 2.0 ストレージデバイスのデバイス名は、以前の Solaris リリースで同じハードウェア構成で使用していた場合には、このリリースにアップグレードした後で変更されることがあります。この変更は、アップグレードによりこれらのデバイスが USB 2.0 デバイスとして認識され、制御が EHCI コントローラに引き継がれるために発生します。コントローラの番号 ( /dev/ [r] dsk/cwtxdysz の w) は、それらのデバイスに合わせて変更されます。

---

USB 2.0 デバイスのサポートの詳細については、ehci (7D) および usba (7D) のマニュアルページを参照してください。

## USB 2.0 ケーブル

- サポートされている最大ケーブル長は 5 m です。
- 延長ケーブルを使用しないでください。ケーブルを延長するには、最良の結果が得られるよう、自己電源供給方式のハブを使ってください。
- 詳細については、<http://www.usb.org/channel/training/warning/> を参照してください。

## バス電源供給方式のデバイス

バス電源供給方式のハブは、接続先の USB バスの電源を利用して、接続されているデバイスに電源を供給します。これらのハブの負荷が大きくなりすぎないように、十分に注意してください。これらのハブから接続先のデバイスに供給できる電源は限られているためです。

- バス電源供給方式のハブは、カスケード接続しないでください。たとえば、バス電源供給方式のハブを別のバス電源供給方式のハブに接続しないでください。
- キーボードやマウスなどの低速で消費電力の小さいデバイスを除いて、できるだけバス電源供給方式のデバイスをバス電源供給方式のハブに接続しないでください。ディスク、スピーカ、マイクなどの消費電力の大きいデバイスをバス電源供給方式のハブに接続すると、ハブに接続されたすべてのデバイスで電力が不足することがあります。この場合、デバイスが予期しない動作をすることがあります。

## USB 大容量ストレージデバイス

この Solaris リリースでは、すべての USB ストレージデバイスがリムーバブルメディアデバイスとしてアクセスされます。この変更には、次のような利点があります。

- 標準の MS-DOS または Windows (FAT) ファイルシステムを使用する USB ストレージデバイスがサポートされます。

- `format` コマンドの代わりに、使いやすい `rmformat` コマンドを使用して、すべての USB ストレージデバイスをフォーマットしたり、パーティションを操作したりできます。`format` コマンドの機能が必要な場合は、`format -e` コマンドを使用してください。
- `fdisk` 方式でパーティションを操作する必要がある場合は、`fdisk` コマンドを使用できます。
- ルート権限を必要とする `mount` コマンドが必要なくなったため、ルート以外のユーザーでも USB ストレージデバイスにアクセスできます。デバイスは `vold` コマンドによって自動的にマウントされ、`/rmdisk` ディレクトリの下で利用できます。システムの停止中に新しいデバイスを接続した場合は、`boot -r` コマンドを使用して再構成ブートを実行し、そのデバイスを `vold` に認識させます。`vold` は、ホットプラグ対応のデバイスを自動的に認識しません。システムの稼働中に新しいデバイスを接続した場合は、`vold` を再起動してください。詳細については、`vold(1M)` および `scsa2usb(7D)` のマニュアルページを参照してください。
- FAT ファイルシステムのディスクをマウントしてアクセスできます。たとえば、次のようになります。

```
mount -F pcfs /dev/dsk/c2t0d0s0:c /mnt
```

- LOG SENSE ページをサポートするデバイスを除いて、すべての USB ストレージデバイスの電源が管理されます。LOG SENSE ページを使用するデバイスは通常、USB-SCSI ブリッジデバイスを介して接続する SCSI デバイスです。以前の Solaris リリースでは、一部の USB ストレージデバイスはリムーバブルメディアとして認識されなかったため、電源管理の対象外でした。
- USB 大容量ストレージデバイスでは、アプリケーションの動作が異なることがあります。USB ストレージデバイスでアプリケーションを使用するときには、次の点に考慮してください。
  - 以前のリリースでは、フロッピーディスクや Zip ドライブなどの小容量のデバイスだけがリムーバブルメディアとして認識されていたため、アプリケーションがメディアのサイズを正しく認識しないことがあります。
  - メディアを取り出すことができないデバイス (ハードディスクドライブなど) に対して、メディアを取り出す要求を行うと、要求は成功しますが、何も実行されません。
  - 以前の Solaris リリースの動作が必要な場合、つまり、すべての USB 大容量ストレージがリムーバブルメディアデバイスとして認識されるには限らないようにするには、`/kernel/drv/scsa2usb.conf` ファイルを更新すれば、以前の動作を強制的に適用できます。

USB 大容量ストレージデバイスの使用方法の詳細については、`scsa2usb(7D)` のマニュアルページを参照してください。

## USB 大容量ストレージデバイスの問題の障害追跡

USB 大容量ストレージデバイスを追加または削除するときに問題が発生した場合は、次のヒントを参考にしてください。

- システムの停止中に USB デバイスを追加または削除した場合は、再構成ブートを実行する必要があります。

```
ok boot -r
```

システムの稼働中に接続したデバイスにアクセスするときに、問題が発生した場合は、次のコマンドを実行してください。

```
# devfsadm
```

- 保存停止モードでシステムの電力消費を抑えている場合は、デバイスを移動しないでください。詳細については、125 ページの「SPARC: USB 電源管理」を参照してください。
- アプリケーションがデバイスを使用しているときに、そのデバイスが取り外されて使用できなくなっている場合は、アプリケーションを停止してください。そのデバイスノードが削除されているかどうかを確認するには、`prtconf` コマンドを使用します。

## SPARC: USB ドライバの機能拡張

この節では、この Solaris リリースで拡張された USB ドライバの機能について説明します。

- 新しい汎用 **USB** ドライバ - UNIX® 標準の `read(2)` および `write(2)` システムコールを使って、アプリケーションから USB デバイスのアクセスおよび操作を実行できるようになりました。特別なカーネルドライバを記述する必要はありません。次の機能が追加されています。
  - アプリケーションから `raw` デバイスのデータおよびステータスにアクセスできます。
  - 制御転送、バルク転送、および割り込み (入力と出力) 転送がサポートされます。

詳細については、`ugen(7D)` のマニュアルページ、および次の Web サイトで USB DDK を参照してください。

<http://developers.sun.com/solaris/developer/support/driver/usb.html>

- **Digi Edgeport USB** のサポート - シリアルポート変換デバイス用に複数の Digi Edgeport のサポートを提供します。
  - 新しいデバイスは、`/dev/term/[0-9]*` および `/dev/cua/[0-9]*` としてアクセスします。
  - USB シリアルポートは、ローカルシリアルコンソールとして使用できない点を除き、ほかのシリアルポートと同様に使用できます。USB シリアルポートのデータが USB ポートを使用して転送されますが、このことをユーザーは意識する必要はありません。

詳細については、`usbser_edge(7D)` を参照するか、<http://www.digi.com> および <http://www.sun.com/io> を参照してください。

- ユーザーが記述したカーネルおよびユーザー独自のドライバの文書サポートおよびバイナリサポート – Solaris USB Driver Development Kit (DDK) を利用できます。DDK に関する情報など、USB ドライバの開発に関する最新の情報については、次の Web サイトを参照してください。  
<http://developers.sun.com/solaris/developer/support/driver/usb.html>

## EHCI ドライバおよび OHCI ドライバ

EHCI ドライバには、次の特徴があります。

- USB 2.0 をサポートする拡張ホストコントローラインタフェースに準拠しています。
- 高速の制御転送、バルク転送、および割り込み転送をサポートします。
- 現在、高速のアイソクロナス (isochronous、等時性) トランザクションはサポートされていません。たとえば、USB 2.0 オーディオデバイスはサポートされていません。

システムに USB 2.0 および USB 1.x デバイスが実装されている場合には、EHCI ドライバおよび OHCI ドライバは、システムに接続されているデバイスの種類に応じてデバイス制御を渡します。

- USB 2.0 PCI カードには、1 つの EHCI コントローラおよび 1 つ以上の OHCI コントローラが組み込まれています。
- USB 1.1 デバイスを接続すると、OHCI コントローラに動的に割り当てられます。USB 2.0 デバイスを接続すると、EHCI コントローラに動的に割り当てられます。

---

## USB デバイスの概要

Universal Serial Bus (USB) は PC 業界で開発された、周辺機器 (キーボード、マウス、プリンタなど) をシステムに接続するための低コストのソリューションです。

USB コネクタは 1 方向 1 種類のケーブルだけに適合するように設計されています。USB が設計された主な目的は、デバイスごとに異なる何種類ものコネクタを減らすことです。USB の設計により、システムの背面パネルの混雑を軽減できます。

デバイスは、外部 USB ハブ上の USB ポートか、コンピュータ本体に設置されたルートハブ上の USB ポートのいずれかに接続されます。ハブには複数のポートがあるため、1 つのハブからデバイスツリーの複数の枝が伸びることがあります。

次の表に、Solaris 環境でサポートされる USB デバイスを示します。

USB デバイス	USB デバイスがサポートされるシステム
オーディオデバイス上の HID コントロール	SPARC および x86 システム
ハブ	SPARC および x86 システム
キーボードとマウス	SPARC および x86 システム
大容量ストレージ	SPARC および x86 システム  サポートされる構成では、キーボードとマウスは1つずつしか含めることができない。これらのデバイスは、オンボードの USB ホストコントローラに接続する必要がある
プリンタ	SPARC および x86 システム
スピーカおよびマイク	SPARC および x86 システム
USB シリアルコンバータ	SPARC システム

## よく使用される USB 関連の略語

次の表に、Solaris 環境で使用される USB の略語について説明します。USB の構成要素と略語についての詳細は <http://www.usb.org> を参照してください。

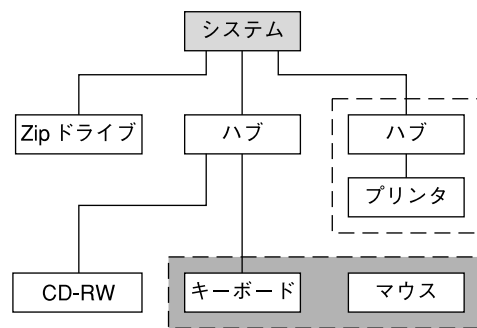
略語	定義
ugen	USB 汎用ドライバ
USB	Universal Serial Bus
USBA	USB アーキテクチャ (Solaris)
USBAI	USBA クライアントドライバインタフェース (Solaris)
HCD	USB ホストコントロールドライバ
EHCI	拡張オープンコントローラインタフェース
OHCI	オープンホストコントローラインタフェース
UHCI	ユニバーサルホストコントローラインタフェース

## USB バスの説明

USB 仕様は、ライセンス料を払わずに入手することができます。USB 仕様は、バスとコネクタの電気的および機械的なインタフェースを定義します。



USB が採用するトポロジでは、ハブが USB デバイスに接続点を提供します。ホストコントローラには、システム内のすべての USB ポートの起点となるルートハブが含まれます。ハブの詳細については、124 ページの「USB ホストコントローラとルートハブ」を参照してください。



- USB ホストコントローラおよびルートハブ
- ⌋ 合成デバイス
- 複合デバイス

図 7-1 USB 物理デバイスの階層

図 7-1 は、有効な USB ポートが 3 つ搭載されたシステムを示しています。1 番目の USB ポートは Zip ドライブに接続されています。2 番目の USB ポートは外部ハブに接続されており、このハブには CD-RW デバイスと、キーボードとマウスの複合デバイスが接続されています。このキーボードは「複合デバイス」であるため USB コントローラが組み込まれており、このコントローラによって、キーボードとキーボードに接続されたマウスの両方が制御されます。キーボードとマウスは、同じ USB コントローラによって制御されるため、同一の USB バスアドレスを共有します。

図 7-1 は、ハブとプリンタの「合成デバイス」の例も示しています。このハブは外部ハブで、プリンタと同じケースに入っています。プリンタはこのハブに固定接続されます。このハブとプリンタは、それぞれ異なる USB バスアドレスを持ちます。

次の表に、図 7-1 に示したデバイスの一部について、デバイスツリーパス名を示します。

Zip ドライブ	/pci@1f,4000/usb@5/storage@1
キーボード	/pci@1f,4000/usb@5/hub@2/device@1/keyboard@0
マウス	/pci@1f,4000/usb@5/hub@2/device@1/mouse@1
CD-RW デバイス	/pci@1f,4000/usb@5/hub@2/storage@3
プリンタ	/pci@1f,4000/usb@5/hub@3/printer@1

## USB デバイスとドライバ

属性とサービスが似ている USB デバイスは、いくつかのデバイスクラスに分類されます。デバイスクラスごとに 1 つのドライバが割り当てられます (フレームワークごとに 1 つ)。1 つのクラス内のデバイスは、同じデバイスドライバの組み合わせで管理されます。ただし、USB 仕様では、特定のクラスに属さないベンダー固有のデバイスも許可しています。

Human Interface Device (HID) クラスには、ユーザーが制御するデバイス (キーボード、マウス、ジョイスティックなど) が含まれます。Communication Device クラスには、電話に接続するデバイス (モデムや ISDN インタフェースなど) が含まれます。その他にも、Audio Device、Monitor Device、Printer Device、Storage Device などのデバイスクラスがあります。各 USB デバイスはデバイスのクラスを表す記述子を持っています。デバイスクラスは、そのメンバーが構成とデータ転送についてどのように動作するかを指定します。クラス情報の詳細については、<http://www.usb.org> を参照してください。

## Solaris USB アーキテクチャ (USBA)

USB デバイスは、2 つのレベルのデバイスツリーノードとして表現できます。デバイスノードは、USB デバイス全体を表します。1 つまたは複数の子インタフェースノードはデバイス上にある個々の USB インタフェースを表します。

ドライバのバインドは互換性のある名前属性の使用によって実現されます。詳細については、『*IEEE 1275 USB binding* (英語版)』の 3.2.2.1 項と『*Writing Device Drivers*』を参照してください。ドライバは、デバイス全体にバインドしてすべてのインタフェースを制御することも、1 つのインタフェースだけにバインドすることも可能です。デバイス全体にバインドするドライバがベンダーにもクラスにも存在しない場合、汎用 USB マルチインタフェースドライバがデバイスレベルのノードにバインドされます。IEEE 1275 バインド仕様の 3.2.2.1 項で定義されているように、このドライバは互換名プロパティを使用して、各インタフェースに対してドライバのバインドを試みます。

Solaris USB アーキテクチャ (USBA) は、USB 1.1 および USB 2.0 の仕様に加え、Solaris ドライバ条件にも準拠しています。USBA モデルは Sun Common SCSI Architecture (SCSA) に似ています。USBA は、汎用 USB トランスポート層という概念をクライアントドライバに提供する薄い層で、汎用 USB の主要な機能を実装するサービスをクライアントドライバに提供します。



図 7-2 Solaris USB アーキテクチャ (USBA)

---

## Solaris 環境における USB について

次の節では、Solaris 環境における USB について知っておく必要のある情報を説明します。

### USB キーボードとマウス

サポートされているのは、Sun USB キーボードとマウスデバイスだけです。また、複数の USB キーボードとマウスデバイスを含むシステム構成は場合によっては機能しますが、Solaris 環境ではサポートされていません。詳細については、次の説明を参照してください。

- USB キーボードおよびマウスは、バス上の任意の場所に接続して、コンソールキーボードおよびマウスとして構成することができます。キーボードとマウスが外部ハブに接続されている場合は、システムのブートが遅くなります。
- リブート中または ok プロンプトの出ている間は、コンソールキーボードおよびマウスを移動しないでください。システムのリブート後であれば、いつでもコンソールキーボードおよびマウスを別のハブに移動することができます。キーボードおよびマウスは、差し込んだ後は再び完全に機能します。
- SPARC のみ - USB キーボードの電源キーと Sun タイプ 5 キーボードの電源キーの動作は異なります。USB キーボードでは、「SUSPEND/SHUTDOWN」キーを使用してシステムを中断またはシャットダウンすることができますが、そのキーを使用してシステムの電源を入れることはできません。
- Sun 社製以外の USB キーボードでは、キーパッドの左側にある機能は使用できません。

- 複数のキーボードはサポートされません。
  - 複数のキーボードは認識され、使用できますが、コンソールキーボードとしては認識されません。
  - ブート時に最初に認識されたキーボードがコンソールキーボードとなります。このため、複数のキーボードが接続されていると、ブート時に混乱の原因となります。
  - コンソールキーボードを取り外しても、次に利用可能な USB キーボードはコンソールキーボードにはなりません。次にホットプラグされるキーボードがコンソールキーボードになります。
- 複数のマウスはサポートされません。
  - 複数のマウスは認識され、使用できますが、コンソールマウスとしては認識されません。
  - ブート時に最初に認識されたマウスがコンソールマウスとなります。このため、複数のマウスが差し込まれていると、ブート時に混乱の原因となります。
  - コンソールマウスを取り外しても、次に利用可能な USB マウスはコンソールマウスにはなりません。次にホットプラグされるマウスがコンソールマウスになります。
- Sun 社製以外の、キーボードとマウスを組み合わせたデバイスを使用する場合、このキーボードがブート時に最初に認識されると、PS/2 マウスが差し込まれていなくても、このキーボードとマウスがコンソールキーボードとマウスになります。このため、別の USB マウスがシステムに差し込まれていても、コンソールマウスとして構成されないので機能しません。
- USB マウスまたは PS/2 マウスデバイスでは、3 ボタン以上の使用がサポートされています。
- USB マウスまたは PS/2 マウスデバイスでは、ホイール付きマウスによるスクロールが使用可能です。つまり、USB マウスまたは PS/2 マウスでホイールを回転させると、マウスが置かれたアプリケーションまたはウィンドウがスクロールします。StarSuite™、Mozilla™、および GNOME アプリケーションはホイール付きマウスによるスクロールをサポートします。ただし、ホイール付きマウスによるスクロールをサポートしないアプリケーションもあります。

## USB ホストコントローラとルートハブ

USB ハブは次のことを行います。

- ポートにおけるデバイスの取り付けと取り外しの監視
- ポートにおける個々のデバイスの電源管理
- ポートへの電源の制御

USB ホストコントローラは「ルートハブ」という埋め込みハブを持っています。システムの背面パネルに見えるポートはルートハブのポートです。USB ホストコントローラは次のことを行います。

- USB バスの管理。個々のデバイスはバスの調整はできません。
- デバイスによって決定されるポーリング間隔による、デバイスのポーリング。ポーリング間隔 (時間) を考慮してデバイスに十分なバッファがあることを前提とします。
- USB ホストコントローラとそれに接続されているデバイス間でのデータの送信。ピアツーピア通信はサポートされません。

## USB ハブデバイス

- SPARC システムと x86 システムのどちらにおいても、ハブを 4 段を超えて多段接続しないでください。SPARC システムでは、OpenBoot™ PROM (OBP) は 4 段を超えるデバイスを正確に認識できません。
- バス電源供給方式のハブ同士をカスケード接続しないでください。バス電源供給方式のハブは独自の電源を持っていません。
- 大量の電源を必要とするデバイスをバス電源供給方式のハブに接続しないでください。これらのデバイスがバス電源供給方式のハブ上で正しく機能しなかったり、他のデバイス用の電源がなくなったりする可能性があります。このようなデバイスとして、USB フロッピーディスクデバイスなどがあります。
- この Solaris リリースでは、ロースピードまたはフルスピードのデバイスを SPARC システムの USB 2.0 ポートに接続された USB 2.0 ハブに接続することはできません。

## SPARC: USB 電源管理

SPARC システムでは、USB デバイスの保存停止および復元再開機能が完全にサポートされます。ただし、稼働中のデバイスを保存停止したり、システムの保存停止で電源がオフになっているときにデバイスを取り外すことは決してしないでください。

SPARC システムで電源管理を有効にしている場合、USB のフレームワークはすべてのデバイスの電源管理を最大限に試みます。USB デバイスの電源管理により、ハブドライバはデバイスが接続されているポートの中断も行います。「リモートウェイクアップ」をサポートするデバイスは、そのデバイスが利用可能な状態になるように、そのデバイスのバス上にあるすべてのデバイス呼び起こすようシステムに通知できます。アプリケーションがデバイスに入出力を送信した場合も、ホストシステムはデバイスを呼び起こすことができます。

リモートウェイクアップ機能がサポートされている場合、すべての HID (キーボード、マウス、スピーカ、マイクなど)、ハブ、およびストレージデバイスは、デフォルトで電源管理されます。USB プリンタが電源管理されるのは、2 つの印刷ジョブ間だけです。汎用の USB ドライバ (UGEN) に接続しているデバイスの電源は、デバイスが閉じているときのみ管理されます。

電源消費を減らすために電源管理を行なっている場合は、まず USB 末端デバイスの電源が切断されます。また、ハブのポートに接続されているすべてのデバイスの電源が切断されると、しばらくしてからハブの電源が切断されます。もっとも効率的に電源管理をするためには、あまり多くのハブをカスケード接続しないでください。

## USB ケーブルに関するガイドライン

USB ケーブルを接続する際には、以下のガイドラインに従ってください。

- USB 2.0 デバイスを接続するときは、必ず USB 2.0 に準拠したフルレート(480M ビット/秒)の 20/28 AWG ケーブルを使用してください。
- USB 1.0 または 1.1 デバイスを接続するときは、必ず USB 1.0 に準拠したフルレート (12M ビット/秒)の 20/28 AWG ケーブルを使用してください。また、バス電源供給方式のハブは低速のデバイスだけに使用してください。USB デバイスを接続するときは、必ずフルレート (12M ビット/秒)の 20/28 AWG ケーブルを使用してください。
- サポートされている最長ケーブル長は 5 m です。
- 延長ケーブルを使用しないでください。ケーブルを延長するには、最良の結果が得られるよう、自己電源供給方式のハブを使ってください。

詳細については、<http://www.usb.org/channel/training/warning> を参照してください。

## 第 8 章

# USB デバイスの使用 (手順)

この章では、Solaris 環境で USB デバイスを使用する手順について説明します。

USB デバイスの使用手順については、次の節を参照してください。

- 127 ページの「Solaris 環境での USB デバイスの管理 (作業マップ)」
- 128 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」
- 139 ページの「USB デバイスのホットプラグ (作業マップ)」
- 143 ページの「USB オーディオデバイスの使用 (作業マップ)」
- 148 ページの「cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)」

USB デバイスの使用に関する概要については、第 7 章を参照してください。

## Solaris 環境での USB デバイスの管理 (作業マップ)

Solaris 環境でのすべての USB デバイス管理作業を次のマップに一覧表示します。作業ごとにいくつかの詳細タスク (USB デバイスの使用、USB デバイスのホットプラグ、USB オーディオデバイスの追加など) を説明しています。

作業	説明	参照先
USB デバイスを使用する	ファイルシステムを作成およびマウントする前に、USB デバイスをフォーマットする必要がある	128 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」

作業	説明	参照先
USB デバイスをホットプラグする	USB デバイスをシステムに動的に追加し、システムから動的に取り外す	
	USB デバイスを物理的にシステムに追加し、物理的にシステムから取り外すことができる	139 ページの「USB デバイスのホットプラグ (作業マップ)」
	cfgadm コマンドを使用して、USB デバイスをシステムに物理的または論理的に追加し、システムから物理的または論理的に取り外す	148 ページの「cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)」
USB オーディオデバイスを追加する	この作業マップを使用して、USB オーディオデバイスの追加に関連する作業を特定する	143 ページの「USB オーディオデバイスの使用 (作業マップ)」

---

## USB 大容量ストレージデバイスの使用 (作業マップ)

作業	説明	参照先
USB 大容量ストレージデバイスを使用できるように準備する	vdvd を使用して USB 大容量ストレージデバイスを使用できるように準備する	130 ページの「vdvd を使用して USB 大容量ストレージデバイスを使用できるように準備する方法」
	vdvd を使用しないで USB 大容量ストレージデバイスを使用できるように準備する	131 ページの「vdvd を使用しないで USB 大容量ストレージデバイスを使用できるように準備する方法」
USB デバイス情報を表示する	prtconf コマンドを使用して USB デバイス情報を表示する	132 ページの「USB デバイス情報を表示する方法 (prtconf)」
USB 大容量ストレージデバイスをフォーマットする	USB 大容量ストレージデバイスをフォーマットして、データを格納できるようにする	133 ページの「vdvd を使用しないで USB 大容量ストレージデバイスをフォーマットする方法」



作業	説明	参照先
USB 大容量ストレージデバイスをマウントする	vol1d を使用して USB 大容量ストレージデバイスをマウントする	135 ページの「vol1d を使用して USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」
	vol1d を使用しないで USB 大容量ストレージデバイスをマウントする	136 ページの「vol1d を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」
(省略可能) USB デバイスドライバを無効にする	システムで USB サポートを必要としない場合は、USB デバイスドライバを無効にする	
(省略可能) 未使用の USB デバイスのリンクを削除する	devfsadm コマンドを使って、未使用の USB デバイスのリンクを削除する	138 ページの「使用されていない USB デバイスのリンクを削除する方法」

## USB 大容量ストレージデバイスの使用

注 - この Solaris リリースで USB 大容量ストレージデバイスを使用する場合の最新情報については、116 ページの「USB 大容量ストレージデバイス」を参照してください。

Solaris 9 4/04 から、USB の CD-RW、ハードディスク、DVD、デジタルカメラ、Zip、Peerless、SmartMedia、CompactFlash、ORB、および USB フロッピーディスクデバイスなどのリムーバブル大容量ストレージデバイスがサポートされるようになりました。

Solaris 環境でサポートされるすべての USB デバイスを確認するには、[http://www.sun.com/io\\_technologies/USB.html](http://www.sun.com/io_technologies/USB.html) を参照してください。

これらのデバイスは、ボリューム管理を実行している場合でも実行していない場合でも管理することができます。ボリューム管理を実行している場合のデバイス管理については、vol1d(1M) のマニュアルページを参照してください。

## USB フロッピーディスクデバイスの使用

USB フロッピーディスクデバイスは、ほかの USB デバイスと同様に、リムーバブルメディアデバイスとして表示されます。USB フロッピーディスクデバイスは、fd (フロッピー) ドライバでは管理されません。アプリケーションから fd (ネイティブフロッピー) ドライバに対する ioctl(2) 呼び出しを実行すると、そのアプリケーションは失敗します。read(2) および write(2) 呼び出しのみを実行するアプリケーションは成功します。SunPCI や rmformat などのアプリケーションも成功します。

---

注 - 現時点では、CDE のファイルマネージャは USB フロッピーディスクを完全にはサポートしていません。ただし、UFS ファイルシステムを含むフロッピーディスクは、ファイルマネージャのリムーバブルメディアマネージャから開いたり、名前を変更したり、フォーマットしたりすることはできます。PCFS ファイルシステムを含むフロッピーディスクを開くには、リムーバブルメディアマネージャを使用する必要があります。フロッピーディスクがいずれかの種類のファイルシステムを含む場合には、フロッピーディスクとファイルマネージャ間でファイルをドラッグ&ドロップできます。

---

ボリューム管理 (vold) では、USB フロッピーディスクデバイスは SCSI リムーバブルメディアデバイスとして認識されます。ボリューム管理を使用して、/rmdisk ディレクトリの USB フロッピーディスクデバイスにアクセスできます。

USB フロッピーディスクデバイスの使用方法の詳細については、第 1 章を参照してください。

## 非標準拠 USB 大容量ストレージデバイスの使用

デバイスの識別が間違っていたり、USB 大容量ストレージクラスと互換性がないとされているドライバでも、USB 大容量ストレージドライバをサポートする場合があります。scsa2usb.conf ファイルには、大容量デバイスに対応するかどうかを示す、ベンダー ID、製品 ID、およびバージョンを一覧表示した属性オーバーライドリストが含まれています。またデフォルトのデバイス属性をオーバーライドするフィールドも含まれています。このリストのエントリにはデフォルトでコメントが表示されています。特定のデバイスのサポートを有効にするために、コメントをコピーしたり削除したりすることもできます。

この Solaris リリースで動作するシステムに USB 大容量ストレージデバイスを接続したが、システムでそのデバイスを使用できない場合、/kernel/drv/scsa2usb.conf ファイルでこのデバイスに一致するコメントエントリがあるかどうか確認できます。scsa2usb.conf ファイルの指示に従って、オーバーライド情報を使用して、特定のデバイスをサポートできるかどうか確認してください。推奨される USB 大容量ストレージデバイスについては、[http://www.sun.com/io\\_technologies/USB.html](http://www.sun.com/io_technologies/USB.html) を参照してください。

詳細については、scsa2usb(7D) のマニュアルページを参照してください。

## vold を使用して USB 大容量ストレージデバイスを使用できるように準備する方法

Solaris 共通デスクトップ環境 (CDE) が実行されている場合は、USB リムーバブル大容量ストレージデバイスは、CDE ファイルマネージャのコンポーネントであるリムーバブルメディア・マネージャによって管理されます。CDE ファイルマネージャの詳細については、dtfile(1) のマニュアルページを参照してください。

---

注 – この節に記載されているマニュアルページを表示するには、MANPATH 変数に /usr/dt/man ディレクトリを含める必要があります。これらのコマンドを使用するには、パスに /usr/dt/bin ディレクトリを含め、さらに CDE を実行している必要があります。また、これらのコマンドをリモートで使用する場合は、DISPLAY 変数を設定する必要があります。

---

次の表に、リムーバブルメディア・マネージャが CDE 環境からストレージデバイスを管理する際に使用するコマンドを示します。

コマンド	マニュアルページ	作業
sdtmedia_format	sdtmedia_format(1)	デバイスのフォーマットおよびラベル付けを行う
sdtmedia_prop	sdtmedia_prop(1)	デバイスのプロパティを表示する
sdtmedia_prot	sdtmedia_prot(1)	デバイスのアクセス権を変更する
sdtmedia_slice	sdtmedia_slice(1)	デバイス上のスライスを作成または変更する

USB デバイスのフォーマットが終了すると、通常は /rmdisk/label ディレクトリの下にマウントされます。リムーバブルストレージデバイスの構成方法については、rmmount.conf(4) または vold.conf(4) のマニュアルページを参照してください。

デバイスノードは /vol/dev ディレクトリの下に作成されます。詳細については、scsa2usb(7D) のマニュアルページを参照してください。

次の手順は、vold を使用しないで USB 大容量ストレージデバイスを管理する方法を示しています。デバイスノードは、文字型デバイスについては /dev/rdisk ディレクトリ、ブロック型デバイスについては /dev/dsk ディレクトリの下に作成されます。デバイスリンクは、デバイスをホットプラグしたときに作成されます。詳細については、scsa2usb(7D) のマニュアルページを参照してください。

## ▼ vold を使用しないで USB 大容量ストレージデバイスを使用できるように準備する方法

USB 大容量ストレージデバイスは、ボリューム管理 (vold) を使用しないで準備することもできます。次のコマンドを実行して vold を停止します。

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

または、次の手順を実行して、vold を実行します。この場合、USB 大容量ストレージデバイスは vold には登録されません。

- 手順 1. スーパーユーザーになります。
2. 以下に示すように `/etc/vold.conf` ファイル内の次の行で `#` マークを挿入してコメント扱いにし、**USB** 大容量ストレージデバイスのボリュームマネージャの登録を削除します。

```
# use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

3. 上記のようにコメントアウトした後、**vold** を再起動します。

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



---

注意 - この行をコメントにすると、そのシステム内に他の SCSI、ATAPI Zip、Peerless などのリムーバブルデバイスがある場合、それらのデバイスの **vold** の登録も無効になります。

---

詳細については、`vold.conf(4)` のマニュアルページを参照してください。

## USB デバイス情報を表示する方法 (`prtconf`)

`prtconf` コマンドを使用して USB デバイス情報を表示します。

```
$ prtconf
usb, instance #0
  hub, instance #2
    device, instance #8
      interface (driver not attached)
    printer (driver not attached)
  mouse, instance #14
    device, instance #9
      keyboard, instance #15
      mouse, instance #16
  storage, instance #7
    disk (driver not attached)
  communications, instance #10
    modem (driver not attached)
    data (driver not attached)
storage, instance #0
  disk (driver not attached)
storage, instance #1
  disk (driver not attached)
```

`prtconf` コマンドの `-D` オプションを使用すれば、追加ドライバ情報を表示できます。この情報を使用して、SPARC システム上の USB A 1.0 フレームワークが使用しているポートおよびデバイスを確認できます。たとえば、次のように表示されます。

```
$ prtconf -D
.
```

```

.
SUNW,Sun-Blade-1500
.
.
1 pci, instance #0 (driver name: pcisch)
   isa, instance #0 (driver name: ebus)
.
.
.
2 usb, instance #0 (driver name: ohci)
   usb, instance #1 (driver name: ohci)
.
.
.
3 pci, instance #0 (driver name: pci_pci)
4 usb, instance #0 (driver name: usba10_ohci)
   usb, instance #1 (driver name: usba10_ohci)
   usb, instance #0 (driver name: usba10_ehci)
   storage, instance #9 (driver name: usba10_scsa2usb)
   disk, instance #9 (driver name: usb_sd)
   firewire, instance #0 (driver name: hci1394)
.
.
.

```

上の出力から、次の構成の特徴を確認できます。

- PCI カードポートは、usb ノードの上に出力されている一連の pci ノード番号からわかります。  
PCI カードポート (4) は、pci ノード 1 およびノード 3 という 2 つの階層になっています。これらのポートは、マザーボードと PCI カードの両方が使用するためです。オンボードポート (2) は、1 つの PCI ノード (1) として出力されています。これらのポートは、メインシステムバスに近い 1 つのハードウェアアーキテクチャ層であるためです。
- デバイスノードに関連付けられたドライバの名前は、デバイスおよびそのデバイスが割り当てられているポートを使用しているフレームワークを示しています。(4) のうち、usba10 で始まるすべての USB インスタンスのドライバは、USB A 1.0 フレームワークがこれらのポートおよびポートに割り当てられているデバイスを管理していることを示しています。これらのポートだけが、ハイスピード USB 2.0 デバイスをサポートしています。

## ▼ vold を使用しないで USB 大容量ストレージデバイスをフォーマットする方法

USB 大容量ストレージデバイスは、Solaris オペレーティングシステムが使用する他のすべてのデバイスと同様に、使用する前にフォーマットしてファイルシステムを作成する必要があります。USB 大容量ストレージデバイス (フロッピーディスクも含む) は、PCFS および UFS の両方のファイルシステムをサポートします。PCFS または UFS ファイルシステムを作成する前に、ディスクがフォーマットされていることを確認してください。

- 手順
1. **vold**を無効にする方法については、131 ページの「**vold** を使用しないで **USB** 大容量ストレージデバイスを使用できるように準備する方法」を参照してください。
  2. (省略可能) **USB** フロッピーディスクデバイスをシステムに追加します。  
USB デバイスをホットプラグする方法については、以下を参照してください。
    - 143 ページの「**USB** オーディオデバイスの使用 (作業マップ)」
    - 148 ページの「**cfgadm** コマンドを使用した **USB** デバイスのホットプラグ (作業マップ)」

3. (省略可能) フロッピーディスクデバイスを特定します。  
たとえば、次のようになります。

```
# cd /dev/rdisk
# devfsadm -C
# ls -l c*0 | grep usb
lrwxrwxrwx  1 root  root   55 Mar  5 10:35 c2t0d0s0 ->
../../../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
この例では、フロッピーディスクデバイスは c2t0d0s0 です。
```

4. フロッピーディスクをフロッピーディスクドライブに挿入します。
5. フロッピーディスクをフォーマットします。

```
% rmformat -Flong raw-device
たとえば、次のようになります。
```

```
% rmformat -Flong /dev/rdisk/c2t0d0s0
```

6. ファイルシステムの種類を決定し、次のいずれかの作業を選択します。
  - a. **PCFS** ファイルシステムを作成します。

```
# mkfs -F pcfs -o nofdisk,size=size raw-device
-size オプションを 512 バイトブロック単位で指定します。
```

次の例は、1.4M バイトのフロッピーディスクに **PCFS** ファイルシステムを作成する方法を示しています。

```
# mkfs -F pcfs -o nofdisk,size=2880 /dev/rdisk/c4t0d0s0
```

次の例は、100M バイトの Zip ドライブに **UFS** ファイルシステムを作成する方法を示しています。

```
# mkfs -F pcfs -o nofdisk,size=204800 /dev/rdisk/c5t0d0s0
```

このコマンドの実行には、数分かかることがあります。

- b. **UFS** ファイルシステムを作成します。

```
# newfs raw-device
```

たとえば、次のようになります。

```
# newfs /dev/rdisk/c4t0d0s0
```

---

注 - フロッピーディスクの記憶容量は少ないので、UFS ファイルシステムのためにフロッピーディスクのかかなりの容量が消費されます。

---

## ▼ vold を使用して USB 大容量ストレージデバイスをマウントまたはマウント解除する方法

- 手順 1. すべてのリムーバブル大容量ストレージデバイス (**USB 大容量ストレージデバイス**を含む) について、デバイスの別名を表示します。

```
$ eject -n
.
.
.
cdrom0 -> /vol/dev/rdisk/c0t6d0/audio_cd      (Generic CD device)
zip0 -> /vol/dev/rdisk/c1t0d0/zip100         (USB Zip device)
zip1 -> /vol/dev/rdisk/c2t0d0/fat32         (USB Zip device)
rmdisk0 -> /vol/dev/rdisk/c5t0d0/unnamed_rmdisk (Peerless, HD or floppy)
rmdisk1 -> /vol/dev/rdisk/c4t0d0/clik40      (Generic USB storage)
```

2. 次のいずれかの方法を選択して、**USB 大容量ストレージデバイス**をマウントまたはマウント解除します。

- a. 上記のようにリストされたデバイス別名を使用して、**USB 大容量ストレージデバイス**をマウントします。

```
$ volrmount -i device-alias
```

次の例では、USB Zip ドライブ (/rmdisk/zip0) をマウントします。

```
$ volrmount -i zip0
```

- b. **USB 大容量ストレージデバイス**をマウント解除します。

```
$ volrmount -e device-alias
```

次の例では、USB Zip ドライブ (/rmdisk/zip0) をマウント解除します。

```
$ volrmount -e zip0
```

3. **USB デバイス**を、汎用の **USB ドライブ**から取り出します。

```
$ eject device-alias
```

たとえば、次のようになります。

```
$ eject rmdisk0
```

---

注 - デバイスのマウントが解除されていない場合、`eject` コマンドは、そのデバイスのマウント解除も行います。また、そのデバイスにアクセスするアプリケーションが実行されている場合は、そのアプリケーションを終了させます。

---

## ▼ `vold` を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法

- 手順
1. `vold` を無効にする方法については、131 ページの「`vold` を使用しないで USB 大容量ストレージデバイスを使用できるように準備する方法」を参照してください。
  2. スーパーユーザーになります。
  3. (省略可能) フロッピーディスクデバイスを特定します。  
たとえば、次のようになります。

```
# cd /dev/rdisk
# devfsadm -C
# ls -l c*0 | grep usb
lrwxrwxrwx  1 root  root   55 Mar  5 10:35 c2t0d0s0 ->
../../../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
この例では、フロッピーディスクデバイスは c2t0d0s0 です。
```

4. 次のいずれかの方法を選択して、USB 大容量ストレージデバイスをマウントまたはマウント解除します。
  - a. USB 大容量ストレージデバイスをマウントします。

```
# mount [ -F fstype ] block-device mount-point
```

次の例は、UFS ファイルシステムを使用しているデバイスのマウント方法を示しています。

```
# mount /dev/dsk/c1t0d0s2 /mnt
```

次の例は、PCFS ファイルシステムを使用しているデバイスのマウント方法を示しています。

```
# mount -F pcfs /dev/dsk/c1t0d0s0:c /mnt
```

次の例は、読み取り専用の HSFS ファイルシステムを使用している CD のマウント方法を示しています。

```
# mount -F hsfs -o ro /dev/dsk/c1t0d0s2 /mnt
```

- b. USB 大容量ストレージデバイスをマウント解除します。  
デバイス上のファイルシステムを使用しているユーザーがいないことを最初に確認してください。  
たとえば、次のようになります。



```
# fuser -c -u /mnt
# umount /mnt
```

c. デバイスを取り出します。

```
# eject /dev/[r]dsk/cntndnsn
たとえば、次のようになります。

# eject /dev/rdsk/c1t0d0s2
```

## 特定の USB ドライバを無効にする

特定の種類の USB デバイスを無効にするには、対応するクライアントドライバを無効にします。たとえば、USB プリンタを無効にするには、そのプリンタを使用している `usbprn` ドライバを無効にします。`usbprn` を無効にしても、USB ストレージデバイスなどの他のデバイスには影響しません。

両方のフレームワークでデバイスタイプを無効にする必要があります。一方のフレームワークだけでデバイスタイプを無効にすることはできません。次の表に、USB デバイスタイプとそれに対応するドライバを示します。

デバイスの種類	無効にするドライバ
オーディオ	<code>usb_ac</code> および <code>usb_as</code>
HID (通常はキーボードとマウス)	<code>hid</code>
ストレージ	<code>scsa2usb</code>
プリンタ	<code>usbprn</code>
シリアル	<code>usbser_edge</code>

システムに接続されている USB デバイスのドライバを無効にすると、次のようなコンソールメッセージが表示されます。

```
usba10: WARNING: usba: no driver found for device name
```

## ▼ 特定の USB ドライバを無効にする方法

手順 1. スーパーユーザーになります。

2. 削除するドライバの別名を記録します。

```
# cp /etc/driver_aliases /etc/driver_aliases.orig
```

3. 無効にする USB ドライバの別名を特定します。

たとえば、次のようになります。

```
# grep usbprn /etc/driver_aliases
usbprn "usbif,class7.1.1"
usbprn "usbif,class7.1.2"
```

4. ドライバの別名エントリを削除します。  
たとえば、次のようになります。

```
# update_drv -d -i "usbif,class7.1.1" usbprn
# update_drv -d -i "usbif,class7.1.2" usbprn
```

5. システムをリブートします。

```
# init 6
```

## ▼ 使用されていない USB デバイスのリンクを削除する方法

システムの電源がオフのときに USB デバイスを取り外した場合には、次の手順を実行します。システムの電源が切断されているときに USB デバイスを取り外すと、存在しないデバイスへのリンクが残る場合があります。

- 手順
1. スーパーユーザーになります。
  2. そのデバイスにアクセスする可能性のあるアプリケーションをすべて閉じます。
  3. 特定の **USB** クラスの未使用のリンクを削除します。  
たとえば、次のようになります。

```
# devfsadm -C -c audio
または、関連するリンクをすべて削除します。
# devfsadm -C
```

---

## USB デバイスのホットプラグ (作業マップ)

作業	説明	参照先
USB 大容量ストレージデバイスを追加する	void を使用して USB 大容量ストレージデバイスを取り付ける	140 ページの「void を使用して USB 大容量ストレージデバイスを追加する方法」
	void を使用しないで USB 大容量ストレージデバイスを取り付ける	140 ページの「void を使用しないで USB 大容量ストレージデバイスを追加する方法」
USB 大容量ストレージデバイスを取り外す	void を使用して USB 大容量ストレージデバイスを取り外す	141 ページの「void を使用して USB 大容量ストレージデバイスを取り外す方法」
	void を使用しないで USB 大容量ストレージデバイスを取り外す	141 ページの「void を使用しないで USB 大容量ストレージデバイスを取り外す方法」
USB カメラを追加する	USB カメラを追加して、デジタル画像にアクセスする	142 ページの「USB カメラを追加する方法」

デバイスのホットプラグとは、オペレーティングシステムをシャットダウンすることなくあるいはシステムの電源を切ることなく、デバイスを追加したり取り外したりすることを指します。USB デバイスはすべてホットプラグ可能です。

USB デバイスは、ホットプラグするとすぐにシステムのデバイス階層に表示されます (prtconf コマンドで確認可能)。また、デバイスが使用中でない限り、USB デバイスを取り外すとシステムのデバイス階層から消えます。

使用中の USB デバイスを取り外した場合、ホットプラグの動作は若干異なります。使用中の USB デバイスを取り外した場合、デバイスノードは残り、このデバイスを制御しているドライバはデバイス上のすべての動作を停止します。それ以降、このデバイスに発行される新しい入出力動作はエラーで戻されます。

このような場合、システムは元のデバイスを接続するようにユーザーにプロンプトを表示します。デバイスが使用できない場合は、アプリケーションを停止してください。数秒後に、ポートが再び使用できるようになります。

---

注 - 動作中の、つまり開いているデバイスを削除すると、データの整合性が損なわれる可能性があります。デバイスを取り外す前には、必ずデバイスを閉じるようにしてください。ただし、コンソールキーボードとマウスは例外で、動作中でも移動することができます。

---

## ▼ vold を使用して USB 大容量ストレージデバイスを追加する方法

次の手順は、vold を使用して USB デバイスを追加する方法を示します。

- 手順
1. **USB** 大容量ストレージデバイスを接続します。
  2. **vold** を使用して新しいデバイスがあるかどうか調べます。  

```
# touch /etc/vold.conf
```
  3. **vold** を再起動します。  

```
# pkill -HUP vold
```
  4. デバイスが取り付けられていることを確認します。  

```
$ ls device-alias
```

ボリューム管理デバイス名の詳細については、[第 1 章](#)を参照してください。

## ▼ vold を使用しないで USB 大容量ストレージデバイスを追加する方法

次の手順は、vold を使用しないで USB デバイスを追加する方法を示します。

- 手順
1. 必要に応じて、**vold** を無効にする方法について、[131 ページ](#)の「**vold** を使用しないで **USB** 大容量ストレージデバイスを使用できるように準備する方法」を参照してください。
  2. **USB** 大容量ストレージデバイスを接続します。
  3. **USB** デバイスが追加されたことを確認します。  
非 USB ストレージデバイスのデバイスリンクの間にあるかもしれない USB ディスクのデバイスリンクを、次のようにして確認します。  

```
$ cd /dev/rdisk  
$ ls -l c*0 | grep usb  
lrwxrwxrwx 1 root root 67 Apr 30 15:12 c1t0d0s0 ->
```

```
../../devices/pci@1f,0/pci@5/pci@0/usb@8,2/storage@1/disk@0,0:a,raw
```

## ▼ vold を使用して USB 大容量ストレージデバイスを取り外す方法

次の手順では、vold を使用して USB デバイスを取り外す例として、Zip ドライブを使用しています。

- 手順
1. そのデバイスを使用しているアプリケーションを実行中の場合は、そのアプリケーションを停止します。
  2. デバイスをマウント解除します。  
たとえば、次のようになります。  

```
$ volrmount -e zip0
```
  3. デバイスを取り出します。  
たとえば、次のようになります。  

```
$ eject zip0
```
  4. スーパーユーザーになり、vold を停止します。  

```
# /etc/init.d/volmgt stop
```
  5. USB 大容量ストレージデバイスを取り外します。
  6. vold を開始します。  

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

## ▼ vold を使用しないで USB 大容量ストレージデバイスを取り外す方法

次の手順は、vold を使用しないで USB デバイスを取り外す方法を示します。

- 手順
1. 必要に応じて、vold を無効にする方法について、131 ページの「vold を使用しないで USB 大容量ストレージデバイスを使用できるように準備する方法」を参照してください。
  2. スーパーユーザーになります。
  3. そのデバイスを使用しているアプリケーションを実行中の場合は、そのアプリケーションを停止します。
  4. USB デバイスを取り外します。

- a. デバイスをマウント解除します。  
たとえば、次のようになります。

```
# umount /mnt
```

- b. デバイスを取り外します。

## ▼ USB カメラを追加する方法

USB カメラを追加するには、次の手順を実行します。

手順 1. スーパーユーザーになります。

2. **USB** カメラを差し込み、電源をオンにします。

システムによって、カメラ用の論理デバイスが作成されます。カメラが差し込まれると、`/var/adm/messages` ファイルにメッセージが出力され、デバイスの接続が確認されます。カメラは、システムのストレージデバイスとみなされます。

3. `/var/adm/messages` ファイルの出力を確認します。

出力表示を確認すると、どの論理デバイスが作成されたかを確認でき、そのデバイスを使用してイメージにアクセスできます。出力表示は次のようになります。

```
# more /var/adm/messages
Jul 15 09:53:35 buffy usba: [ID 349649 kern.info]    OLYMPUS, C-3040ZOOM,
000153719068
Jul 15 09:53:35 buffy genunix: [ID 936769 kern.info] scsa2usb1 is
/pci@0,0/pci925,1234@7,2/storage@2
Jul 15 09:53:36 buffy scsi: [ID 193665 kern.info] sd3 at scsa2usb1:
target 0 lun 0
```

次のコマンドを実行して、デバイスをマウント可能な `/dev/dsk` リンクエントリに関連付けます。

```
# ls -l /dev/dsk/c*0 | grep /pci@0,0/pci925,1234@7,2/storage@2
lrwxrwxrwx  1 root    root          58 Jul 15  2002 c3t0d0p0 ->
../devices/pci@0,0/pci925,1234@7,2/storage@2/disk@0,0:a
```

4. **USB** カメラファイルシステムをマウントします。

ほとんどの場合、カメラのファイルシステムは PCFS ファイルシステムです。作成されたデバイス上にファイルシステムをマウントするために、ディスクを表すスライスを指定する必要があります。スライスは通常、SPARC システムでは `s0`、x86 システムでは `p0` です。

たとえば、x86 システムにファイルシステムをマウントするには、次のコマンドを実行します。

```
# mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

SPARC システムにファイルシステムをマウントするには、次のコマンドを実行します。

```
# mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
```

ファイルシステムのマウントについては、第 17 章を参照してください。

異なる PCFS ファイルシステムのマウントの詳細については、mount\_pcfs(1M) のマニュアルページを参照してください。

5. イメージファイルが使用可能であることを確認します。  
たとえば、次のようになります。

```
# ls /mnt/DCIM/100OLYMP/  
P7220001.JPG* P7220003.JPG* P7220005.JPG*  
P7220002.JPG* P7220004.JPG* P7220006.JPG*
```

6. USB カメラが作成したイメージファイルを表示します。

```
# /usr/dt/bin/sdtimage P7220001.JPG &
```

7. カメラを切り離す前に、ファイルシステムをマウント解除します。  
たとえば、次のようになります。

```
# umount /mnt
```

8. カメラの電源をオフにし、切り離します。

---

## USB オーディオデバイスの使用 (作業マップ)

作業	説明	参照先
USB オーディオデバイスを追加する	USB マイクおよびスピーカを追加する	145 ページの「USB オーディオデバイスを追加する方法」
システムの主オーディオデバイスを特定する	システムの主オーディオデバイスを特定する	146 ページの「システムの主オーディオデバイスを識別する方法」
主 USB オーディオデバイスを変更する	USB オーディオデバイスを取り外したり変更したりする場合、特定の 1 つのオーディオデバイスを主オーディオデバイスにすることがある	147 ページの「主 USB オーディオデバイスを変更する方法」

作業	説明	参照先
使用していない USB デバイスリンクを削除する	システムの電源がオフのときに USB オーディオデバイスを取り外すと、 <code>/dev/audio</code> デバイスが、存在しない <code>/dev/sound/*</code> デバイスを指したままになることがある	138 ページの「使用されていない USB デバイスのリンクを削除する方法」
USB オーディオに関する問題の障害追跡	USB スピーカからサウンドが出力されない場合は、この節を参照する	147 ページの「USB オーディオの問題を解決する」

## USB オーディオデバイスの使用

この Solaris リリースでは、2 つの連携するドライバ、`usb_ac` および `usb_as` の実装によって、USB オーディオサポートを提供しています。オーディオコントロールドライバである `usb_ac` は USB A (Solaris USB Architecture) 準拠のクライアントドライバで、ユーザーアプリケーションのインタフェースを制御します。オーディオストリーミングドライバである `usb_as` は、再生中および録音中にオーディオデータメッセージを処理します。また、サンプル周波数と精度を設定し、`usb_ac` ドライバからの要求を符号化します。どちらのドライバも、USB オーディオクラス 1.0 仕様に準拠しています。

一部のオーディオデバイスでは、ソフトウェアが制御している音量を設定できます。この機能を管理するために、STREAMS モジュールの `usb_ah` が HID ドライバの先頭に置かれます。

Solaris では、再生専用、録音専用、録音および再生用の USB オーディオデバイスをサポートします。USB オーディオデバイスのホットプラグがサポートされます。

- USB オーディオデバイスは、USB コネクタを備えた SPARC Ultra および x86 プラットフォームでサポートされます。
- Solaris 8 10/01、Solaris 8 2/02、Solaris 9 のいずれかのリリース上で再生または録音するには、USB オーディオデバイスが 44100 Hz または 48000 Hz の固定サンプリングレートをサポートしている必要があります。
- サポートされているオーディオデータ形式をすべて確認するには、`usb_ac(7D)` のマニュアルページを参照してください。

主オーディオデバイスは、`/dev/audio` です。次のコマンドを使用して、`/dev/audio` が USB オーディオを指しているかを確認できます。

```
% mixerctl
Device /dev/audioctl:
  Name      = USB Audio
  Version   = 1.0
  Config    = external

Audio mixer for /dev/audioctl is enabled
```



USB オーディオデバイスを接続した後、`audioplay` コマンドおよび `audiorecord` コマンドを使用し、`/dev/sound/N` デバイスリンクを介してデバイスにアクセスします。

`/dev/audio` および `/dev/sound/N` デバイスは、スピーカ、マイク、またはコンボデバイスを参照できます。不正なデバイスタイプを参照すると、そのコマンドは失敗します。たとえば、マイクに対して `audioplay` を使用しようとする、そのコマンドは失敗します。

ほとんどの Sun オーディオアプリケーションでは、特定のデフォルトオーディオデバイスを選択できます。たとえば、`audioplay` や `audiorecord` の場合には、`AUDIODEV` シェル変数を設定するか、`-d` オプションを指定します。ただし、`/dev/audio` をオーディオファイルとしてハードコードしているサードパーティ製のアプリケーションでは `AUDIODEV` は動作しません。

USB オーディオデバイスを差し込むと、`/dev/audio` が使用中でない限り、自動的にそれが主オーディオデバイス `/dev/audio` になります。オンボードのオーディオから USB オーディオへ、および USB オーディオからオンボードのオーディオへ `/dev/audio` を変更する方法については、[147 ページの「主 USB オーディオデバイスを変更する方法」](#) および `usb_ac(7D)` のマニュアルページを参照してください。

## 複数の USB オーディオデバイスのホットプラグ

USB オーディオデバイスがシステムに差し込まれると、それが主オーディオデバイス `/dev/audio` になります。そのデバイスは、システムをリブートした後でも、主オーディオデバイスのままです。USB オーディオデバイスが追加で差し込まれた場合、最後に差し込まれたデバイスが主オーディオデバイスになります。

USB オーディオデバイスに関する問題の障害追跡については、`usb_ac(7D)` のマニュアルページを参照してください。

## ▼ USB オーディオデバイスを追加する方法

USB オーディオデバイスを追加するには、次の手順を実行します。

手順 **1. USB スピーカを差し込みます。**

主オーディオデバイス `/dev/audio` は、USB スピーカを指します。

```
% ls -l /dev/audio
lrwxrwxrwx  1 root      root      10 Feb 13 08:46 /dev/audio -> usb/audio0
```

**2. (省略可能) スピーカを取り外します。その後、再度差し込みます。**

スピーカを取り外すと、`/dev/audio` デバイスがオンボードのオーディオに戻ります。

```
% ls -l /dev/audio
lrwxrwxrwx  1 root      root      7 Feb 13 08:47 /dev/audio -> sound/0
```

### 3. USB マイクを追加します。

```
% ls -l /dev/audio
lrwxrwxrwx  1 root      root      10 Feb 13 08:54 /dev/audio -> usb/audio1
```

## ▼ システムの主オーディオデバイスを識別する方法

この手順は、すでに USB オーディオデバイスが接続されていることを前提としています。

- 手順 ● システムの新しいオーディオリンクを調べます。

たとえば、次のようになります。

```
% ls -lt /dev/audio*
lrwxrwxrwx  1 root      root      7 Jul 23 15:46 /dev/audio -> usb/audio0
lrwxrwxrwx  1 root      root      10 Jul 23 15:46 /dev/audioc1 ->
usb/audioc10/
% ls -lt /dev/sound/*
lrwxrwxrwx  1 root      root      74 Jul 23 15:46 /dev/sound/1 ->
../../../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,a...
lrwxrwxrwx  1 root      root      77 Jul 23 15:46 /dev/sound/1ctl ->
../../../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,a...
lrwxrwxrwx  1 root      other    66 Jul 23 14:21 /dev/sound/0 ->
../../../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx  1 root      other    69 Jul 23 14:21 /dev/sound/0ctl ->
../../../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioc1
%
```

主オーディオデバイス `/dev/audio` が、新しく差し込まれた USB オーディオデバイスの `/dev/usb/audio0` を指していることがわかります。

`prtconf` コマンドを使用して USB デバイス情報を参照して、システム上の USB オーディオデバイスを調べることもできます。

```
% prtconf
.
.
.
usb, instance #0
  hub, instance #0
    mouse, instance #0
    keyboard, instance #1
    device, instance #0
      sound-control, instance #0
      sound, instance #0
      input, instance #0
.
.
.
```

## 主 USB オーディオデバイスを変更する方法

- オンボードのオーディオデバイスを主オーディオデバイスにするには、USB オーディオデバイスを取り外してください。すると、`/dev/audio` リンクは `/dev/sound/0` エントリを指すはずですが、`/dev/sound/0` エントリが主オーディオデバイスでない場合は、システムをシャットダウンして `boot -r` コマンドを実行するか、`devfsadm -i` コマンドをスーパーユーザーとして実行してください。
- USB オーディオデバイスを主オーディオデバイスにするには、USB オーディオデバイスを差し込んだあと、リンクを確認してください。

## USB オーディオデバイスに関する問題の障害追跡

この節では、USB オーディオデバイス問題の障害追跡の方法を説明します。

### USB オーディオの問題を解決する

ドライバを適用し、音量も上げているのに、USB スピーカから音が出ないことがあります。デバイスをホットプラグしてもこの動作が変化しないことがあります。

この問題を解決するには、USB スピーカの電源を再投入します。

### オーディオデバイスの所有権に関する注意事項

オーディオデバイスを操作するときは、オーディオデバイスの所有権に関する、次に挙げる点に注意してください。

- USB オーディオデバイスを差し込む時にコンソールからログインしていると、コンソールが `/dev/*` エントリの所有者になります。つまり、コンソールでログインしている限り、オーディオデバイスを使用できることになります。
- USB オーディオデバイスを差し込む時にコンソールにログインしていない場合、`root` がそのデバイスの所有者になります。ただし、その後にコンソールにログインして USB オーディオデバイスにアクセスしようとする、デバイスの所有権はコンソールに変更されず。詳細については、`logindevperm(4)` のマニュアルページを参照してください。
- リモートから `rlogin` コマンドでログインして USB オーディオデバイスにアクセスしようとした場合は、所有権は変更されません。たとえば、権限のないユーザーが、他の人の所有するマイクを通して行われる会話を聞くことはできません。

## cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)

作業	説明	参照先
USB バス情報を表示する	USB デバイスおよびバスについての情報を表示する	149 ページの「USB バス情報を表示する方法 (cfgadm)」
USB デバイスの構成を解除する	システムに物理的に接続されている USB デバイスを論理的に構成解除する	150 ページの「USB デバイスの構成を解除する方法」
USB デバイスを構成する	以前に構成を解除した USB デバイスを構成する	151 ページの「USB デバイスの構成方法」
論理的に USB デバイスの接続を解除する	物理的にシステムの近くにいないときには、USB デバイスを論理的に切り離すことができる	151 ページの「論理的に USB デバイスを接続解除する方法」
論理的に USB デバイスに接続する	以前に論理的に接続解除または構成解除した USB デバイスを論理的に接続する	151 ページの「論理的に USB デバイスを接続する方法」
USB デバイスのサブツリーの接続を解除する	ハブの下位階層 (または下位のツリー) である USB デバイスサブツリーの接続を解除する	152 ページの「論理的に USB デバイスのサブツリーを接続解除する方法」
USB デバイスをリセットする	USB デバイスをリセットして、デバイスを論理的に削除し、再作成する	152 ページの「USB デバイスのリセット方法」
複数の構成を持つ USB デバイスのデフォルト構成を変更する	複数の構成を持つ USB デバイスのデフォルト構成を変更する	153 ページの「複数の構成を持つ USB デバイスのデフォルト構成を変更する方法」

## cfgadm コマンドを使った USB デバイスのホットプラグ

cfgadm コマンドを使用せずに稼働中のシステムから USB デバイスを追加または削除することができます。ただし、USB デバイスは、デバイスを物理的に削除しなくても「論理的に」ホットプラグすることができます。この方法は、リモートで作業中に機能していない USB デバイスを無効にしたりリセットしたりする必要がある場合に便利です。cfgadm コマンドを使うと、メーカーや製品情報を含む USB デバイスツリーを表示することもできます。

cfgadm コマンドは接続点についての情報を表示します。接続点とは、動的再構成を行うことができるシステム内の特定の場所のことです。接続点は、次の要素から構成されています。

- 占有装置 (occupant)。USB デバイスなどの、システムに構成可能なハードウェアリソースのことです。
- 受容体 (receptacle)。USB ポートなどの、占有装置を受け入れる場所のことです。

接続点は、論理と物理の両方の接続点 ID (Ap\_Id) で表現されます。物理 Ap\_Id は接続点の物理的なパス名です。論理 Ap\_Id は物理 Ap\_Id に代わるユーザーに理解しやすい ID です。Ap\_Id の詳細については、cfgadm\_usb(1M) のマニュアルページを参照してください。

cfgadm コマンドを使用すると、USB デバイスステータス情報を取得できます。

受容体の状態	説明
empty/unconfigured	デバイスが物理的に接続されていない
disconnected/unconfigured	デバイスは物理的に接続されているかもしれないが、論理的に接続解除されており利用不可
connected/unconfigured	デバイスは論理的に接続されているが利用不可。このデバイスは、prtconf の出力に表示される
connected/configured	デバイスは接続されており利用可能

次の節では、ソフトウェアから cfgadm コマンドを使用して USB デバイスにホットプラグする方法について説明します。次のすべてのサンプル USB デバイス情報は、関連した情報に焦点を合わせるために一部省略されています。

## USB バス情報を表示する方法 (cfgadm)

cfgadm コマンドを使用して USB バス情報を表示します。たとえば、次のようになります。

```
% cfgadm
Ap_Id                Type                Receptacle  Occupant  Condition
usb0/4.5             usb-hub             connected   configured ok
usb0/4.5.1           usb-device          connected   configured ok
usb0/4.5.2           usb-printer         connected   configured ok
usb0/4.5.3           usb-mouse           connected   configured ok
usb0/4.5.4           usb-device          connected   configured ok
usb0/4.5.5           usb-storage         connected   configured ok
usb0/4.5.6           usb-communic        connected   configured ok
usb0/4.5.7           unknown            empty       unconfigured ok
usb0/4.6             usb-storage         connected   configured ok
```

```
usb0/4.7                usb-storage connected    configured    ok
```

前の例で usb0/4.5.1 は、第 2 レベルの外部ハブのポート 1 に接続されているデバイスを識別します。この第 2 レベルハブは第 1 レベルの外部ハブのポート 5 に接続されており、また第 1 レベルのハブは最初の USB コントローラのルートハブであるポート 4 に接続されています。

次の `cfgadm` コマンドを使用して、特定の USB デバイス情報を表示します。たとえば、次のようになります。

```
% cfgadm -l -s "cols=ap_id:info"
Ap_Id      Information
usb0/4.5.1  Mfg: Inside Out Networks Product: Edgeport/421 NConfigs: 1
Config: 0  : ...
usb0/4.5.2  Mfg: <undef> Product: <undef>   NConfigs: 1 Config: 0 ...
usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1 Config: 0 ...
usb0/4.5.4  Mfg: NMB Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0
usb0/4.5.5  Mfg: Hagiwara Sys-Com Product: SmartMedia R/W NConfigs: 1
Config: 0  : ...
usb0/4.5.6  Mfg: 3Com Inc. Product: U.S.Robotics 56000 Voice USB Modem
NConfigs: 2 ...
usb0/4.5.7
usb0/4.6    Mfg: Iomega Product: USB Zip 250 NConfigs: 1 Config: 0
: Default
usb0/4.7    Mfg: Iomega Product: USB Zip 100 NConfigs: 1 Config: 0
: Default
```

`prtconf` コマンドを使用して USB 構成情報を表示する例については、132 ページの「USB デバイス情報を表示する方法 (`prtconf`)」を参照してください。

## ▼ USB デバイスの構成を解除する方法

システムに物理的に接続されている USB デバイスの構成を解除することはできますが、ドライバを適用することはできません。USB デバイスの構成を解除しても、そのデバイスは `prtconf` 出力に表示されることに注意してください。

手順 1. スーパーユーザーになります。

2. **USB** デバイスの構成を解除します。

```
# cfgadm -c unconfigure usb0/4.7
Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7
This operation will suspend activity on the USB device
Continue (yes/no)? y
```

3. デバイスの構成が解除されていることを確認します。

```
# cfgadm
Ap_Id      Type      Receptacle  Occupant    Condition
usb0/4.5   usb-hub   connected   configured  ok
usb0/4.5.1 usb-device connected   configured  ok
```

```

usb0/4.5.2          usb-printer  connected  configured  ok
usb0/4.5.3          usb-mouse   connected  configured  ok
usb0/4.5.4          usb-device  connected  configured  ok
usb0/4.5.5          usb-storage connected  configured  ok
usb0/4.5.6          usb-communi connected  configured  ok
usb0/4.5.7          unknown     empty      unconfigured ok
usb0/4.6            usb-storage connected  configured  ok
usb0/4.7            usb-storage connected  unconfigured ok

```

## ▼ USB デバイスの構成方法

手順 1. スーパーユーザーになります。

2. **USB** デバイスを構成します。

```
# cfgadm -c configure usb0/4.7
```

3. **USB** デバイスが構成されていることを確認します。

```
# cfgadm usb0/4.7
Ap_Id              Type          Receptacle  Occupant    Condition
usb0/4.7           usb-storage   connected    configured  ok
```

## ▼ 論理的に USB デバイスを接続解除する方法

システムから USB デバイスを取り外し、prtconf 出力を削除したいが、物理的にシステムの近くにいない場合、USB デバイスの接続を論理的に解除できます。デバイスが物理的に接続されているが、論理的に接続解除されている場合には、そのデバイスは使用できず、システムにも表示されません。

手順 1. スーパーユーザーになります。

2. **USB** デバイスの接続を解除します。

```
# cfgadm -c disconnect -y usb0/4.7
```

3. デバイスが接続解除されていることを確認します。

```
# cfgadm usb0/4.7
Ap_Id              Type          Receptacle  Occupant    Condition
usb0/4.7           unknown       disconnected  unconfigured ok
```

## ▼ 論理的に USB デバイスを接続する方法

次の手順を使用して、以前に論理的に接続解除または構成解除された USB デバイスを論理的に接続します。

手順 1. スーパーユーザーになります。

2. **USB** デバイスを接続します。

```
# cfgadm -c configure usb0/4.7
```

3. デバイスが接続されていることを確認します。

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

デバイスを利用できるようになり、システムにも表示されるようになります。

## ▼ 論理的に USB デバイスのサブツリーを接続解除する方法

次の手順を使用して、USB デバイスのサブツリーを接続解除します。サブツリーは、ハブの下位デバイスの階層 (ツリー) です。

手順 1. スーパーユーザーになります。

2. **USB** デバイスサブツリーを削除します。

```
# cfgadm -c disconnect -y usb0/4
```

3. **USB** デバイスサブツリーの接続解除を確認します。

```
# cfgadm usb0/4
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4	unknown	disconnected	unconfigured	ok

## ▼ USB デバイスのリセット方法

USB デバイスでエラーが発生した場合は、`cfgadm` コマンドを使ってデバイスをリセットします。このコマンドを使うと、デバイスを論理的に削除し、再作成することができます。

手順 1. スーパーユーザーになります。

2. デバイスが使用中でないことを確認します。

3. デバイスをリセットします。

```
# cfgadm -x usb_reset -y usb0/4.7
```

4. デバイスが接続されていることを確認します。

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------



## ▼ 複数の構成を持つ USB デバイスのデフォルト構成を変更する方法

複数の構成を持つ USB デバイス进行操作するときには、次の点を考慮してください。

- USB デバイスの構成には、デバイス自体がどのようにオペレーティングシステムに表示されるかを定義します。この構成方法は、前の節で `cfgadm` について説明したシステムデバイスの構成方法とは異なります。
- 一部の USB デバイスでは、複数の構成がサポートされますが、一度に有効にできる構成は 1 つだけです。
- 複数の構成を持つデバイスを特定するには、`cfgadm -lv` の出力を確認します。`Nconfigs` は、1 より大きい値になります。
- デフォルトの USB 構成は、`configuration 1` です。現在の構成は、`cfgadm -lv` 出力の `Config` に反映されます。
- デフォルトの構成を変更しても、同じポートに再接続している間は、デバイスのリブート、ホットリムーブ、および再構成を行なっても、構成の変更は適用されません。

手順 1. デバイスが使用中でないことを確認します。

2. デフォルトの **USB** 構成を変更します。

たとえば、次のようになります。

```
# cfgadm -x usb_config -o config=2 usb0/4
Setting the device: /devices/pci@1f,0/usb@c,3:4
to USB configuration 2
This operation will suspend activity on the USB device
Continue (yes/no)? yes
```

3. デバイスの変更を確認します。

たとえば、次のようになります。

```
# cfgadm -lv usb0/4
Ap_Id  Receptacle  Occupant      Condition  Information  When  Type
      Busy      Phys_Id
usb0/4 connected  unconfigured ok           Mfg: Sun 2000
Product: USB-BOB0 aka Robotech
With 6 EPPS High Clk Mode  NConfigs: 7  Config: 2  : EVAL Board Setup
unavailable
usb-device  n           /devices/pci@1f,0/usb@c,3:4
Config が「2」になります。
```



## 第 9 章

---

# デバイスへのアクセス (概要)

---

ここでは、システム上のデバイスにアクセスする方法について説明します。

この章の内容は次のとおりです。

- 155 ページの「デバイスへのアクセス」
- 157 ページの「論理ディスクデバイス名」
- 160 ページの「論理テープデバイス名」
- 161 ページの「論理リムーバブルメディアデバイス名」

デバイスの構成についての概要は、第 5 章を参照してください。

---

## デバイスへのアクセス

コマンドを使用してディスク、ファイルシステムなどのデバイスを管理する場合、デバイス名を指定する方法を知っている必要があります。通常、論理デバイス名を使用して、システムに接続されたデバイスを表すことができます。論理デバイス名と物理デバイス名は、システム上でそれぞれ論理デバイスファイルと物理デバイスファイルによって表現されます。

## デバイス情報が作成される方法

システムがブートされると、デバイス階層が作成されて、システムに接続されたすべてのデバイスが表示されます。カーネルは、このデバイス階層情報を使用して、ドライバを該当するデバイスに対応づけて、特定の操作を実行するドライバへの一連のポインタを与えます。デバイス階層の詳細については、『*OpenBoot 3.x コマンド・リファレンスマニュアル*』を参照してください。

## デバイスの管理方法

devfsadm コマンドによって、/dev と /devices ディレクトリ内にある特殊デバイスファイル进行管理します。デフォルトでは、devfsadm コマンドはすべてのドライバをシステムに読み込み、可能な限りのデバイスに接続しようとします。次に、devfsadm コマンドはデバイスファイルを /devices ディレクトリに作成し、論理リンクを /dev ディレクトリに作成します。devfsadm コマンドは、/dev と /devices のディレクトリの管理に加えて、path\_to\_inst(4) インスタンスデータベースも保守します。

動的再構成イベントに回答する、再構成ブート処理とディレクトリ /dev および /devices の更新は、両方とも devfsadmd (devfsadm コマンドのデーモン版) によって処理されます。このデーモンは、システムブート時に /etc/rc\* スクリプトから起動します。

devfsadmd デーモンは再構成イベントによるデバイス構成の変化を自動的に検出するため、このコマンドを対話的に実行する必要はありません。

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

## デバイス名の命名規則

Solaris 環境では、デバイスは次の 3 つの方法で参照されます。

- 物理デバイス名 – デバイス情報階層内の完全デバイスパス名を表します。物理デバイス名は、次のコマンドを使用して表示できます。
  - dmesg
  - format
  - sysdef
  - prtconf物理デバイスファイルは、/devices ディレクトリにあります。
- インスタンス名 – システム上のデバイスすべてのカーネル短縮名を表します。たとえば、sd0 と sd1 は、2 つのディスクデバイスのインスタンス名を表します。インスタンス名は、/etc/path\_to\_inst ファイルにマップされており、次のコマンドによって表示できます。
  - dmesg
  - sysdef
  - prtconf
- 論理デバイス名 – デバイスを参照する際に、ほとんどのファイルシステムコマンドで使用されます。論理デバイス名を使用するファイルコマンドについては、[表 9-1](#) を参照してください。/dev ディレクトリの論理デバイスファイルは、/devices ディレクトリの物理デバイスファイルにシンボリックリンクされています。

## 論理ディスクデバイス名

論理デバイス名は、次の場合に、ディスクデバイスにアクセスするために使用されます。

- システムに新しいディスクを追加する
- システム間でディスクを移動する
- ローカルディスク上にあるファイルシステムにアクセスまたはそれをマウントする
- ローカルファイルシステムのバックアップをとる

管理コマンドの多くは、ディスクスライスまたはファイルシステムを参照する引数を使用します。

シンボリックリンクされるサブディレクトリ (`/dev/dsk` または `/dev/rdisk` のどちらか) に続けて、特定のコントローラ、ディスク、およびスライスを識別する文字列を指定することによって、ディスクデバイスを参照してください。

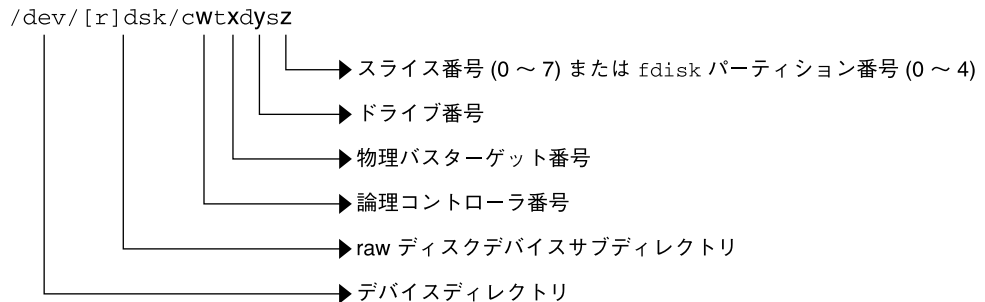


図 9-1 論理デバイス名

## ディスクサブディレクトリの指定

ディスクとファイルの管理コマンドには、`raw` (または「キャラクタ型」) デバイスインタフェースか、「ブロック」 デバイスインタフェースを使用する必要があります。この区別は、データがデバイスから読み取られる方法によって決まります。

`raw` デバイスインタフェースは、一度に少量のデータだけを転送します。ブロック デバイスインタフェースには、大量のデータブロックが一度に読み取られるバッファが含まれます。

コマンドによって、必要なインタフェースは異なります。

- コマンドが `raw` デバイスインタフェースを必要とする場合は、`/dev/rdsk` サブディレクトリを指定してください。( `rdsk` の `r` は、`raw` を表します。)

- コマンドがブロックデバイスインタフェースを必要とする場合は、`/dev/dsk` サブディレクトリを指定してください。
- コマンドが `/dev/dsk` または `/dev/rdisk` のどちらを必要とするかがわからない場合は、そのコマンドのマニュアルページの説明を参照してください。

次の表に、一般的に使用されるディスクコマンドとファイルシステムコマンドの一部に必要なインタフェースを示します。

表 9-1 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

特定ディスク上の特定スライスを識別するために使用する文字列は、コントローラのタイプが、直接またはバス指向のどちらであるかによって決まります。次の表に、さまざまなプラットフォームにおけるさまざまなタイプの直接コントローラまたはバス指向コントローラを示します。

表 9-2 コントローラのタイプ

直接コントローラ	バス指向コントローラ
IDE (x86)	SCSI (SPARC/x86)
	FCAL (SPARC)
	ATA (SPARC/x86)

両方のタイプのコントローラについて、次の節で説明します。

注 - コントローラ番号は、システム初期設定時に自動的に割り当てられます。この番号は、厳密に論理的なものであり、物理コントローラに直接対応するものではありません。

## x86: 直接コントローラでアクセスされるディスク

x86 システムにおいて IDE コントローラでアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

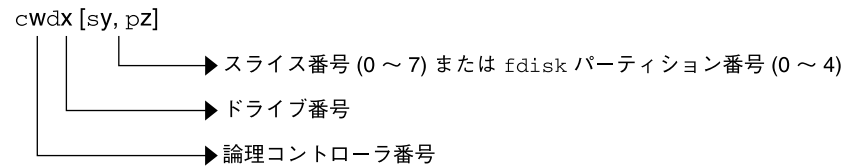


図 9-2 x86: 直接コントローラでアクセスされるディスク

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

## SPARC: バス指向コントローラでアクセスされるディスク

SPARC システムにおいてバス指向コントローラ (SCSI など) でアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

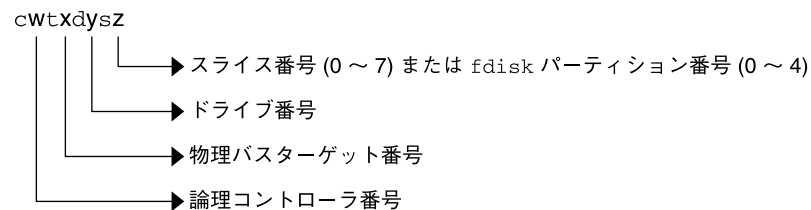


図 9-3 SPARC: バス指向コントローラでアクセスされるディスク

直接接続されるディスク (Ultra10 の IDE ディスクなど) を備えている SPARC システムでは、バス指向コントローラを備えているシステムと同じ命名規則になります。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

SCSI コントローラの場合、*x* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*y* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*y* は通常 0 になります。

ディスク全体を示すには、スライス 2 (s2) を指定してください。

## x86: SCSI コントローラでアクセスされるディスク

x86 システムにおいて SCSI コントローラでアクセスされる特定ディスク上の特定スライスを指定するには、下の図に示す命名規則に従ってください。

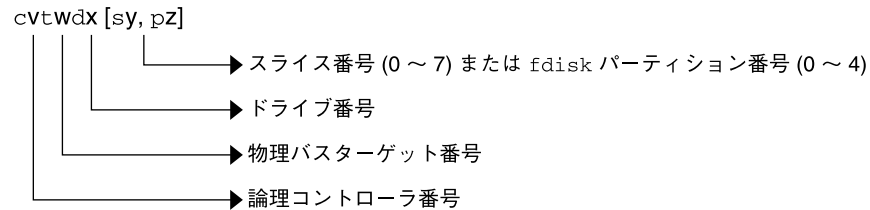


図 9-4 x86: SCSI コントローラでアクセスされるディスク

システムにコントローラが1つしかない場合、*v* は、通常、0 になります。

SCSI コントローラの場合、*w* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*x* はターゲットに接続されたドライブの論理ユニット番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*x* は通常 0 になります。

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

---

## 論理テープデバイス名

論理テープデバイスファイルは、/devices ディレクトリからのシンボリックリンクとして、/dev/rmt/\* ディレクトリにあります。



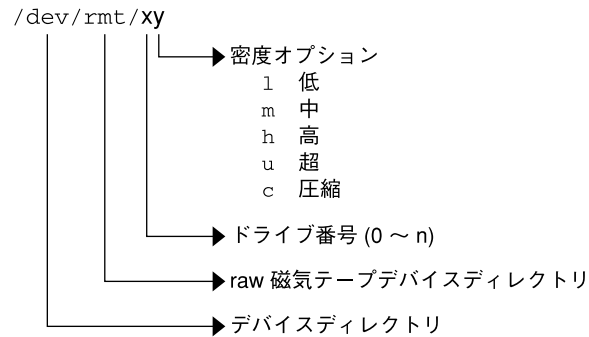


図 9-5 論理テープデバイス名

システムに接続された最初のテープデバイスは 0 (`/dev/rmt/0`) です。テープ密度の値 (`l`、`m`、`h`、`c`、および `u`) の詳細については、[第 28 章](#)を参照してください。

---

## 論理リムーバブルメディアデバイス名

リムーバブルメディアは、ボリューム管理 (`vol`) によって管理されるため、論理デバイス名は、手動でメディアをマウントしない限り、通常使用されません。

システムのリムーバブルメディアデバイスを表す論理デバイス名については、[第 2 章](#)を参照してください。



## 第 10 章

---

# ディスクの管理 (概要)

---

この章では、Solaris ディスクスライスの概要および format ユーティリティについて説明します。

この章の内容は次のとおりです。

- 163 ページの「Solaris 9 アップデートリリースの新しいディスク管理機能」
- 168 ページの「ディスク管理作業についての参照先」
- 168 ページの「ディスク管理の概要」
- 169 ページの「ディスク関連の用語」
- 169 ページの「ディスクスライスについて」
- 175 ページの「format ユーティリティ」
- 178 ページの「ディスクラベルについて」
- 181 ページの「ディスクをスライスに分割する」

システムにディスクを追加する方法については、第 12 章 または 第 13 章を参照してください。

---

## Solaris 9 アップデートリリースの新しい ディスク管理機能

この節では、この Solaris リリースの新しいディスク管理機能について説明します。

### SPARC: EFI ディスクラベルによるマルチテラバイト ディスクのサポート

**Solaris 9 4/03** – この Solaris リリースでは、64 ビット Solaris カーネルを実行しているシステムで 1T バイトを超えるディスクをサポートします。

EFI の仕様は、[http://www.intel.com/technology/efi/main\\_specification.htm](http://www.intel.com/technology/efi/main_specification.htm) からダウンロードできます。

EFI ラベルは、物理ディスクボリュームと仮想ディスクボリュームをサポートします。このリリースには、1T バイトを超えるディスクを管理するための更新版ディスクユーティリティが付属しています。UFS ファイルシステムには EFI ディスクラベルとの互換性があり、1T バイトを超えるサイズの UFS ファイルシステムを作成することができます。マルチテラバイトの UFS ファイルシステムの作成方法については、[249 ページの「SPARC: マルチテラバイトの UFS ファイルシステムのサポート」](#)を参照してください。

1T バイトを超えるファイルシステムを作成する必要がある場合は、別パッケージの Sun QFS ファイルシステムも使用できます。Sun QFS ファイルシステムについては、<http://docs.sun.com/db/doc/816-2542-10> を参照してください。

今回の Solaris リリースでは、Solaris ボリュームマネージャでも 1T バイトを超えるディスク管理が可能です。Solaris ボリュームマネージャの使用方法については、『Solaris ボリュームマネージャの管理』を参照してください。

サイズが 1T バイトに満たないディスクでは、引き続き VTOC ラベルを使用できます。現在のシステムでサイズが 1T バイトに満たないディスクしか使用しない場合は、以前の Solaris リリースと同じ方法でディスク管理を行います。サイズが 1T バイトに満たないディスクに EFI ラベルでラベル付けを行う場合は、`format-e` コマンドを使用します。詳細については、[例 11-6](#)を参照してください。

## EFI ラベルと VTOC ラベル

EFI ディスクラベルと VTOC ディスクラベルとの相違点は次のとおりです。

- サイズが 1T バイトを超えるディスクをサポートする
- スライス 0～6 を使用できる (スライス 2 はその他のスライス)
- 一次 (バックアップ) ラベルまたはその他のパーティションを使ってパーティションやスライスをオーバーラップすることはできない。EFI ラベルのサイズは通常 34 セクターであり、パーティションの開始セクターは 34 になる。つまり、開始セクターが 0 のパーティションは存在しない
- シリンダ、ヘッド、およびセクターの情報は、ラベルには格納されない。サイズはブロック単位で報告される
- 代替シリンダ領域 (ディスクの終わりから 2 つめまでのシリンダ) に格納されていた情報は、スライス 8 に格納される
- `format` ユーティリティを使ってパーティションサイズを変更する場合、サイズ 0 のパーティションには `unassigned` パーティションタグが割り当てられる。`format` ユーティリティでは、0 より大きいサイズのパーティションには、デフォルトにより `usr` パーティションタグが割り当てられる。パーティションを変更したあと新しいパーティションタグを割り当てたい場合は、パーティション変更メニューを使用する。ただし、サイズ 0 以外のパーティションに `unassigned` パーティションタグを割り当て直すことはできない

## EFI ディスクラベルの制限

サイズが 1T バイトを超えるディスクを使用することが現在の環境にとって適切かどうかを判断するときは、次の点を考慮してください。

- SCSI ドライバの `ssd` は、現時点で最大 2T バイトをサポートする。2T バイトを超えるディスク容量を必要とする場合は、Solaris ボリュームマネージャなどのボリューム管理製品を使用して大規模なデバイスを作成する
- EFI ラベル付きディスクを使用するシステム向けの階層化されたソフトウェア製品で、EFI ラベル付きディスクにアクセスできないことがある
- 以前の Solaris リリースを実行しているシステムは、EFI ラベル付きディスクを認識しない
- IDE ディスクでは EFI ディスクラベルがサポートされていない
- EFI ラベル付きディスクからブートを実行できない
- Solaris 管理コンソールの「ディスク」ツールでは、EFI ラベル付きディスクを管理できない。`format` ユーティリティを使って EFI ラベル付きディスクをパーティションに分割する。すると、Solaris 管理コンソールの拡張ストレージツールを使って、EFI ラベル付きディスクを含むボリュームやディスクセットを管理できるようになる
- EFI 仕様では、スライスのオーバーラップは禁止されている。`cxydz` でディスク全体を表現する
- ディスクやパーティションのサイズ情報を提供する。使用可能な単位はセクターまたはブロック。シリンダおよびヘッドは使用できない
- 次の `format` オプションは、EFI ラベル付きディスクではサポートされていないか、不適切である
  - `save` オプション。EFI ラベル付きディスクは `format.dat` ファイルにエントリを必要としない。したがって、このオプションはサポートされない
  - `backup` オプション。ディスクドライバは、一次ラベルを検出してディスクに書き込む。したがって、このオプションは不適切である

## EFI ラベル付きディスクを使用したシステムへのインストール

Solaris インストールユーティリティは、EFI ラベル付きディスクを自動的に認識します。しかし、これらのディスクのパーティション分割をやり直すことはできません。インストール前またはインストール後にこれらのディスクのパーティション分割をやり直したい場合は、`format` ユーティリティを使用してください。Solaris Upgrade ユーティリティおよび Live Upgrade ユーティリティも EFI ラベル付きディスクを認識します。しかし、EFI ラベル付きディスクからシステムのブートを実行することはできません。

EFI ラベル付きディスクを使用するシステム上に Solaris をインストールした場合、次のようなパーティションテーブル情報が得られます。

Current partition table (original):  
Total disk sectors available: 2576924638 + 16384 (reserved sectors)

Part	Tag	Flag	First Sector	Size	Last Sector
0	root	wm	34	1.20TB	2576924636
1	unassigned	wm	0	0	0
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	unassigned	wm	0	0	0
8	reserved	wm	2576924638	8.00MB	2576941021

## EFI ラベル付きディスクの管理

EFI ラベル付きディスクの管理方法は、次の表で確認できます。

作業	参照先
インストール済みのシステムにディスクを接続し、再構成(ブート)を実行する	207 ページの「SPARC: システムディスクまたは二次ディスクの追加(作業マップ)」
format ユーティリティを使ってディスクのパーティション分割を行う(必要に応じて)	211 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」
ディスクボリュームを作成する。その後、Solaris ボリュームマネージャでソフトパーティションを作成する(必要に応じて)	『Solaris ボリュームマネージャの管理』の第2章「記憶装置管理の概念」
newfs コマンドを使用して新しいディスク用の UFS ファイルシステムを作成する	215 ページの「SPARC: UFS ファイルシステムを作成する方法」
または、QFS ファイルシステムを作成する	<a href="http://docs.sun.com/db/coll/20445.2">http://docs.sun.com/db/coll/20445.2</a>

## EFI ラベル付きディスクの複製

以前の Solaris リリースでは、ディスク全体がスライス 2 (s2) で表されていました。dd コマンドを使って、ディスクを複製またはコピーできます。このコマンドは、次のように入力します。

```
dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2 bs=128k
```

ここでは、複製後のディスクに一意的 UUID を割り当てる必要があります。したがって、1T バイトを超えるサイズのディスクの複製(コピー)方法が若干異なります。たとえば、次のようになります。

1. dd コマンドを使って EFI ラベル付きディスクを複製します。

```
# dd if=/dev/rdisk/c0t0d0 of=/dev/rdisk/c0t2d0 bs=128k
```

2. 複製対象のディスクの prtvtoc 出力を fmthard コマンドにパイプして、複製済みディスク用の新しいラベルを作成します。

```
# prtvtoc /dev/rdisk/c0t0d0 | fmthard -s - /dev/rdisk/c0t2d0
```



---

注意 - 複製済みディスク用の新しいラベルは、必ず作成してください。作成しないと、その他のソフトウェア製品によって UUID の重複が検出された時点で、EFI ラベル付きディスクのデータが破壊されることがあります。

---

## EFI ディスクラベルに関する問題の障害追跡

EFI ラベル付きディスクに関する問題の障害追跡には、次のエラーメッセージと解決法を参考にしてください。

### エラーメッセージ

```
The capacity of this LUN is too large.  
Reconfigure this LUN so that it is < 2TB.
```

### 原因

SCSI デバイス上に 2T バイトを超えるパーティションを作成しようとしてしました。

### 解決法

SCSI デバイス上には、2T バイト未満のパーティションを作成してください。

### エラーメッセージ

```
Dec 3 09:26:48 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/  
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):  
Dec 3 09:26:48 holoship disk has 2576941056 blocks, which is too large  
for a 32-bit kernel
```

### 原因

32 ビット SPARC カーネルを実行しているシステムを、サイズが 1T バイトを超えるディスクでブートしようとしてしました。

### 解決法

サイズが 1T バイト以上のディスクでは、64 ビット SPARC カーネルを実行しているシステムをブートしてください。

### エラーメッセージ

```
Dec 3 09:12:17 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/  
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):  
Dec 3 09:12:17 holoship corrupt label - wrong magic number
```

### 原因

Solaris 9 8/03 より古い Solaris リリースで動作するシステムにこのディスクを追加しようとしてしました。

### 解決法

このディスクは、EFI ディスクラベルをサポートしている Solaris リリースで動作するシステムに追加してください。

---

## ディスク管理作業についての参照先

ディスク管理の手順については、次を参照してください。

ディスク管理作業	参照先
ディスクをフォーマットしてディスクラベルを確認する。	第 11 章
SPARC システムに新しいディスクを追加する。	第 12 章
x86 システムに新しいディスクを追加する。	第 13 章
SCSI または PCI ディスクのホットプラグを実行する。	第 6 章

---

## ディスク管理の概要

一般に、Solaris 環境におけるディスクの管理とは、システムを設定し、Solaris インストールプログラムを実行し、適切なディスクスライスおよびファイルシステムを作成してオペレーティングシステムをインストールすることを意味します。また、`format` ユーティリティを使用して、新しいディスクドライブを追加したり、欠陥ディスクドライブを交換したりしなければならない場合もあります。

---

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

---



---

## ディスク関連の用語

この節の説明を有効に利用するには、基本的なディスクアーキテクチャを理解しておく必要があります。特に、次の用語を理解しておいてください。

用語	説明
トラック	ディスクが回転するときに1つの静止したディスクヘッドの下を通過する同心リング
シリンダ	ディスクが回転する軸から同じ距離にあるトラックの集まり
セクター	各ディスクプラッタのセクション。1セクターは512バイト
ブロック	ディスク上のデータ記憶領域。1ディスクブロックは512バイト
ディスクコントローラ	ディスクドライブを制御するチップおよび関連する回路
ディスクラベル	ディスクのジオメトリおよびパーティション情報を含む、ディスクの第1セクター
デバイスドライバ	ハードウェアまたは仮想デバイスを制御するカーネルモジュール

詳細については、ディスクメーカーの製品情報を参照してください。

---

## ディスクスライスについて

ディスク上に格納されたファイルは、ファイルシステム中で管理されます。ディスク上の各ファイルシステムは「スライス」、つまりファイルシステム用に確保されたセクターセットのグループに割り当てられます。オペレーティングシステム (および、システム管理者) からは、各ディスクスライスは別個のディスクドライブであるかのように見えます。

ファイルシステムの詳細については、[第15章](#)を参照してください。

---

注 - スライスをパーティションと呼ぶこともあります。このマニュアルでは「スライス」と呼びますが、`format` ユーティリティなど、特定のインタフェースではスライスを「パーティション」と呼びます。

---

スライスを設定するときには、次の規則に注意してください。

- 各ディスクスライスは、ファイルシステムを1つしか持てない。
- ファイルシステムを複数のスライスにまたがって割り当てることはできない。

SPARC プラットフォームと x86 プラットフォームでは、スライスの設定が少し異なります。次の表に、主な相違点を示します。

表 10-1 プラットフォームによるスライスの違い

SPARC プラットフォーム	x86 プラットフォーム
ディスク全体が Solaris オペレーティング環境用になる。	ディスクはオペレーティングシステムごとに1つの <code>fdisk</code> パーティションに分割される。
<b>VTOC</b> - ディスクは8スライス (スライス0 ~ 7) に分割される。	<b>VTOC</b> - Solaris の <code>fdisk</code> パーティションは10スライス (スライス0 ~ 9) に分割される。
<b>EFI</b> - ディスクは7スライス (スライス0 ~ 6) に分割される。	

Solaris ボリュームマネージャ (以前の Solstice DiskSuite™) には「ソフトパーティション」と呼ばれるパーティション分割機能が備わっており、これを使えば、1つのディスクに9個以上のパーティションを作成できます。

Solaris ボリュームマネージャの一般的な情報については、『Solaris ボリュームマネージャの管理』の第2章「記憶装置管理の概念」を参照してください。ソフトパーティションについては、『Solaris ボリュームマネージャの管理』の第12章「ソフトパーティション (概要)」を参照してください。

## SPARC: ディスクスライス

次の表に、SPARC システムのスライスの説明を示します。

表 10-2 SPARC: ディスクスライス

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	説明
0	ルート (/)	両方	オペレーティングシステムを構成するファイルとディレクトリを含む。 <b>EFI</b> - <b>EFI</b> ラベル付きディスクからはブートを実行できない。
1	スワップ	両方	仮想メモリー、つまり「スワップ空間」を提供する。

表 10-2 SPARC: ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	説明
2	—	両方	<p><b>VTOC</b> – 慣例的にディスク全体を表す。このスライスのサイズは変更しないこと。</p> <p><b>EFI</b> – サイトの必要に応じてオプションスライスを定義する。</p>
3	/export	両方	<p>サイトの必要に応じて定義可能なスライス (任意)。</p> <p>クライアントシステムが必要とする代替オペレーティングシステムを格納するため、サーバー上で使用できる。</p>
4		両方	<p>サイトの必要に応じて定義可能なスライス (任意)。</p>
5		両方	<p>サイトの必要に応じて定義可能なスライス (任意)。</p> <p>システムに追加されるアプリケーションソフトウェアを格納する。インストール時に、/opt ファイルシステムにスライスが割り当てられていなければ、スライス 0 に /opt ディレクトリが作成される。</p>
6	/usr	両方	<p>オペレーティングシステムのコマンド (「実行可能」コマンドとも呼ぶ) を含む。このスライスには、マニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。</p>
7	/home または /export/home	両方	<p><b>VTOC</b> – ユーザーによって作成されたファイルを格納する。</p> <p><b>EFI</b> – 適用できない。</p>
8	なし	なし	<p><b>VTOC</b> – 適用できない。</p> <p><b>EFI</b> – デフォルトで作成された予約済みスライス。VTOC の代替シリンダによく似た領域。このスライスは変更または削除しないこと。</p>

## x86: ディスクスライス

x86 システム上では、ディスクは fdisk パーティションに分割されます。fdisk パーティションは、Solaris など、特定のオペレーティングシステムで使用するよう確保されたディスクの一部です。

次の表に示すように、Solaris は Solaris fdisk パーティション上に、0 から 9 までの番号が付いた 10 個のスライスを配置します。

表 10-3 x86: ディスクスライス

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	目的
0	ルート (/)	両方	オペレーティングシステムを構成するファイルとディレクトリを含む。
1	スワップ	両方	仮想メモリー、つまり「スワップ空間」を提供する。
2	—	両方	慣例的に、このスライスは Solaris fdisk パーティション全体を表す。このスライスのサイズは変更しないこと。
3	/export	両方	サイトの必要に応じて定義可能なスライス (任意)。  クライアントシステムが必要とする代替オペレーティングシステムを格納するため、サーバー上で使用できる。
4			サイトの必要に応じて定義可能なスライス (任意)。
5		両方	サイトの必要に応じて定義可能なスライス (任意)。  システムに追加されるアプリケーションソフトウェアを格納する。インストール時に、/opt ファイルシステムにスライスが割り当てられていなければ、スライス 0 に /opt ディレクトリが作成される。
6	/usr	両方	オペレーティングシステムのコマンド (「実行可能」コマンドとも呼ぶ) を含む。このスライスには、マニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。

表 10-3 x86: ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	目的
7	/home または /export/home	両方	ユーザーによって作成されるファイルを含む。
8	—	両方	Solaris がハードディスクからブートするために必要な情報を含む。このスライスのスライス番号は 8 であるが、Solaris fdisk パーティションの先頭に存在するため、ブートスライスと呼ばれる。
9	—	両方	代替ディスクブロック用に予約された領域。スライス 9 は代替セクタスライスと呼ばれる。

## raw データスライスの使用

SunOS オペレーティングシステムは、各ディスクのブロック 0 にディスクラベルを格納します。これは、raw データスライスを作成する Sun 以外の製品のデータベースアプリケーションを使用するときは、ブロック 0 から開始してはならないことを意味します。この領域に raw データスライスを作成すると、ディスクラベルが上書きされて、ディスク上のデータにアクセスできなくなります。

ディスク上の次の領域は、raw データスライス用に使用しないでください。raw データスライスは Sun 以外のデータベースアプリケーションによって作成されることがあります。

- ブロック 0 (ディスクラベルが格納される領域)
- スライス 2 (VTOC ラベル付きディスク全体を表す)

## 複数のディスク上のスライス配置

十分な大きさのディスクであれば、1 台ですべてのスライスとそれに対応するファイルシステムを確保できますが、通常はシステムのスライスとファイルシステムを確保するために複数のディスクが使用されます。

---

注-1 つのスライスを複数のディスクに分割することはできません。ただし、複数のスワップスライスを別々のディスクに配置することはできます。

---

たとえば、1 台のディスクにルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムを入れ、別のディスクにユーザーデータを含む /export/home ファイルシステムやその他のファイルシステムを入れます。

複数のディスクを使用する場合、オペレーティングシステムソフトウェアとスワップ領域が入っているディスク (つまり、ルート (/)、/usr ファイルシステム、およびスワップ領域用のスライスが入っているディスク) を、「システムディスク」と呼びます。システムディスク以外のディスクを、「二次ディスク」または「非システムディスク」と呼びます。

システムのファイルシステムを複数のディスクに入れると、システムをシャットダウンしたりオペレーティングシステムソフトウェアをロードし直したりしなくても、二次ディスクのファイルシステムとスライスを変更できます。

また、複数のディスクを使用すると、入出力 (I/O) のパフォーマンスが改善されます。ディスク負荷を複数のディスクに分散すると、I/O のボトルネックを回避できます。

## 使用するスライスの決定

ディスクのファイルシステムを設定するときには、各スライスのサイズだけでなく、どのスライスを使用するかも決定します。どのように決定するかは、ディスクを接続するシステムの構成と、ディスクにインストールするソフトウェアによって異なります。

次のシステム構成があります。

- サーバー
- スタンドアロンシステム

システムの構成が異なれば、スライスの使用方法も異なります。次の表に例を挙げます。

表 10-4 システム構成とスライス

スライス	サーバー	スタンドアロンシステム
0	ルート	ルート
1	スワップ	スワップ
2	—	—
3	/export	—
6	/usr	/usr
7	/export/home	/home

システム構成の詳細については、『Solaris のシステム管理 (基本編)』の「システムタイプの概要」を参照してください。

---

注 – Solaris インストールプログラムは、インストール用に選択したソフトウェアに基づいてデフォルトのスライスサイズを表示します。

---

## format ユーティリティ

手順や参照情報のセクションに進む前に、以下の情報に目を通して format ユーティリティの概要とその使用方法を確認してください。

format ユーティリティは、Solaris システム用にハードディスクドライブを用意するためのシステム管理ツールです。

次の表に、format ユーティリティの機能とその利点を示します。

表 10-5 format ユーティリティの機能と利点

機能	利点
システム内で接続されている全ディスクドライブを検索する	次の状態を報告する <ul style="list-style-type: none"><li>■ ターゲットの位置</li><li>■ ディスクのジオメトリ</li><li>■ ディスクがフォーマット済みかどうか</li><li>■ 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</li></ul>
ディスクラベルを検索する	修復処理に使用する
欠陥セクターを修復する	回復可能なエラーが発生したディスクドライブをメーカーに返送しなくても、熟練した管理者なら修復できる
ディスクをフォーマットして、分析する	ディスク上でセクターを作成し、検査する
ディスクをパーティションに分割する	個々のファイルシステムを別々のスライス上で作成できるようにディスクを分割する
ディスクにラベルを付ける	後から検索できるように (通常は修復用)、ディスクにディスク名と構成情報を書き込む

format ユーティリティのオプションの詳細については、[第 14 章](#)を参照してください。

## format ユーティリティを使用する場合

Solaris のインストール時に、Solaris インストールプログラムによってディスクがパーティションに分割され、ラベルが付けられます。次のような場合に、format ユーティリティを使用できます。

- スライス情報を表示する
- ディスクをスライスに分割する
- 既存のシステムにディスクドライブを追加する
- ディスクドライブをフォーマットする
- ディスクにラベルを付ける
- ディスクドライブを修復する
- ディスクのエラーを分析する

システム管理者が format ユーティリティを使用するのは、主にディスクをディスクスライスに分割するためです。これらの手順については、第 12 章と第 13 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、次の節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 10-6 format ユーティリティのガイドライン

作業	注意事項	参照先
ディスクをフォーマットする	<ul style="list-style-type: none"><li>■ 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</li><li>■ ディスクドライブをフォーマットしてパーティションに分割した状態で出荷するメーカーが増えているので、ディスクドライブをフォーマットする必要性は減少している。既存のシステムにディスクドライブを追加する場合は、format ユーティリティを使用しなくてもすむことがある</li><li>■ 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</li></ul>	187 ページの「ディスクをフォーマットする方法」



表 10-6 format ユーティリティのガイドライン (続き)

作業	注意事項	参照先
システムディスクを交換する	<ul style="list-style-type: none"> <li>■ 損傷したシステムディスクのデータは、バックアップメディアから復元しなければならない。復元しなければ、インストールプログラムを使用してシステムをもう一度インストールしなければならないことになる</li> </ul>	<p>209 ページの「SPARC: システムディスクを接続してブートする方法」または 219 ページの「x86: システムディスクを接続してブートする方法」、システムをインストールし直さなければならない場合は『Solaris 9 9/04 インストールガイド』を参照</p>
ディスクをスライスに分割する	<ul style="list-style-type: none"> <li>■ すでにスライスに分割されているディスクで、パーティションを再分割してラベルを付け直すと、既存のデータが失われる</li> <li>■ ディスクのパーティションを分割し直して復元する前に、既存のデータをバックアップメディアにコピーする必要がある</li> </ul>	<p>211 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 227 ページの「x86: ディスクスライスとディスクラベルを作成する方法」</p>
既存のシステムにディスクを追加する	<ul style="list-style-type: none"> <li>■ 二次ディスクをフォーマットし直すか、パーティションに分割し直す場合は、既存のデータをバックアップメディアから復元しなければならない</li> </ul>	<p>209 ページの「SPARC: 二次ディスクを接続してブートする方法」または 219 ページの「x86: 二次ディスクを接続してブートする方法」</p>
ディスクドライブを修復する	<ul style="list-style-type: none"> <li>■ 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</li> <li>■ 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</li> <li>■ 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</li> </ul>	<p>201 ページの「欠陥セクターの修復」</p>

## ディスクのフォーマット

ほとんどの場合、ディスクはメーカーまたは再販業者によってフォーマットされています。このため、ドライブをインストールするときにフォーマットし直す必要はありません。ディスクがフォーマットされているかどうかを判別するには、`format` ユーティリティを使用します。詳細については、187 ページの「ディスクがフォーマット済みかを調べる方法」を参照してください。

ディスクがフォーマットされていない場合、`format` ユーティリティを使用してフォーマットしてください。

ディスクのフォーマットでは、次の 2 つのステップが行われます。

- ディスクメディアを使用できるようにする
- 表面解析に基づいてディスクの欠陥リストを作成する



---

注意 - フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常はメーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、`format` ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。詳細については、187 ページの「ディスクをフォーマットする方法」を参照してください。

---

データに利用できる合計ディスク容量のうち、ごくわずかな容量が欠陥情報とフォーマット情報の格納に使用されます。この容量はディスクのジオメトリによって異なり、使用年数がたち欠陥箇所が多くなるにつれて、少なくなります。

ディスクの種類とサイズに応じて、フォーマットは数分から数時間かかります。

---

## ディスクラベルについて

どのディスクにも、そのディスクのコントローラ、ジオメトリ、スライスに関する情報を格納する特殊な領域が確保されています。そのような情報をディスクの「ラベル」と呼びます。VTOC ラベル付きのディスク上のディスクラベルを「VTOC (Volume Table of Contents)」と呼びます。「ディスクにラベルを付ける」とは、ディスクにスライス情報を書き込むことを意味します。通常は、ディスクのスライスを変更した後にラベルを付けます。

スライスを作成した後でディスクにラベルを付けないと、オペレーティングシステムはスライスを「認識」する方法がないので、そのスライスを利用できなくなります。

## パーティションテーブル

ディスクラベルのうち重要な部分は「パーティションテーブル」です。この部分は、ディスクのスライス、スライスの境界(シリンダ単位)、スライスの合計サイズを表します。ディスクのパーティションテーブルは、format ユーティリティを使用して表示できます。次の表に、パーティションテーブル関連の用語を示します。

表 10-7 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	<b>VTOC</b> -0~7の番号が付いたパーティションまたはスライス  <b>EFI</b> -0~6の番号が付いたパーティションまたはスライス
タグ	0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME 11=RESERVED	一般にこのパーティションにマウントされたファイルシステムを記述する数値
フラグ	wm  wu rm  rm	パーティションは書き込み可能でマウント可能  パーティションは書き込み可能でマウント不可。これは、スワップ領域専用のパーティションのデフォルト状態である。ただし、mount コマンドでは「マウント不可」のフラグはチェックされない  パーティションは読み取り専用でマウント可能

パーティションのフラグとタグは必ず割り当てられるので、管理する必要はありません。

パーティションテーブルを表示する手順については、189 ページの「ディスクスライス情報を表示する方法」または 193 ページの「ディスクラベルを検査する方法」を参照してください。

## パーティションテーブル情報の表示

次の例は、format ユーティリティを使って、4.0G バイトの VTOC ラベル付きディスクのパーティションテーブルを表示したものです。

```
Total disk cylinders available: 8892 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	1110 - 4687	1.61GB	(0/3578/0) 3381210
1	swap	wu	0 - 1109	512.00MB	(0/1110/0) 1048950

```

2      backup      wm      0 - 8891      4.01GB      (0/8892/0) 8402940
3 unassigned      wm      0              0              (0/0/0)      0
4 unassigned      wm      0              0              (0/0/0)      0
5 unassigned      wm      0              0              (0/0/0)      0
6 unassigned      wm      0              0              (0/0/0)      0
7      home        wm      4688 - 8891    1.89GB      (0/4204/0) 3972780

```

format コーティリテイを使用して表示されるパーティションテーブルには、次の情報が含まれます。

列名	説明
Part	パーティション (またはスライス番号)。この列についての説明は、表 10-7 を参照。
Tag	パーティションのタグ。この列についての説明は、表 10-7 を参照。
Flags	パーティションのフラグ。この列についての説明は、表 10-7 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。
First Sector	<b>EFI</b> - 開始ブロック番号。
Last Sector	<b>EFI</b> - 終了ブロック番号。

次の例は、prtvtoc コマンドを使用して EFI ディスクラベルを表示した結果です。

```

# prtvtoc /dev/rdsk/c4t1d0s0
* /dev/rdsk/c4t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
* 2576941056 sectors
* 2576940989 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
* Partition  Tag  Flags      First      Sector      Last
*           Count  Sector
* 0         2    00          34    629145600  629145633
* 1         4    00    629145634  629145600 1258291233
* 6         4    00 1258291234 1318633404 2576924637
* 8        11    00 2576924638      16384 2576941021
*
* Flags:
*   1: unmountable
*  10: read-only

```

\*

prtvtoc コマンドで、次の情報が取得できます。

列名	説明
Dimensions	このセクションでは、ディスクドライブの物理的な構成を示す。
Flags	このセクションでは、パーティションテーブルのセクションに記載されたフラグを示す。パーティションフラグについての説明は、表 10-7 を参照。
Partition (または Slice) テーブル	このセクションには次の情報が含まれます。
Partition	パーティション (またはスライス番号)。この列についての説明は、表 10-7 を参照。
Tag	パーティションのタグ。この列についての説明は、表 10-7 を参照。
Flags	パーティションのフラグ。この列についての説明は、表 10-7 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	スライスの最後のセクターを示す。
Mount Directory	ファイルシステムの最後のマウントポイントのディレクトリを示す。

## ディスクをスライスに分割する

format ユーティリティは、主にシステム管理者がディスクをスライスに分割する場合に使われます。手順を次に示します。

- どのスライスが必要かを決定する
- 各スライスのサイズを決定する
- format ユーティリティを使用してディスクをスライスに分割する
- 新しいスライス情報を使用してディスクにラベルを付ける
- スライスごとにファイルシステムを作成する

ディスクをスライスに分割するには、format ユーティリティの partition メニューで modify コマンドを実行するのがもっとも簡単です。modify コマンドを使用すると、開始シリンダ境界を追跡しなくても、各スライスのサイズを指定してスライスを作成できます。modify コマンドを使用すると、「free hog」スライス内の残りのディスク領域も追跡できます。

## free hog スライスの使用方法

`format` ユーティリティを使用してディスクスライスのサイズを変更するときには、サイズ変更操作に対応して拡大縮小する一時スライスを指定します。

このスライスは、スライスを拡大すると領域を「解放 (free)」し、スライスを圧縮すると放棄された領域を「回収 (hog)」します。このため、提供側のスライスを「free hog」と呼びます。

free hog スライスは、インストール時または `format` ユーティリティの実行時にのみ存在します。日常の操作中に free hog スライスが継続して存在することはありません。

free hog スライスの使用方法については、211 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 227 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

## 第 11 章

# ディスクの管理 (手順)

この章では、ディスク管理の手順について説明します。Solaris システム上でディスクを管理する方法に精通している場合は、この章で説明する多くの内容を読み飛ばすことができます。

ディスク管理に関連した手順の詳細については、183 ページの「ディスクの管理 (作業マップ)」を参照してください。

ディスク管理の概要については、第 10 章を参照してください。

# ディスクの管理 (作業マップ)

作業	説明	参照先
システム上のディスクの確認	システム上のディスクの種類が分からない場合は、 <code>format</code> ユーティリティを使用して確認する。	185 ページの「システム上のディスクを確認する方法」
ディスクのフォーマット	<code>format</code> ユーティリティを使用して、ディスクがフォーマット済みかどうかを判断する。	187 ページの「ディスクがフォーマット済みかを調べる方法」
	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。	187 ページの「ディスクをフォーマットする方法」

作業	説明	参照先
スライス情報の表示	format ユーティリティを使用してスライス情報を表示する。	189 ページの「ディスクスライス情報を表示する方法」
ディスクラベルの作成	format ユーティリティを使用してディスクラベルを作成する。	191 ページの「ディスクラベルを作成する方法」
ディスクラベルの検査	prtvtoc コマンドを使用してディスクラベルを検査する。	193 ページの「ディスクラベルを検査する方法」
破損したディスクラベルの復元	システム障害または電源障害のために破損したディスクラベルの復元を試みる。	195 ページの「破損したディスクラベルを復元する方法」
format.dat のエントリの作成	サードパーティのディスクをサポートするために format.dat のエントリを作成する。	198 ページの「format.dat のエントリを作成する方法」
SCSI ディスクの自動構成	特定のドライブタイプが /etc/format.dat ファイルに記載されていない場合でも、ディスクデバイスモードセクションの SCSI-2 仕様を利用して、SCSI ディスクを自動構成できる。	200 ページの「SCSI ドライブを自動構成する方法」
欠陥ディスクセクターの検出	format ユーティリティを使用して欠陥ディスクセクターを調べる。	202 ページの「表面解析を使用して欠陥セクターを調べる方法」
欠陥ディスクセクターの修復 (必要な場合)	format ユーティリティを使用して欠陥ディスクセクターを修復する。	203 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 14 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

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを使用して、システム上で認識されるディスクを確認します。

```
# format
format ユーティリティは、AVAILABLE DISK SELECTIONS という見出しの下に、認識されるディスクのリストを表示します。
```

### 例 11-1 システム上のディスクを確認する

次の **format** 出力は、単一のディスクを持つシステムのものです。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t0d0 <ST34321A cyl 8892 alt 2 hd 15 sec 63>
     /pci@1f,0/pci@1,1/ide@3/dad@0,0
Specify disk (enter its number):
```

**format** の出力は、ディスクの物理デバイス名と論理デバイス名を括弧 <> 内の商品名に対応させています。次の例を参照してください。この方法では、どの論理デバイス名がシステムに接続されたディスクを表しているかをすぐに識別できます。論理デバイス名と物理デバイス名については、[第 9 章](#)を参照してください。

次の例では、ワイルドカードを使用して、追加コントローラに接続されたディスクを表示します。

```
# format /dev/rdisk/c2*
AVAILABLE DISK SELECTIONS:
  0. /dev/rdisk/c2t10d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
     /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
  1. /dev/rdisk/c2t11d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
     /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
  2. /dev/rdisk/c2t14d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@3,0/SUNW,fas@3,8800000/sd@e,0
  3. /dev/rdisk/c2t15d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@3,0/SUNW,fas@3,8800000/sd@f,0
Specify disk (enter its number):
```

次の例では、SPARC システム上のディスクを表示します。

```
# format
0. c0t3d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
   /iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0
Specify disk (enter its number):
```

format の出力は、ディスク 0 (ターゲット 3) が第 1 の SCSI ホストアダプタ (espdma@...) に接続されており、そのホストアダプタは第 1 の SBus デバイス (sbus@0...) に接続されていることを示しています。また、この出力は物理デバイス名と論理デバイス名をディスクの商品名 SUN2.1G に対応しています。

次の例では、x86 システム上のディスクを確認する方法を示します。

```
# format
AVAILABLE DISK SELECTIONS:
  0. c0d0 <DEFAULT cyl 615 alt 2 hd 64 sec 63>
     /pci@0,0/pci-ide@7,1/ata@0/cmdk@0,0
  1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
     /pci@0,0/pci-ide@7,1/ata@0/cmdk@1,0
  2. c1d0 <DEFAULT cyl 817 alt 2 hd 256 sec 63>
     /pci@0,0/pci-ide@7,1/ata@1/cmdk@0,0
Specify disk (enter its number):
```

format の出力は、ディスク 0 が最初の PCI ホストアダプタ (pci-ide@7..) に接続され、このアダプタが ATA デバイス (ata...) に接続されていることを示しています。x86 システム上での format の出力には、ディスクは商品名では表示されません。

参照 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。

- 第 12 章または 第 13 章
- 198 ページの「format.dat のエントリの作成」
- 191 ページの「ディスクラベルを作成する方法」
- ディスクのハードウェアマニュアルを参照して、ディスクをシステムに接続する

---

## ディスクのフォーマット

ディスクはメーカーまたは再販業者によってフォーマットされています。通常は再フォーマットしなくてもドライブを取り付けることができます。

次の作業の前にディスクをフォーマットしておかなければなりません。

- データの書き込み。ただし、ほとんどのディスクはフォーマット済みです。
- Solaris インストールプログラムを使用して行うシステムのインストール。



---

注意 - フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常はメーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、format ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。

---

## ▼ ディスクがフォーマット済みかを調べる方法

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、チェックするディスクの番号を入力します。

```
Specify disk (enter its number): 0
```

4. ディスクがフォーマット済みかを調べます。選択したディスクがフォーマット済みであれば、次のメッセージが表示されます。

```
[disk formatted]
```

### 例 11-2 ディスクがフォーマット済みかを調べる

次の例は、ディスク `c1t0d0` がフォーマット済みであることを示しています。

```
# format /dev/rdisk/c1*
AVAILABLE DISK SELECTIONS:
 0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@0,0
 1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@1,0
 2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@8,0
 3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting /dev/rdisk/c1t0d0s0
[disk formatted]
```

## ▼ ディスクをフォーマットする方法

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、フォーマットするディスクの番号を入力します。

```
Specify disk (enter its number): 0
```



---

注意 - システムディスクを選択しないでください。システムディスクをフォーマットすると、オペレーティングシステムや、システムディスク上のデータがすべて削除されます。

---

4. ディスクのフォーマットを開始するには、**format>** プロンプトで **format** と入力します。 **y** と入力してコマンドを確認します。

```
format> format
Ready to format.  Formatting cannot be interrupted
and takes 23 minutes (estimated).  Continue? yes
```

5. フォーマットが正常に行われたことを、次のメッセージによって確認します。

```
Beginning format.  The current time Tue ABC xx xx:xx:xx xxxx

Formatting...
done

Verifying media...
    pass 0 - pattern = 0xc6dec6de
    2035/12/18

    pass 1 - pattern = 0x6db6db6d
    2035/12/18

Total of 0 defective blocks repaired.
```

### 例 11-3 ディスクをフォーマットする

次の例は、ディスク c0t3d0 のフォーマット方法を示します。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t0d0 <SUNW18G cyl 7506 alt 2 hd 19 sec 248
    /pci@1f,0/pci@1,1/scsi@2/sd@0,0
  1. c0t1d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@1,0
  2. c0t2d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@2,0
  3. c0t3d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@3,0
  4. c0t4d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@4,0
  5. c0t5d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@5,0
  6. c0t6d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@6,0
Specify disk (enter its number): 6
```

```
selecting c0t6d0
[disk formatted]
format> format
Ready to format. Formatting cannot be interrupted
and takes 332 minutes (estimated). Continue? y
Beginning format. The current time is Wed Jan 7 16:16:05 2004

Formatting...
 99% complete (00:00:21 remaining) done

Verifying media...
  pass 0 - pattern = 0xc6dec6de
71132922

  pass 1 - pattern = 0x6db6db6d
71132922

Total of 0 defective blocks repaired.
format> quit
```

---

## 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、211 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 227 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

---

注 - format ユーティリティでは、「スライス」ではなく「パーティション」という用語を使用します。

---

### ▼ ディスクスライス情報を表示する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. **format** ユーティリティを起動します。  

```
# format
```
  3. 画面に表示されたリストから、スライス情報を表示するディスクの番号を入力します。

```
Specify disk (enter its number):1
```

4. パーティションメニューを選択します。

```
format> partition
```

5. 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. **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 11-4 ディスクスライス情報を表示する

次に示すのは、VTOC ラベル付きディスクのスライス情報を表示する例です。

```
# format  
Searching for disks...done  
Specify disk (enter its number):1  
Selecting c0t0d0  
format> partition  
partition> print  
Current partition table (original):  
Total disk cylinders available: 8892 + 2 (reserved cylinders)  
  
Part      Tag      Flag      Cylinders      Size      Blocks  
0         root     wm        1110 - 4687    1.61GB    (0/3578/0) 3381210  
1         swap     wu         0 - 1109      512.00MB  (0/1110/0) 1048950  
2         backup  wm         0 - 8891      4.01GB    (0/8892/0) 8402940  
3         unassigned wm         0              0          (0/0/0) 0  
4         unassigned wm         0              0          (0/0/0) 0  
5         unassigned wm         0              0          (0/0/0) 0  
6         unassigned wm         0              0          (0/0/0) 0  
7         home     wm        4688 - 8891    1.89GB    (0/4204/0) 3972780  
partition> q  
format> q  
#
```

これらの例に表示されるスライス情報についての説明は、[第 10 章](#)を参照してください。

次に示すのは、EFI ラベル付きディスクのスライス情報を表示する例です。

```
# format  
Searching for disks...done  
Specify disk (enter its number): 9
```

```
selecting c4t1d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
partition> q
format> q
Part      Tag      Flag      First Sector      Size      Last Sector
0         root     wm         34                300.00GB  629145633
1         usr      wm         629145634         300.00GB  1258291233
2 unassigned  wm         0                 0         0
3 unassigned  wm         0                 0         0
4 unassigned  wm         0                 0         0
5 unassigned  wm         0                 0         0
6         usr      wm         1258291234        628.77GB  2576924637
8 reserved   wm         2576924638        8.00MB    2576941021
```

---

## ディスクラベルの作成と検査

一般に、ディスクにラベルを付ける操作は、システムのインストール時、または新しいディスクスライスを作成するときに行います。電源障害などが原因でディスクラベルが破損した場合は、ディスクラベルを作成し直さなければならないことがあります。

format ユーティリティは、ラベルが付いていない SCSI ディスクを自動構成しようとします。format ユーティリティがラベルの付いていないディスクを自動構成できる場合、次のようなメッセージが表示されます。

```
c0t0d1: configured with capacity of 4.00GB
```

---

ヒント - 複数のディスクに同じディスクラベルを付ける方法については、[204 ページ](#)の「`prtvtoc` と `fmthard` コマンドを使用して複数のディスクにラベルを付ける」を参照してください。

---

### ▼ ディスクラベルを作成する方法

次の手順に従って、ディスクに VTOC ラベルを付けることができます。または、サイズが 1T バイトを超えるディスクに EFI ラベルを付けることができます。サイズが 1T バイトに満たないディスクに EFI ラベルを付ける場合は、[例 11-6](#) を参照してください。

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、ラベルを作成したいディスクの番号を入力します。

```
Specify disk (enter its number):1
```

4. 次のいずれかを選択します。

- a. ディスクにラベルが付いておらず、正常に構成された場合、手順 5 に進みラベルを付けます。

**format** ユーティリティにより、ディスクラベルを付けるかを尋ねるプロンプトが表示されます。

- b. ディスクにラベルが付いており、そのタイプを変更する場合、または **format** ユーティリティでディスクを自動構成できなかった場合、手順 6 と 7 を実行してディスクタイプを設定し、ラベルを付けます。

5. **Label it now?** プロンプトで **y** と入力して、ディスクにラベルを付けます。

```
Disk not labeled. Label it now? y
```

これでディスクラベルが作成されました。手順 10 に進んで **format** ユーティリティを終了します。

6. **format>** プロンプトで **type** と入力します。

```
format> type
```

Available Drive Types メニューが表示されます。

7. ディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number) [12]: 12
```

または、0 を選択して SCSI-2 ディスクを自動構成します。詳細については、200 ページの「[SCSI ドライブを自動構成する方法](#)」を参照してください。

8. ディスクにラベルを付けます。ディスクにラベルが付いていない場合は、次のメッセージが表示されます。

```
Disk not labeled. Label it now? y
```

ディスクラベルが付いている場合は、次のメッセージが表示されます。

```
Ready to label disk, continue? y
```

9. ディスクラベルを検査します。

```
format> verify
```

10. **format** ユーティリティを終了します。

```
partition> q
```

```
format> q
```

```
#
```



### 例 11-5 ディスクラベルを作成する

次の例では、1.05G バイトのディスクを自動構成してラベルを付ける方法を示します。

```
# format
  clt0d0: 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. clt0d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
    /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
Specify disk (enter its number): 1
Disk not labeled. Label it now? yes
format> verify
#
```

### 例 11-6 サイズが 1T バイトに満たないディスクに EFI ラベルを付ける

次に、format -e コマンドを使って、サイズが 1T バイトに満たないディスクに EFI ラベルを付ける例を示します。階層化されたソフトウェア製品が EFI ラベル付きディスクのシステムでも動作することを確認しておいてください。EFI ラベルの一般的な制限事項については、165 ページの「EFI ディスクラベルの制限」を参照してください。

```
# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  1. clt0d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@0,0
  2. clt1d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@1,0
  3. clt8d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@8,0
  4. clt9d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 4
selecting clt9d0
[disk formatted]
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 1
Ready to label disk, continue? yes
format> quit
```

## ▼ ディスクラベルを検査する方法

ディスクラベル情報の検査には、prtvtoc コマンドを使用します。ディスクラベルの説明と prtvtoc コマンドで表示される情報については、第 10 章を参照してください。

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ディスクラベル情報を表示します。

```
# prtvtoc /dev/rdisk/device-name
```

*device-name* には、検査する raw ディスクデバイスを指定してください。

### 例 11-7 ディスクラベルを検査する

次に示すのは、VTOC ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdisk/c0t0d0s0
* /dev/rdisk/c0t0d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   63 sectors/track
*   15 tracks/cylinder
*   945 sectors/cylinder
*   8894 cylinders
*   8892 accessible cylinders
*
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition Tag  Flags      First      Sector      Last
* Partition Tag  Flags      Sector      Count      Sector  Mount Directory
*   0      2      00      1048950    3381210    4430159  /
*   1      3      01           0      1048950    1048949
*   2      5      00           0      8402940    8402939
*   7      8      00      4430160    3972780    8402939  /export/home
```

次に示すのは、EFI ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdisk/c3t1d0s0
* /dev/rdisk/c3t1d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
* 2479267840 sectors
* 2479267773 accessible sectors
*
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition Tag  Flags      First      Sector      Last
* Partition Tag  Flags      Sector      Count      Sector  Mount Directory
*   0      2      00           34      262144      262177
*   1      3      01      262178      262144      524321
*   6      4      00      524322 2478727100 2479251421
*   8      11     00 2479251422      16384 2479267805
```

---

## 破損したディスクラベルの復元

電源障害やシステム障害が原因で、ディスクが認識されなくなることがあります。このような場合に、必ずしもスライス情報やディスクのデータを作成し直したり、復元しなければならないとは限りません。

破損したディスクラベルを復元する作業の最初の手順は、正しいジオメトリとディスクタイプ情報を使用してディスクにラベルを付けることです。この作業は、通常のディスクラベル作成方法 (自動構成またはディスクタイプの手動指定) で実行できます。

`format` ユーティリティでディスクタイプが認識されたら、次の手順はバックアップラベルを検索してディスクにラベルを付けることです。バックアップラベルを使用してディスクにラベルを付けると、ディスクタイプとジオメトリだけでなく、正しいパーティション情報を使用してディスクラベルが作成されます。

### ▼ 破損したディスクラベルを復元する方法

- 手順 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、『Solaris のシステム管理 (基本編)』の第 10 章「システムのブート (手順)」または『Solaris のシステム管理 (基本編)』の第 11 章「システムのブート (手順)」を参照してください。
2. ディスクのラベルを作成し直します。
- ```
# format
```
- この時点で、`format` はラベルが付いていない SCSI ディスクを自動構成しようとします。ラベルが付いておらず破損したディスクを自動構成できない場合は、次のメッセージが表示されます。
- ```
cwtxdy: configured with capacity of abcMB
```
- 次に、システム上のディスクのリストが表示されます。
3. 画面に表示されたリストから、復元したいディスクの番号を入力します。
- ```
Specify disk (enter its number): 1
```
4. 次のいずれかを選択して、ディスクラベルの作成方法を決定します。
- ディスクが正常に構成された場合、手順 5 と 6 を実行してから、手順 12 に進みます。
  - ディスクが正常に構成されなかった場合、手順 7 - 11 までを実行します。その後、手順 12 に進みます。

5. バックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk or
use the 'backup' command.
Backup label contents:
Volume name = <          >
ascii name = <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
pcyl       = 2038
ncyl       = 2036
acyl       = 2
nhead      = 14
nsect      = 72
Part      Tag      Flag      Cylinders      Size      Blocks
0         root      wm         0 - 300        148.15MB   (301/0/0)  303408
1         swap      wu        301 - 524      110.25MB   (224/0/0)  225792
2         backup    wm         0 - 2035      1002.09MB  (2036/0/0) 2052288
3 unassigned wm         0              0          (0/0/0)    0
4 unassigned wm         0              0          (0/0/0)    0
5 unassigned wm         0              0          (0/0/0)    0
6         usr       wm        525 - 2035     743.70MB   (1511/0/0) 1523088
7 unassigned wm         0              0          (0/0/0)    0
```

6. **format** ユーティリティがバックアップラベルを検出することができ、その内容が適切な場合は、**backup** コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
これで、ディスクラベルが復元されました。手順 12 へ進みます。
```

7. **format** でディスクを自動構成できなかった場合は、**type** コマンドを使用してディスクタイプを指定します。

```
format> type
Available Drives Type メニューが表示されます。
```

8. 0 を選択してディスクを自動構成するか、またはディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number) [12]: 12
```

9. ディスクが正常に自動構成された場合は、ディスクラベルを作成するかどうか尋ねるプロンプトが表示されたときに **no** と応答します。

```
Disk not labeled. Label it now? no
```

10. **verify** コマンドを使用してバックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
```

```
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
```

```
.
.
.
```

11. **format** ユーティリティがバックアップラベルを検出することができ、その内容が適切な場合は、**backup** コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y
Searching for backup labels...found.
Restoring primary label
これで、ディスクラベルが復元されました。
```

12. **format** ユーティリティを終了します。

```
format> q
```

13. **fsck** コマンドを使用して、復元されたディスク上のファイルシステムを確認します。

**fsck** コマンドの使用方法については、[第 20 章](#)を参照してください。

---

## Sun 製品以外のディスクの追加

Solaris 環境では、Sun 製品以外の多数のディスクがサポートされます。ただし、ディスクを認識させるためには、デバイスドライバか、`format.dat` エントリのいずれか 1 つ、またはその両方を用意しなければならない場合があります。ディスク追加時のその他のオプションを、次に示します。

- SCSI ディスクを追加する場合、**format** ユーティリティの自動構成機能の使用を試みることができます。詳細については、[199 ページ](#)の「[SCSI ディスクドライブの自動構成](#)」を参照してください。
- PCI、SCSI、または USB ディスクのホットプラグを試みることもできます。詳細については、[第 5 章](#)を参照してください。

Sun 以外のディスクが標準の SunOS 互換デバイスドライバで機能するように設計されている場合は、適切な `format.dat` エントリを作成するだけで、ディスクは **format** ユーティリティに認識されるはずです。それ以外の場合は、そのディスクをサポートするために Sun 以外のデバイスドライバをロードする必要があります。

---

注 – Sun の `format` ユーティリティが Sun 以外の製品のどのディスクドライバでも正常に機能するとは限りません。ディスクドライバに Solaris の `format` ユーティリティとの互換性がない場合は、ディスクドライブのベンダーが独自のフォーマットプログラムを提供しているはずですが。

---

ここでは、ソフトウェアサポートのいずれかが不足している場合に必要な作業について説明します。一般に、`format` ユーティリティを起動し、ディスクタイプが認識されないなどという場合に、不足しているソフトウェアサポートがあることがわかります。

この節の説明にしたがって不足しているソフトウェアを用意してから、[第 12 章](#)または[第 13 章](#)で説明されている、システムディスクまたは二次ディスクを構成する手順を参照してください。

## format.dat のエントリの作成

認識されないディスクは、そのディスクのジオメトリと運用パラメータに関する正確な情報がなければフォーマットできません。この情報は、`/etc/format.dat` ファイル内で指定します。

---

注 – SCSI-2 ドライブには `format.dat` のエントリは不要です。再構成ブート時にドライブに電源が投入されていれば、`format` ユーティリティは SCSI-2 ドライブを自動的に構成します。SCSI ディスクドライブを自動構成する手順については [200 ページ](#)の「[SCSI ドライブを自動構成する方法](#)」を参照してください。

---

ディスクが認識されない場合は、テキストエディタを使用して `format.dat` にディスクのエントリを作成します。作業を始める前に、ディスクとそのコントローラに関連するすべての技術仕様を収集する必要があります。この情報はディスクと一緒に提供されているはずですが、提供されない場合は、ディスクメーカーまたは購入先に問い合わせてください。

### ▼ format.dat のエントリを作成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. `/etc/format.dat` ファイルのコピーを作成します。

```
# cp /etc/format.dat /etc/format.dat.gen
```
  3. [第 14 章](#)で説明する `format.dat` 情報を使用して、`/etc/format.dat` ファイルに、Sun 以外の製品のディスクのエントリを入力します。

ディスクのハードウェア製品マニュアルを参照して、必要な情報を収集してください。

## SCSI ディスクドライブの自動構成

/etc/format.dat ファイルに特定のドライブタイプが含まれていない場合でも、format ユーティリティは SCSI ディスクドライブを自動的に構成します。この機能によって、ディスクデバイスモード検知に関する SCSI-2 仕様に準拠しているディスクドライブで、フォーマット、スライス作成、およびラベル作成を行うことができます。

ディスクの追加に関するその他のオプションを、次に示します。

- SCSI ディスクを追加する場合、format ユーティリティの自動構成機能の使用を試みることができます。詳細については、[199 ページの「SCSI ディスクドライブの自動構成」](#)を参照してください。
- PCI、SCSI、または USB ディスクのホットプラグを試みることもできます。詳細については、[第 5 章](#)を参照してください。

自動構成を使用して SCSI ドライブを構成する場合、次の手順を実行します。

- システムをシャットダウンする。
- SCSI ディスクドライブをシステムに接続する。
- ディスクドライブの電源をオンにする。
- 再構成用ブートを実行する。
- format ユーティリティを使用して SCSI ディスクドライブを自動構成する。

再構成ブートを実行した後に、format ユーティリティを呼び出すと、format はディスクを構成しようとします。成功すると、ディスクが構成されたことを示すメッセージを表示します。SCSI ディスクドライブを自動構成する手順については、[200 ページの「SCSI ドライブを自動構成する方法」](#)を参照してください。

1.3G バイトの SCSI ディスクドライブに関して、format ユーティリティが表示するパーティションテーブルの例を、次に示します。

| Part | Tag    | Flag | Cylinders  | Size     | Blocks     |
|------|--------|------|------------|----------|------------|
| 0    | root   | wm   | 0 - 96     | 64.41MB  | (97/0/0)   |
| 1    | swap   | wu   | 97 - 289   | 128.16MB | (193/0/0)  |
| 2    | backup | wu   | 0 - 1964   | 1.27GB   | (1965/0/0) |
| 6    | usr    | wm   | 290 - 1964 | 1.09GB   | (1675/0/0) |

SCSI 自動構成機能の使用方法については、[第 14 章](#)を参照してください。

## ▼ SCSI ドライブを自動構成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. システムのブート時に読み込まれる **/reconfigure** ファイルを作成します。  

```
# touch /reconfigure
```
  3. システムをシャットダウンします。  

```
# shutdown -i0 -gn -y
```

-in システムを init レベル 0 (電源切断) にする。  
-gn ログインしているユーザーに、*n* 秒後にシステムのシャットダウンを開始することを通知する。  
-y ユーザーの介入なしでコマンドを実行するように指定する。  
システムのシャットダウン後に、ok 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、ディスクの背面にそのための小型のスイッチが付いています。
  6. ディスクがシステムに正しく接続されているかどうかを確認します。  
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
  7. すべての外部周辺デバイスの電源を入れます。
  8. システムの電源を入れます。  
システムがブートし、ログインプロンプトが表示されます。
  9. スーパーユーザーになるか、同等の役割を引き受けて、再度ログインします。
  10. **format** ユーティリティを起動して、自動構成するディスクを選択します。  

```
# format
Searching for disks...done
c1t0d0: configured with capacity of 1002.09MB
AVAILABLE DISK SELECTIONS:
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 1
```
  11. プロンプトで **y** と入力してディスクにラベルを付けます。



y と入力すると、SCSI 自動構成機能により、ディスクラベルの生成およびディスクへの書き込みが実行されます。

```
Disk not labeled. Label it now? y
```

## 12. ディスクラベルを検査します。

```
format> verify
```

## 13. format ユーティリティを終了します。

```
format> q
```

---

# 欠陥セクターの修復

システム上のディスクに欠陥セクターが存在する場合は、次の手順にしたがって修復できます。欠陥セクターを発見するのは次のような場合です。

### ■ ディスク上で表面解析を実行した場合

format ユーティリティの解析機能については、236 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、202 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に出力されます。

```
WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):  
  Error for command 'read' Error Level: Retryable  
  Requested Block 126, Error Block: 179  
  Sense Key: Media Error  
  Vendor 'name':  
  ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
```

上記のコンソールメッセージは、ブロック 179 が不良であることを示しています。format ユーティリティの repair コマンドを使用して欠陥ブロックを配置し直すか、修復オプションを有効にして analyze コマンドを実行してください。

## ▼ 表面解析を使用して欠陥セクターを調べる方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 欠陥セクターの存在するスライス内のファイルシステムをマウント解除します。  

```
# umount /dev/dsk/device-name
```

詳細については、mount (1M) のマニュアルページを参照してください。
  3. **format** ユーティリティを起動します。  

```
# format
```
  4. 調べるディスクを選択します。  

```
Specify disk (enter its number):1
selecting c0t2d0:
[disk formatted]
Warning: Current Disk has mounted partitions.
```
  5. **analyze** メニューを選択します。  

```
format> analyze
```
  6. **analyze>** プロンプトで **setup** と入力して、検索手順に使用する解析パラメータを設定します。  
次のパラメータを使用してください。  

```
analyze> setup
Analyze entire disk [yes]? n
Enter starting block number [0, 0/0/0]: 12330
Enter ending block number [2052287, 2035/13/71]: 12360
Loop continuously [no]? y
Repair defective blocks [yes]? n
Stop after first error [no]? n
Use random bit patterns [no]? n
Enter number of blocks per transfer [126, 0/1/54]: 1
Verify media after formatting [yes]? y
Enable extended messages [no]? n
Restore defect list [yes]? y
Create defect label [yes]? y
```
  7. **read** コマンドを使用して欠陥を見つけます。  

```
analyze> read
Ready to analyze (won't harm SunOS). This takes a long time,
but is interruptible with Control-C. Continue? y
    pass 0
    2035/12/1825/7/24
    pass 1
Block 12354 (18/4/18), Corrected media error (hard data ecc)
    25/7/24
^C
```

```
Total of 1 defective blocks repaired.
```

## ▼ 欠陥セクターを修復する方法

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **format** ユーティリティを起動します。

```
# format
```

3. 欠陥セクターの存在するディスクを選択します。

```
Specify disk (enter its number): 1
selecting c0t3d0
[disk formatted]
format>
```

4. **repair** コマンドを選択します。

```
format> repair
```

5. 欠陥ブロック番号を入力します。

```
Enter absolute block number of defect: 12354
Ready to repair defect, continue? y
Repairing block 12354 (18/4/18)...ok.
format>
```

欠陥セクターの特定に使う形式がわからない場合は、202 ページの「表面解析を使用して欠陥セクターを調べる方法」を参照してください。

---

## ディスク管理のヒント

次のヒントに従って、ディスクの管理効率を高めることができます。

### format セッションのデバッグ

**format -M** と入力して、ATA および SCSI デバイス用の拡張および診断メッセージを有効にします。

この例の **Inquiry:** の下の数値の列は、数値の右側に表示される **inquiry** データの 16 進数値です。

```

# format -M
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 0
selecting c0t3d0
[disk formatted]
format> inquiry
Inquiry:
00 00 02 02 8f 00 00 12 53 45 41 47 41 54 45 20      .....NAME....
53 54 31 31 32 30 30 4e 20 53 55 4e 31 2e 30 35      ST11200N SUN1.05
38 33 35 38 30 30 30 33 30 32 30 39 00 00 00 00      835800030209....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      .....
00 43 6f 70 79 72 69 67 68 74 20 28 63 29 20 31      .Copyright (c) 1
39 39 32 20 53 65 61 67 61 74 65 20 41 6c 6c 20      992 NAME All
72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 20      rights reserved
30 30 30   000
Vendor:      name
Product:     ST11200N SUN1.05
Revision:    8358
format>

```

## prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける

prtvtoc コマンドと fmthard コマンドを使用して、同じディスクジオメトリを持つ複数のディスクにラベルを付けます。

この for ループをスクリプト内で使用して、1 台のディスクからディスクラベルをコピーし、複数のディスク上で複製します。

```

# for i in xyz
> do
> prtvtoc /dev/rdisk/cwtxdysz | fmthard -s - /dev/rdisk/cwt${i}d0s2
> done

```

例 11-8 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

```

例 11-8 複数のディスクにラベルを付ける (続き)

```
> done
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
#
```



## 第 12 章

# SPARC: ディスクの追加 (手順)

この章では、SPARC システムにディスクを追加する手順について説明します。

SPARC システムへのディスクの追加に関連した手順については、207 ページの「[SPARC: システムディスクまたは二次ディスクの追加 \(作業マップ\)](#)」を参照してください。

ディスク管理の概要については、[第 10 章](#)を参照してください。x86 システムにディスクを追加する手順については、[第 13 章](#)を参照してください。

---

## SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)

次の作業マップは、SPARC システムにディスクを追加する手順を示します。

| 作業             | 説明                                                                | 参照先                                                     |
|----------------|-------------------------------------------------------------------|---------------------------------------------------------|
| 1. ディスクの接続とブート | システムディスク<br>新しいディスクを接続して、ローカルまたはリモートの Solaris CD または DVD からブートする。 | 209 ページの「 <a href="#">SPARC: システムディスクを接続してブートする方法</a> 」 |
|                | 二次ディスク<br>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。              | 209 ページの「 <a href="#">SPARC: 二次ディスクを接続してブートする方法</a> 」   |

| 作業                 | 説明                                                                                                                                | 参照先                                          |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| 2. スライスとディスクラベルの作成 | ディスクスライスを作成してディスクにラベルを付ける (ディスクメーカーにより実行されていない場合)。                                                                                | 211 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」     |
| 3. ファイルシステムの作成     | <code>newfs</code> コマンドを使用してディスクスライス上に UFS ファイルシステムを作成する。システムディスクの場合はルート (/) または <code>/usr</code> ファイルシステム、あるいはその両方を作成しなければならない。 | 215 ページの「SPARC: UFS ファイルシステムを作成する方法」         |
| 4. ファイルシステムの復元     | システムディスク上にルート (/) または <code>/usr</code> ファイルシステム、あるいはその両方を復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。                                      | 第 25 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。                                                                      | 216 ページの「SPARC: システムディスクにブートブロックをインストールする方法」 |

## SPARC: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと `/usr` ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元する。

二次ディスクには、ルート (/) ファイルシステムも `/usr` ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。



## ▼ SPARC: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

- 手順
1. 損傷したシステムディスクをシステムから外します。
  2. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。  
通常は、ディスクの背面にそのための小型スイッチが付いています。
  3. 交換用のディスクをシステムに正しく接続します。  
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
  4. ローカルの **Solaris CD/DVD** またはリモートの **Solaris CD/DVD** (ネットワーク経由) のどちらでブートするかに応じて、次の表の手順で操作します。

| ブート方法                        | 作業                                                                                                        |
|------------------------------|-----------------------------------------------------------------------------------------------------------|
| ローカルドライブの Solaris CD または DVD | 1. CD または DVD がドライブに入っていることを確認する。<br><br>2. メディアからシングルユーザーモードでブートする。<br><br><code>ok boot cdrom -s</code> |
| ネットワーク経由                     | ネットワークからシングルユーザーモードでブートする。<br><br><code>ok boot net -s</code>                                             |

数分後に、スーパーユーザーのプロンプト (#) が表示されます。

- 参照 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。211 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

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. ディスクタイプが **Solaris** ソフトウェアでサポートされていない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。

ディスク用の `format.dat` エントリを作成する方法については、198 ページの「`format.dat` のエントリを作成する方法」を参照してください。

3. システムのブート時に読み込まれる `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

`/reconfigure` ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0 システムを実行レベル 0 (電源切断状態) にする。

-gn ログインしているユーザーに、*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。

Solaris オペレーティングシステムのシャットダウン後に、`ok` プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。  
通常は、ディスクの背面にそのための小型スイッチが付いています。
7. ディスクがシステムに正しく接続されているかどうかを確認します。  
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。  
システムがブートし、ログインプロンプトが表示されます。

参照 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。211 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

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **format** ユーティリティを起動します。

```
# format
```

利用可能なディスクのリストが表示されます。詳細については、`format (1M)` のマニュアルページを参照してください。

3. 画面に表示されるリストから、パーティション分割をし直したいディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

`disk-number` は、パーティションの再分割を実行するディスクの番号です。

4. **partition** メニューを選択します。

```
format> partition
```

5. 現在のパーティション (スライス) テーブルを表示します。

```
partition> print
```

6. 変更作業を開始します。

```
partition> modify
```

7. ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]? 1
```

**free hog** スライスの詳細については、[182 ページの「free hog スライスの使用方法」](#)を参照してください。

8. 続行するかどうかを尋ねるプロンプトが表示されたら、**y** と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition table based on  
above table[yes]? y
```

9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0) およびスワップ (スライス 1) (必須)
- /usr (スライス 6)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。

ディスクスライスの作成例については、[例 12-1](#) を参照してください。

10. プロンプトが表示されたら **y** と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? y
```

現在のパーティションテーブルが希望どおりでないために変更したい場合は、no と応答して手順 6 に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

*partition-name* には、新しいパーティションテーブルの名前を指定します。

12. 新しいディスク上でスライスの割り当てが終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. **partition** メニューを終了します。

```
partition> q
```

14. ディスクラベルを検査します。

```
format> verify
```

15. **format** メニューを終了します。

```
format> q
```

### 例 12-1 SPARC: システムディスクのディスクスライスとラベルを作成する

次の例では、**format** ユーティリティを使用して 18G バイトのディスクを 3 つのスライスに分割します。各スライスをルート (/) ファイルシステム、スワップ領域、**/usr** ファイルシステムに割り当てます。

```
# format
AVAILABLE DISK SELECTIONS:
  0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@0,0
  1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@1,0
  2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@8,0
  3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting c1t0d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  0. Current partition table (original)
```

```

1. All Free Hog
Part      Tag      Flag      Cylinders      Size      Blocks
0         root      wm         0                0          (0/0/0)      0
1         swap      wu         0                0          (0/0/0)      0
2         backup    wu         0 - 7505        16.86GB    (7506/0/0)  35368272
3         unassigned wm         0                0          (0/0/0)      0
4         unassigned wm         0                0          (0/0/0)      0
5         unassigned wm         0                0          (0/0/0)      0
6         usr       wm         0                0          (0/0/0)      0
7         unassigned wm         0                0          (0/0/0)      0

```

```

Choose base (enter number) [0]? 1
table based on above table[yes]? yes
Free Hog partition[6]? 6

```

```

Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

```

```

Part      Tag      Flag      Cylinders      Size      Blocks
0         root      wm         0 - 1780        4.00GB    (1781/0/0)  8392072
1         swap      wu         1781 - 3561     4.00GB    (1781/0/0)  8392072
2         backup    wu         0 - 7505        16.86GB    (7506/0/0)  35368272
3         unassigned wm         0                0          (0/0/0)      0
4         unassigned wm         0                0          (0/0/0)      0
5         unassigned wm         0                0          (0/0/0)      0
6         usr       wm         3562 - 7505     8.86GB    (3944/0/0)  18584128
7         unassigned wm         0                0          (0/0/0)      0

```

```

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "disk0"
Ready to label disk, continue? yes
partition> quit
format> verify
format> quit

```

## 例 12-2 SPARC: 二次ディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティを使用して 18G バイトのディスクを 1 つのスライスに分割し、それを /export/home ファイルシステムに割り当てます。

```

# format /dev/rdisk/c1*
AVAILABLE DISK SELECTIONS:
0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@0,0
1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@1,0
2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@8,0
3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
   /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 1

```

```

selecting c1t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  0. Current partition table (original)
  1. All Free Hog
Choose base (enter number) [0]? 1
Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0              0      (0/0/0)      0
  1      swap      wu         0              0      (0/0/0)      0
  2      backup    wu        0 - 7505      16.86GB  (7506/0/0) 35368272
  3 unassigned  wm         0              0      (0/0/0)      0
  4 unassigned  wm         0              0      (0/0/0)      0
  5 unassigned  wm         0              0      (0/0/0)      0
  6      usr      wm         0              0      (0/0/0)      0
  7 unassigned  wm         0              0      (0/0/0)      0

Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 7
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0              0      (0/0/0)      0
  1      swap      wu         0              0      (0/0/0)      0
  2      backup    wu        0 - 7505      16.86GB  (7506/0/0) 35368272
  3 unassigned  wm         0              0      (0/0/0)      0
  4 unassigned  wm         0              0      (0/0/0)      0
  5 unassigned  wm         0              0      (0/0/0)      0
  6      usr      wm         0              0      (0/0/0)      0
  7 unassigned  wm        0 - 7505      16.86GB  (7506/0/0) 35368272

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "home"
Ready to label disk, continue? y
partition> q
format> verify
format> q
#

```

次の例では、format ユーティリティを使って EFI ラベル付きの 1.15T バイトのディスクを 3 つのスライスに分割します。

```

# format
.
.
.
partition> modify
Select partitioning base:

```

```

0. Current partition table (original)
1. All Free Hog
Choose base (enter number) [0]? 1
Part      Tag      Flag      First Sector      Size      Last Sector
0         root     wm         0                  0          0
1         usr      wm         0                  0          0
2 unassigned wm         0                  0          0
3 unassigned wm         0                  0          0
4 unassigned wm         0                  0          0
5 unassigned wm         0                  0          0
6         usr      wm         0                  0          0
8 reserved wm         2576924638        8.00MB     2576941021
Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition 0 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 1 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 2 [0b, 34e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 3 [0b, 838860834e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 5 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Enter size of partition 6 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Part      Tag      Flag      First Sector      Size      Last Sector
0 unassigned wm         0                  0          0
1 unassigned wm         0                  0          0
2         usr      wm         34                400.00GB   838860833
3         usr      wm         838860834         400.00GB   1677721633
4         usr      wm         1677721634        428.77GB   2576924637
5 unassigned wm         0                  0          0
6 unassigned wm         0                  0          0
8 reserved wm         2576924638        8.00MB     2576941021
Ready to label disk, continue? yes

partition> q

```

参照 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。215 ページの「SPARC: UFS 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: UFS ファイルシステムを作成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```

/dev/rdisk/cwtxdysz は、作成するファイルシステムの raw デバイスです。

newfs コマンドの詳細については、第 16 章または newfs (1M) のマニュアルページを参照してください。

- マウントして、新規ファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls lost+found
```

- 参照
- 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ります。第 25 章を参照してください。
  - 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します。216 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」を参照してください。
  - 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ります。第 25 章を参照してください。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ります。
  - 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 17 章を参照してください。

## ▼ 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

- 手順
- スーパーユーザーになるか、同等の役割を引き受けます。
  - ブートブロックをシステムディスクにインストールします。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/cwtxdys0
/usr/platform/`uname -i`/lib/fs /ufs/bootblk
ブートブロックコード
```

```
/dev/rdisk/cwtxdys0
ルート (/) ファイルシステムの raw デバイス
```

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

- 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
```

### 例 12-3 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

次の例では、ブートブロックを Ultra10 システムにインストールする方法を示します。

```
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0
```



## 第 13 章

### x86: ディスクの追加 (手順)

この章では、x86 システムにディスクを追加する手順について説明します。

x86 システムへのディスク追加に関連した手順については、217 ページの「x86: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

ディスク管理の概要については、第 10 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 12 章を参照してください。

---

### x86: システムディスクまたは二次ディスクの追加 (作業マップ)

| 作業             | 説明                                                                | 参照先                                 |
|----------------|-------------------------------------------------------------------|-------------------------------------|
| 1. ディスクの接続とブート | システムディスク<br>新しいディスクを接続して、ローカルまたはリモートの Solaris CD または DVD からブートする。 | 219 ページの「x86: システムディスクを接続してブートする方法」 |
|                | 二次ディスク<br>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。              | 219 ページの「x86: 二次ディスクを接続してブートする方法」   |

| 作業                 | 説明                                                                                                       | 参照先                                                                                   |
|--------------------|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 2. スライスとディスクラベルの作成 | ディスクスライスを作成してディスクにラベルを付ける(ディスクメーカーにより実行されていない場合)。                                                        | 221 ページの「x86: Solaris fdisk パーティションを作成する方法」および 227 ページの「x86: ディスクスライスとディスクラベルを作成する方法」 |
| 3. ファイルシステムの作成     | newfs コマンドを使用してディスクスライス上に UFS ファイルシステムを作成する。システムディスクの場合はルート (/) または /usr ファイルシステム (あるいは両方) を作成しなければならない。 | 228 ページの「x86: ファイルシステムを作成する方法」                                                        |
| 4. ファイルシステムの復元     | システムディスク上にルート (/) または /usr ファイルシステム (あるいは両方) を復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。                         | 第 25 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。                                             | 229 ページの「x86: システムディスクにブートブロックをインストールする方法」                                            |

## x86: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと /usr ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元する。

二次ディスクには、ルート (/) ファイルシステムも /usr ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。

## ▼ x86: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

- 手順
1. 損傷したシステムディスクをシステムから外します。
  2. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。  
通常は、ディスクの背面にそのための小型スイッチが付いています。
  3. 交換用のディスクをシステムに正しく接続します。  
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
  4. ローカルとリモートのどちらの **Solaris CD** (または **DVD**) からブートするかに応じて、手順 **a** から **e** を実行します。  
ネットワークからブートする場合は、手順 **a** をスキップします。
    - a. ローカルの **Solaris CD** または **DVD** からブートする場合は、ドライブに **Solaris 9 Installation CD** または **DVD** を挿入します。
    - b. **Solaris** ブートフロッピーディスクを主フロッピーディスクドライブ (**DOS** ドライブ **A**) に挿入します。
    - c. **Type any key to continue** プロンプトが表示されたら、任意のキーを押してシステムをリブートします。システムがシャットダウンされている場合は、リセットボタンを押してシステムを再起動します。  
数分後に **Boot Solaris** 画面が表示されます。
    - d. **Boot Solaris** 画面から、ブートデバイスを **CD-ROM** ドライブにするか、ネットワークにするかを選択します。  
**Current Boot Parameters** 画面が表示されます。
    - e. システムをシングルユーザーモードでブートします。  
  
Select the type of installation: **b -s**  
数分後に、スーパーユーザーのプロンプト (**#**) が表示されます。

参照 システムのブート後に、**fdisk** パーティションを作成できます。221 ページの「[x86: Solaris fdisk パーティションを作成する方法](#)」を参照してください。

## ▼ x86: 二次ディスクを接続してブートする方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。

2. ディスクが **Solaris** ソフトウェアでサポートされない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。
3. システムのブート時に読み込まれる **/reconfigure** ファイルを作成します。

```
# touch /reconfigure
```

/reconfigure ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0 システムを実行レベル 0 (電源切断状態) にする。

-gn ログインしているユーザーに、*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。

Type any key to continue プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。  
通常は、ディスクの背面にそのための小型スイッチが付いています。
7. ディスクがシステムに正しく接続されているかどうかを確認します。  
インストールの詳細については、ディスクのハードウェアインストールガイドを参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。  
システムがブートし、ログインプロンプトが表示されます。

参照 システムのブート後に、`fdisk` パーティションを作成できます。221 ページの「[x86: Solaris fdisk パーティションを作成する方法](#)」を参照してください。

## x86: fdisk パーティションの作成上のガイドライン

次のガイドラインに従って `fdisk` パーティションを設定してください。

- ディスクは最大 4 つの `fdisk` パーティションに分割できます。いずれか 1 つのパーティションを Solaris パーティションにしなければなりません。

- Solaris パーティションをディスク上でアクティブなパーティションにしなければなりません。アクティブなパーティションとは、システム起動時にデフォルトでオペレーティングシステムがブートされるパーティションです。
- Solaris の `fdisk` パーティションは、シリンダ境界から開始しなければなりません。
- 最初のディスクの先頭のセクターには、ブート情報 (マスターブートレコードを含む) が書き込まれるので、最初のディスクの 1 番目の `fdisk` パーティションとして、Solaris の `fdisk` パーティションを作成する場合は、ディスクのシリンダ 0 ではなくシリンダ 1 から開始しなければなりません。
- Solaris の `fdisk` パーティションにディスク全体を使用するか、それより小さくして DOS パーティションに使用する余地を残すことができます。また、既存のパーティションに影響を与えずに、ディスク上に新しい `fdisk` パーティションを作成できます (それを作成する余地がある場合)。

---

**x86** のみ – Solaris スライスとはパーティションと呼ばれることがあります。このマニュアルでは「スライス」という用語を使用しますが、Solaris のマニュアルやプログラムによっては、スライスを「パーティション」と呼ぶ場合があります。

混乱を避けるために、Solaris のマニュアルでは、`fdisk` パーティション (x86 版 Solaris でのみサポート) と、スライスまたはパーティションと呼ばれる、Solaris の `fdisk` パーティション内をさらに分割したものとを区別しています。

---

## ▼ x86: Solaris `fdisk` パーティションを作成する方法

手順 1. 220 ページの「**x86: `fdisk` パーティションの作成上のガイドライン**」を参照します。

2. スーパーユーザーになるか、同等の役割を引き受けます。

3. `format` ユーティリティを起動します。

```
# format
```

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

4. 画面に表示されたリストから、**Solaris `fdisk`** パーティションを作成するディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

*disk-number* は、Solaris `fdisk` パーティションを作成するディスクの番号です。

5. `fdisk` メニューを選択します。

```
format> fdisk
```

表示される fdisk メニューは、fdisk パーティションがすでにディスク上に存在しているかどうかによって異なります。次の表を使用して、次に行う手順を決定してください。

| 作業                                                             | 次の手順 | 参照先    |
|----------------------------------------------------------------|------|--------|
| ディスク全体を占有する Solaris fdisk パーティションを作成する。                        | 手順 6 | 例 13-1 |
| 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 | 例 13-2 |
| 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 | 例 13-3 |

6. ディスク全体にまたがる **Solaris fdisk** パーティションを作成してそれをアクティブにするには、プロンプトで **y** を入力します。次に、手順 **14** に進みます。

The recommended default partitioning for your disk is:

```
a 100% "SOLARIS System" partition.
```

```
To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will
let you select other partitions. y
```

7. ディスク全体にまたがる **Solaris fdisk** パーティションを作成しない場合は、プロンプトで **n** を入力します。

```
To select this, please type "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
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exit without updating disk configuration)

Enter Selection: 1

9. 「1(=Solaris)」を選択して、**Solaris fdisk** パーティションを作成します。

```

Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ? 1

```

10. **Solaris fdisk** パーティション用に割り当てるディスクのパーセントを指定します。このパーセントを計算するときには、既存の **fdisk** パーティションのサイズを考慮してください。

```

Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). mm

```

11. プロンプトで **y** を入力して、**Solaris fdisk** パーティションをアクティブにします。

```

Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y

```

**fdisk** パーティションがアクティブになった後、Enter Selection: プロンプトが表示されます。

12. 別の **fdisk** パーティションを作成する場合は、「1. Create a partition」を選択します。

**fdisk** パーティションを作成する手順については、手順 9 ~ 11 を参照してください。

13. ディスク構成を更新し、**Selection** メニューから **fdisk** メニューに戻ります。

```

Selection: 4

```

14. **label** コマンドを使用して、ディスクに新しいラベルを付けます。

```

WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format>

```

15. **format** メニューを終了します。

```

format> quit

```

### 例 13-1 x86: 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
```

### 例 13-2 x86: 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する

次の例は、DOS-BIG fdisk パーティションがすでに存在しているディスクに、Solaris fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
Total disk size is 2694 cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
1          DOS-BIG   1        538    538    20
SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exit without updating disk configuration)
Enter Selection: 1
```



```

Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders

      Cylinder size is 765 (512 byte) blocks
                        Cylinders
Partition  Status   Type      Start  End   Length  %
=====  =====  =====  =====  ===  =====  ==
          1          DOS-BIG      1    538    538    20
          2      Active   SOLARIS   539  2693   2155    80
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Change Active (Boot from) partition
  3. Delete a partition
  4. Exit (Update disk configuration and exit)
  5. Cancel (Exit without updating disk configuration)
Enter Selection: Selection: 4
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> q

```

### 例 13-3 x86: Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する

次の例は、Solaris fdisk パーティションと DOSBIG fdisk パーティションを作成する方法を説明しています。

```

format> fdisk
The recommended default partitioning for your disk is:
  a 100% "SOLARIS System" partition.
To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
                        Cylinders
Partition  Status   Type      Start  End   Length  %
=====  =====  =====  =====  ===  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Change Active (Boot from) partition
  3. Delete a partition
  4. Exit (Update disk configuration and exit)
  5. Cancel (Exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)

```

```

(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?8
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 20
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". n
Total disk size is 2694 cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ===
      1                DOS-BIG      1  538    538    20
SELECT ONE OF THE FOLLOWING:
  1.  Create a partition
  2.  Change Active (Boot from) partition
  3.  Delete a partition
  4.  Exit (Update disk configuration and exit)
  5.  Cancel (Exit without updating disk configuration)Enter
Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ===
      1                DOS-BIG      1  538    538    20
      2      Active  SOLARIS     539 2693   2155    80
SELECT ONE OF THE FOLLOWING:
  1.  Create a partition
  2.  Change Active (Boot from) partition
  3.  Delete a partition
  4.  Exit (Update disk configuration and exit)
  5.  Cancel (Exit without updating disk configuration)
Enter Selection: 4
format> q

```

参照 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。227 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

## ▼ x86: ディスクスライスとディスクラベルを作成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. **format** ユーティリティを起動します。  

```
# format
```
  3. 画面に表示されるリストから、パーティション分割をし直したいディスクの番号を入力します。  

```
Specify disk (enter its number): disk-number
```

*disk-number* は、パーティションの再分割を実行するディスクの番号です。
  4. **partition** メニューを選択します。  

```
format> partition
```
  5. 現在のパーティション (スライス) テーブルを表示します。  

```
partition> print
```
  6. 変更作業を開始します。  

```
partition> modify
```
  7. ディスクをすべて **free hog** に設定します。  

```
Choose base (enter number) [0]? 1
```

**free hog** スライスの詳細については、[182 ページの「free hog スライスの使用方法」](#)を参照してください。
  8. 続行するかどうかを尋ねるプロンプトが表示されたら、**yes** と応答して新しいパーティションテーブルを作成します。  

```
Do you wish to continue creating a new partition table based on above table[yes]? yes
```
  9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。  
システムディスクを追加するときは、次のスライスを設定しなければなりません。
    - ルート (スライス 0)、スワップ (スライス 1) (必須) および
    - /usr (スライス 6)スライスの情報を設定すると、新しいパーティションテーブルが表示されます。
  10. プロンプトが表示されたら **yes** と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。  

```
Okay to make this the current partition table[yes]? yes
```

表示されたパーティションテーブルが希望どおりでないために変更したい場合は、no と応答して手順 6 に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

*partition-name* には、新しいパーティションテーブルの名前を指定します。

12. 新しいディスク上でスライスの割り当てが終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. **partition** メニューを終了します。

```
partition> quit
```

14. 新しいディスクラベルを検査します。

```
format> verify
```

15. **format** メニューを終了します。

```
format> quit
```

参照 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。228 ページの「x86: ファイルシステムを作成する方法」を参照してください。

## ▼ x86: ファイルシステムを作成する方法

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```

*/dev/rdisk/cwtxdysz* は、作成するファイルシステムの raw デバイスです。

*newfs* コマンドの詳細については、第 16 章または *newfs (1M)* のマニュアルページを参照してください。

3. マウントして、新規ファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
```

```
# ls /mnt
```

```
lost+found
```

参照 ■ 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ります。第 25 章を参照してください。

■ 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します。229 ページの「x86: システムディスクにブートブロックをインストールする方法」を参照してください。

- 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ります。第 25 章を参照してください。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ります。
- 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 17 章を参照してください。

## ▼ x86: システムディスクにブートブロックをインストールする方法

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. ブートブロックをシステムディスクにインストールします。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/
`uname -i` /lib/fs/ufs/bootblk /dev/rdsk/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/rdsk/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
```

例 13-4 x86: システムディスクにブートブロックをインストールする

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdsk/c0t6d0s2
```



## 第 14 章

# format ユーティリティ (参照情報)

---

この章では、format ユーティリティのメニューとコマンドについて説明します。

この章の内容は次のとおりです。

- 231 ページの「format ユーティリティを使用する上での推奨事項および要件」
- 232 ページの「format のメニューとコマンドの説明」
- 238 ページの「format.dat ファイル」
- 243 ページの「format コマンドへの入力規則」
- 245 ページの「format ユーティリティのヘルプを利用する」

format ユーティリティの概要については、第 10 章を参照してください。

---

## format ユーティリティを使用する上での推奨事項および要件

format ユーティリティを使用するには、スーパーユーザーになるか、同等の役割を引き受ける必要があります。そうではない場合、format ユーティリティを使用しようとするとき次のエラーメッセージが表示されます。

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

既存のデータを維持しつつ format ユーティリティを使用する場合、次の指針に従って操作を行なってください。

- ディスクドライブ上のすべてのファイルのバックアップを作成します。
- format ユーティリティの dump コマンドを使用して、欠陥領域リストをファイルに保存します。ファイル名には、ドライブタイプ、モデル番号、シリアル番号を含めておくべきです。

- メーカーから出荷時にドライブといっしょに提供された、欠陥領域リストを保管します。

## format のメニューとコマンドの説明

format のメインメニューは次のようになっています。

```

FORMAT MENU:
  disk      - select a disk
  type      - select (define) a disk type
  partition - select (define) a partition table
  current   - describe the current disk
  format    - format and analyze the disk
  repair    - repair a defective sector
  label     - write label to the disk
  analyze   - surface analysis
  defect    - defect list management
  backup    - search for backup labels
  verify    - read and display labels
  save      - save new disk/partition definitions
  inquiry   - show vendor, product and revision
  volname   - set 8-character volume name
  quit
format>

```

次の表に、format のメインメニュー項目を示します。

表 14-1 format のメインメニュー項目の説明

| 項目        | コマンド/メニュー | 説明                                                                                             |
|-----------|-----------|------------------------------------------------------------------------------------------------|
| disk      | コマンド      | システムのドライブをすべて表示する。後の操作で使用するディスクを選択することもできる。このディスクは、「現在のディスク」と呼ばれる。                             |
| type      | コマンド      | 現在のディスクのメーカーとモデルを表示する。認識されているドライブタイプのリストも表示する。SCSI-2 対応ディスクドライブの場合は Auto configure オプションを選択する。 |
| partition | メニュー      | スライスの作成および変更を行う。詳細は、234 ページの「partition メニュー」を参照。                                               |



表 14-1 format のメインメニュー項目の説明 (続き)

| 項目      | コマンド/メニュー | 説明                                                                                                                                                                                                                                                           |
|---------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| current | コマンド      | 現在のディスクに関する次の情報を表示する。<br><ul style="list-style-type: none"> <li>■ デバイス名とデバイスタイプ</li> <li>■ シリンダ数、代替シリンダ数、ヘッド数、セクター数</li> <li>■ 物理デバイス名</li> </ul>                                                                                                            |
| 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。<br><ol style="list-style-type: none"> <li>1. format.dat ファイル内の情報</li> <li>2. 自動構成プロセスからの情報</li> <li>3. format.dat エントリが見つからない場合、プロンプトに入力した情報</li> </ol> <p>このコマンドは、IDE ディスクには適用できない。IDE ディスクは、あらかじめ製造元でフォーマットされる。</p> |
| fdisk   | メニュー      | x86 プラットフォームのみ: fdisk プログラムを実行し、Solaris fdisk パーティションを作成する。                                                                                                                                                                                                  |
| repair  | コマンド      | 現在のディスク上で特定のブロックを修復する。                                                                                                                                                                                                                                       |
| label   | コマンド      | 現在のディスクに新しいラベルを書き込む。                                                                                                                                                                                                                                         |
| analyze | メニュー      | 読み取り、書き込み、比較テストを実行する。詳細は、236 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、237 ページの「defect メニュー」を参照。この機能は、IDE ディスクには適用できない。IDE ディスクは、自動欠陥管理を行う。                                                                                                                                                                      |
| backup  | コマンド      | <b>VTOC</b> - バックアップラベルを検索<br><b>EFI</b> - サポートされない                                                                                                                                                                                                          |
| verify  | コマンド      | 現在のディスクに関する次の情報を出力する。<br><ul style="list-style-type: none"> <li>■ デバイス名とデバイスタイプ</li> <li>■ シリンダ数、代替シリンダ数、ヘッド数、セクター数</li> <li>■ パーティションテーブル</li> </ul>                                                                                                        |
| save    | コマンド      | <b>VTOC</b> - 新しいディスク情報およびパーティション情報を保存<br><b>EFI</b> - 適用できない                                                                                                                                                                                                |
| inquiry | コマンド      | 現在のドライブのベンダ、製品名、リビジョンレベルが出力される (SCSI ディスクのみ)。                                                                                                                                                                                                                |
| volname | コマンド      | 新しい 8 文字のボリューム名を使用してディスクラベルを作成する。                                                                                                                                                                                                                            |

表 14-1 format のメインメニュー項目の説明 (続き)

| 項目   | コマンド/メニュー | 説明                |
|------|-----------|-------------------|
| quit | コマンド      | format メニューを終了する。 |

## partition メニュー

partition メニューは次のようになっています。

```
format> partition
PARTITION MENU:
  0 - change '0' partition
  1 - change '1' partition
  2 - change '2' partition
  3 - change '3' partition
  4 - change '4' partition
  5 - change '5' partition
  6 - change '6' partition
  7 - change '7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit
partition>
```

次の表に、partition メニューの項目を示します。

表 14-2 partition メニュー項目の説明

| サブコマンド               | 説明                                                                                                                                  |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| change 'n' partition | 新しいスライスに次の情報を設定する。<br><ul style="list-style-type: none"> <li>■ 識別タグ</li> <li>■ アクセス権フラグ</li> <li>■ 開始シリンダ</li> <li>■ サイズ</li> </ul> |
| select               | 事前定義されたスライステーブルを選択する。                                                                                                               |
| modify               | スライステーブル内のすべてのスライスを変更可能にする。<br>個々のスライスに対して change 'x' partition コマンドを実行するよりも、このコマンドが使用されることが多い。                                     |
| name                 | 現在のスライステーブルの名前を指定する。                                                                                                                |
| print                | 現在のスライステーブルを表示する。                                                                                                                   |
| label                | スライスマップとラベルを現在のディスクに書き込む。                                                                                                           |

表 14-2 partition メニュー項目の説明 (続き)

| サブコマンド | 説明                   |
|--------|----------------------|
| quit   | partition メニューを終了する。 |

## x86: fdisk メニュー

x86 システム上でのみ、次のような fdisk メニューが表示されます。

```
format> fdisk
          Total disk size is 1855 cylinders
          Cylinder size is 553 (512 byte) blocks
                                Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====
          1      DOS-BIG   0      370   371     20
          2      Active   SOLARIS 370   1851  1482     80

SELECT ONE OF THE FOLLOWING:
1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)

Enter Selection:
```

次の表に、fdisk メニューの項目を示します。

表 14-3 x86: fdisk メニュー項目の説明

| メニュー項目                  | 説明                                                                                                                                                    |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a partition      | fdisk パーティションを作成する。Solaris や DOS など、オペレーティングシステムごとに別々のパーティションを作成しなければならない。1 台のディスクの最大パーティション数は 4 である。fdisk のパーティションのサイズをパーセンテージで入力するように促すプロンプトが表示される。 |
| Change Active partition | ブートに使用するパーティションを指定する。このメニュー項目により、第 1 段階のブートプログラムが第 2 段階のブートプログラムを検索する場所を指定する。                                                                         |
| Delete a partition      | 以前に作成したパーティションを削除する。このコマンドを実行すると、パーティション内のすべてのデータが失われる。                                                                                               |
| Exit                    | 新しいパーティションテーブルを書き込んで fdisk メニューを終了する。                                                                                                                 |
| Cancel                  | パーティションテーブルを変更せずに fdisk メニューを終了する。                                                                                                                    |

## analyze メニュー

analyze メニューは次のようになっています。

```
format> analyze
```

```
ANALYZE MENU:
```

```
read      - read only test      (doesn't harm SunOS)
refresh   - read then write     (doesn't harm data)
test      - pattern testing     (doesn't harm data)
write     - write then read     (corrupts data)
compare   - write, read, compare (corrupts data)
purge     - write, read, write  (corrupts data)
verify    - write entire disk, then verify (corrupts data)
print     - display data buffer
setup     - set analysis parameters
config    - show analysis parameters
quit
```

```
analyze>
```

次の表に、analyze メニューの項目を示します。

表 14-4 analyze メニュー項目の説明

| サブコマンド  | 説明                                                                                                                                                                     |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| read    | 現在のディスクの各セクターを読み込む。デフォルトで欠陥ブロックを修復する。                                                                                                                                  |
| refresh | データを損なわずに、現在のディスク上で読み取りおよび書き込みを実行する。デフォルトで欠陥ブロックを修復する。                                                                                                                 |
| test    | データを損なわずに一連のパターンをディスクに書き込む。デフォルトで欠陥ブロックを修復する。                                                                                                                          |
| write   | 一連のパターンをディスクに書き込んでから、そのデータをディスクから読み込む。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。                                                                                            |
| compare | ディスクに一連のパターンを書き込み、そのデータを読み込み、書き込みバッファ内のデータと比較する。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。                                                                                  |
| purge   | ディスク上のデータをすべて削除し、いかなる手段でも取り出せないようにする。ディスク全体 (またはディスクのセクション) に 3 種類のパターンを書き込むことにより、データを削除する。検査に合格すると 16 進のビットパターンがディスク全体 (またはディスクのセクション) に上書きされる。<br>デフォルトで欠陥ブロックを修復する。 |
| verify  | 1 度目にディスク全体の各ブロックに固有のデータを書き込む。2 度目にそのデータを読み込んで検査する。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。                                                                               |

表 14-4 analyze メニュー項目の説明 (続き)

| サブコマンド | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| print  | 読み取り / 書き込みバッファ内のデータを表示する。                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| setup  | 次の解析パラメータを指定する。<br><br>Analyze entire disk? yes<br>Starting block number: ドライブによって異なる<br>Ending block number: ドライブによって異なる<br>Loop continuously? no<br>Number of passes: 2<br>Repair defective blocks? yes<br>Stop after first error? no<br>Use random bit patterns? no<br>Number of blocks per transfer: 126 (0/n/mn)<br>Verify media after formatting? yes<br>Enable extended messages? no<br>Restore defect list? yes<br>Restore disk label? yes<br><br>太字はデフォルトを示す。 |
| config | 現在の解析パラメータを表示する。                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| quit   | analyze メニューを終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

## defect メニュー

defect メニューは次のようになっています。

```
format> defect

DEFECT MENU:
  primary - extract manufacturer's defect list
  grown   - extract manufacturer's and repaired defects lists
  both    - extract both primary and grown defects lists
  print   - display working list
  dump    - dump working list to file
  quit
```

次の表に、defect メニューの項目を示します。

表 14-5 defect メニュー項目の説明

| サブコマンド  | 説明                                          |
|---------|---------------------------------------------|
| primary | メーカーの欠陥リストをディスクドライブから読み込み、メモリー内の欠陥リストを更新する。 |

表 14-5 defect メニュー項目の説明 (続き)

| サブコマンド | 説明                                             |
|--------|------------------------------------------------|
| grown  | 増分の欠陥リスト(分析により検出された欠陥)を読み込んで、メモリー内の欠陥リストを更新する。 |
| both   | メーカーの欠陥リストと増分の欠陥リストを読み込んで、メモリー内の欠陥リストを更新する。    |
| print  | メモリー内の欠陥リストを表示する。                              |
| dump   | メモリー内の欠陥リストをファイルに保存する。                         |
| quit   | defect メニューを終了する。                              |

## format.dat ファイル

Solaris オペレーティングシステムと一緒に出荷される format.dat ファイルでは、多数の標準的なディスクがサポートされます。使用中のディスクドライブが format.dat ファイルに含まれていない場合は、ファイルにエントリを追加するか、format ユーティリティで処理を実行中に type コマンドを選択してから other オプションを選択してエントリを追加できます。

ディスクドライブをサイト全体で使用する場合は、format.dat ファイルにエン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。

次のいずれかの場合には、システムの /etc/format.dat ファイルを変更する必要があります。

- ディスクが Solaris オペレーティングシステム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 オペレーティングシステムのデフォルト構成とは異なる場合

注 - /etc/format.dat ファイルのデフォルトエン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する場合は、混乱を避けるために、そのエントリをコピーし、別の名前を付けてから変更します。

/etc/format.dat は、EFI ラベル付きディスクには適用できません。

## format.dat ファイルの内容

format.dat データファイルには、format ユーティリティに使用されるディスクドライブ情報が入っています。format.dat ファイル内では、次の3つの項目が定義されています。

- 検索パス
- ディスクタイプ
- スライステーブル

## format.dat ファイルの構文

/etc/format.dat ファイルには、次の構文規則が適用されます。

- ポンド記号 (#) はコメント文字です。ポンド記号に続く1行のテキストは、format ユーティリティでは解釈されません。
- format.dat ファイル内の各定義は、1つの論理行で評価されます。定義が長すぎて1行に収まらない場合は、定義の最終行を除くすべての行末にバックスラッシュ (\) を付けなければなりません。
- 定義は、左辺に識別子、右辺に1つまたは複数の値を持つ一連の代入式からなっています。代入演算子は等号 (=) です。定義内の代入式はコロン (:) で区切らなければなりません。
- format ユーティリティは、空白を無視します。代入値に空白を含める場合は、値全体を二重引用符 (") で囲みます。この構文により、引用符の内側の空白は代入値の一部として保持されます。
- 代入式によっては、右辺に複数の値を指定できるものがあります。値はコンマ (,) で区切ります。

## format.dat ファイル中のキーワード

format.dat ファイルには、format ユーティリティが起動時に読み込むディスク定義が含まれます。各定義の先頭には、キーワード `disk_type` または `partition` が付きます。これらのキーワードについて、次の表で説明します。

表 14-6 format.dat ファイルのキーワードの説明

| キーワード     | 用途                                                                                                                                                                                                                                        |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| disk_type | <p>コントローラとディスクのモデルを定義する。各 disk_type 定義には、ディスクの物理ジオメトリに関する情報が入っている。デフォルトのデータファイルには、Solaris オペレーティングシステムでサポートされるコントローラとディスクの定義が入っている。</p> <p>サポートされないディスクを使用する場合にのみ、新しい disk_type を追加する必要がある。必要に応じて、disk_type 定義をデータファイルにいくつ追加してもかまわない。</p> |
| partition | <p>特定のディスクタイプのスライステーブルを定義する。スライステーブルには、スライス情報だけでなく、format ユーティリティ内で参照可能な名前が入っている。デフォルトの format.dat ファイルには、数種類のディスクドライブに対応するデフォルトのスライス定義が含まれる。システムのディスク上にスライスを作成し直した場合は、スライス定義を追加する。必要に応じて、スライス情報をデータファイルにいくつ追加してもかまわない。</p>               |

## ディスクタイプ (format.dat)

format.dat ファイル内の disk\_type キーワードは、コントローラとディスクのモデルを定義します。各 disk\_type 定義には、ディスクの物理ジオメトリに関する情報が含まれます。デフォルトの format.dat ファイルには、Solaris オペレーティングシステムでサポートされるコントローラとディスクの定義が入っています。サポートされないディスクを使用する場合にのみ、新しい disk\_type を追加する必要があります。必要に応じて、disk\_type 定義をデータファイルにいくつ追加してもかまいません。

キーワード自体が、ディスクタイプ名になります。この名前は、ディスクのラベルの一部になり、format ユーティリティの実行時にディスクタイプを識別するために使用されます。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべての disk\_type 定義でキーワードのほかに割り当てなければならない識別子を示します。

表 14-7 必須の disk\_type 識別子

| 識別子   | 説明                                                                                                      |
|-------|---------------------------------------------------------------------------------------------------------|
| ctrlr | ディスクタイプで有効なコントローラのタイプ。現在、有効な値は SCSI と ATA である。                                                          |
| ncyl  | ディスクタイプ内のデータシリンダ数。この数によって、システムがアクセスできるディスクの論理シリンダ数が決まる。                                                 |
| acyl  | ディスクタイプ内の代替シリンダ数。format ユーティリティは、これらのシリンダを使用して、ドライブの欠陥リストなどの情報を格納する。代替シリンダとして、常に 2 つ以上のシリンダを残しておく必要がある。 |



表 14-7 必須の disk\_type 識別子 (続き)

| 識別子   | 説明                                                                                                      |
|-------|---------------------------------------------------------------------------------------------------------|
| pcyl  | ディスクタイプ内の物理シリンダ数。この数値は、ディスクメディアの境界を計算するために使用される。通常、この数値は ncyl と acyl の合計に等しくなる。                         |
| nhead | ディスクタイプ内のヘッド数。この数値は、ディスクメディアの境界を計算するために使用される。                                                           |
| nsect | ディスクタイプ内の 1トラック当たりのデータセクター数。この数値は、ディスクメディアの境界を計算するために使用される。これはデータセクターだけである。スペアは、各トラックのデータセクション数には含まれない。 |
| rpm   | ディスクタイプの 1分当たりの回転数。この情報はラベルに書き込まれ、後からファイルシステムでファイルデータの最適位置の計算に使用される。                                    |

コントローラによっては、他の識別子が必要な場合があります。次の表に、SCSI コントローラに必要な識別子を示します。

表 14-8 SCSI コントローラの disk\_type 識別子

| 識別子       | 説明                                                               |
|-----------|------------------------------------------------------------------|
| fmt_time  | 所定のドライブのフォーマットに要する時間を示す数値。詳細は、コントローラのマニュアルを参照。                   |
| cache     | format ユーティリティの処理中にオンボードキャッシュの動作を制御する数値。詳細は、コントローラのマニュアルを参照。     |
| trks_zone | 代替セクターのマッピング内で使用される 1つの欠陥領域当たりのトラック数を指定した数値。詳細は、コントローラのマニュアルを参照。 |
| asect     | 所定の欠陥領域内で代替マッピングに利用可能なセクター数を指定する。詳細は、コントローラのマニュアルを参照。            |

次に、disk\_type 定義の例を示します。

```
disk_type = "SUN1.3G" \
: ctrlr = 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" \
: ctrlr = SCSI : fmt_time = 4 \
: ncyl = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \
: rpm = 5400 : bpt = 44823
```

```
disk_type = "SUN2.9G" \
: ctrlr = 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)

format.dat ファイル内のパーティションテーブルに、特定のディスクタイプのスライステーブルが定義されています。

format.dat ファイル内の partition キーワードが、スライステーブル名になります。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべてのスライステーブル内で値を代入しなければならない識別子を示します。

表 14-9 スライステーブルの必須識別子

| 識別子   | 説明                                                                                                                                            |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| disk  | このスライステーブルが定義されている disk_type の名前。この名前は disk_type 内で使用されるとおりに指定しなければならない。                                                                      |
| 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、有効な値は ATA コントローラを表す ATA と SCSI コントローラを表す SCSI である。ここで指定したコントローラタイプは、disk_type 定義で選択した disk_type にも定義する必要がある。 |

スライス定義内の他の識別子では、実際のスライス情報を記述します。識別子は 0 から 7 までの番号です。これらの識別子は省略可能です。明示的に割り当てられていないスライスは、長さ 0 に設定されます。これらの識別子の値は、それぞれコマンドで区切られた数値のペアになります。最初の数値はスライスの開始シリンダで、第 2 はスライス内のセクター数です。次に、スライス定義の例を示します。

```
partition = "SUN1.3G" \  
: disk = "SUN1.3G" : 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 ユーティリティの代替データファイルを指定する

format ユーティリティは、次の方法で代替ファイルの位置を認識します。

1. `format -x` オプションでファイル名を指定した場合、ファイルは常にデータファイルとして使用されます。
2. `-x` オプションを指定しない場合、`format` ユーティリティは現在のディレクトリ内でファイル `format.dat` を検索します。このファイルが見つかったら、データファイルとして使用されます。
3. どちらの方法でもデータファイルが見つからない場合、`format` ユーティリティはデータファイルとして `/etc/format.dat` を使用します。このファイルは Solaris オペレーティングシステムと共に出荷されるので、必ず存在します。

---

## format コマンドへの入力規則

`format` ユーティリティを使用する場合は、さまざまな情報を入力する必要があります。この節では、入力する情報に関する規則について説明します。データ入力時に `format` のヘルプ機能を使用する方法については、[245 ページの「format ユーティリティのヘルプを利用する」](#)を参照してください。

### format コマンドへ番号を指定する

`format` ユーティリティを使用する際、いくつかの数値を入力する必要があります。入力方法には、データを指定する方法と、選択肢のリストから番号を選択する方法があります。どちらの場合も、`help` 機能を使用すると、`format` は期待する数値の上限と下限を表示します。したがって、目的の数値を入力するだけで済みます。数値は、その一部として底を明示的に指定しない限り (16 進数を表す `0x` など)、10 進数と見なされます。

次の例は、整数の入力を示しています。

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

### format コマンドへブロック番号を指定する

ディスクのブロック番号を入力しなければならない場合は、情報を次の 2 つの方法で入力できます。

- ブロック番号を整数として入力する。
- ブロック番号をシリンダ / ヘッド / セクター書式で入力する。

この情報は、論理ブロック番号を表す整数として指定できます。任意の底の数値を指定できますが、デフォルトは 10 進です。また、ここで最大演算子 (ドル記号 `$`) を使用して、`format` ユーティリティに適切な値を選択させることもできます。論理ブロックの形式は、SunOS のディスクドライバによってエラーメッセージに使用されません。

ブロック番号を指定するには、シリンダ / ヘッド / セクター書式を使用する方法もあります。この形式では、ブロック番号の3つの論理構成要素である、シリンダ、ヘッド、およびセクターの値を明示的に指定しなければなりません。これらの値は論理値ですが、メディアのレイアウトに関連するディスク領域の定義に使用できます。

シリンダ / ヘッド / セクター番号を指定しない場合、値は0であると見なされます。また、番号の代わりに最大演算子を使用して、format ユーティリティに適切な値を選択させることもできます。次に、シリンダ、ヘッド、セクターエントリの例を示します。

```
Enter defective block number: 34/2/3
Enter defective block number: 23/1/
Enter defective block number: 457//
Enter defective block number: 12345
Enter defective block number: Oxabcd
Enter defective block number: 334/$/2
Enter defective block number: 892//$
```

format は、ブロック番号を常に上記の両方の書式で出力します。また、help 機能によって、期待されるブロック番号の上限と下限が両方の書式で表示されます。

## format のコマンド名を指定する

format ユーティリティでメニュープロンプトが表示される場合は、コマンド名を入力する必要があります。コマンド名は、目的のコマンドとして区別できる長さまで「省略」できます。

たとえば、p(partition) を使用して format メニューから partition メニューにアクセスできます。次に、p(rint) を使用して現在のスライステーブルを表示できます。

```
format> p
PARTITION MENU:
0      - change '0' partition
1      - change '1' partition
2      - change '2' partition
3      - change '3' partition
4      - change '4' partition
5      - change '5' partition
6      - change '6' partition
7      - change '7' partition
select - select a predefined table
modify  - modify a predefined partition table
name    - name the current table
print   - display the current table
label   - write partition map and label to the disk
quit
partition> p
```

## format コマンドへディスク名を指定する

format ユーティリティでは、名前を指定しなければならない場合があります。このような場合は、名前に使用したい文字列を自由に指定できます。空白を含む名前は、二重引用符 (") で囲まなければならない。二重引用符で囲まなければ、名前の最初の語だけが使用されます。

たとえば、ディスクの特定のパーティションテーブルを指定する場合、partition メニューの name サブコマンドを使用できます。

```
partition> name
Enter table name (remember quotes): "new disk3"
```

---

## format ユーティリティのヘルプを利用する

format ユーティリティにはヘルプ機能が組み込まれており、format ユーティリティが入力待ちの状態であればいつでも使用できます。疑問符 (?) を入力するだけで必要な入力に関するヘルプが表示されます。format ユーティリティでは、どんなタイプの入力が必要かについて簡潔な説明が表示されます。

メニュープロンプトから ? と入力すると、利用できるコマンドのリストが表示されます。

format ユーティリティに関連するマニュアルページには、次が含まれます。

- format (1M) - format ユーティリティの基本機能およびコマンド行で使用可能なすべての変数について説明する。
- format.dat (4) - format ユーティリティで使用するディスクドライブ構成に関する情報を提供する。



## 第 15 章

---

# ファイルシステムの管理 (概要)

---

ファイルシステムの管理は、もっとも重要なシステム管理作業の 1 つです。

この章の内容は次のとおりです。

- 247 ページの「Solaris 9 Update リリースの新しいファイルシステム機能」
- 255 ページの「ファイルシステム管理作業についての参照先」
- 255 ページの「ファイルシステムの概要」
- 256 ページの「ファイルシステムのタイプ」
- 262 ページの「ファイルシステム管理用のコマンド」
- 264 ページの「デフォルトの Solaris ファイルシステム」
- 265 ページの「スワップ空間」
- 265 ページの「UFS ファイルシステム」
- 268 ページの「ファイルシステムのマウントとマウント解除」
- 273 ページの「ファイルシステムのタイプを調べる」

---

## Solaris 9 Update リリースの新しい ファイルシステム機能

この節では、この Solaris リリースの新しいファイルシステム機能について説明します。

### デフォルトで有効な UFS ロギング

**Solaris 9 9/04** – ロギングは、すべての UFS ファイルシステムでデフォルトで有効になります。ただし、次の場合は除きます。

- ロギングが明示的に無効とされた場合

- ログ用のファイルシステム容量が不足している場合

以前の Solaris リリースでは、UFS ロギングは手動で有効にする必要がありました。UFS ロギングの詳細については、267 ページの「UFS ロギング」を参照してください。

このリリースで UFS ロギングを使用する際には、次の点に注意してください。

- ユーザー、アプリケーション、UFS ロギングなどの一般的なシステム要求に対応できるだけのディスク容量が確保されているかどうかを確認してください。
- ディスク容量が不足していてデータをロギングできない場合は、次のようなメッセージが表示されます。

```
# mount /dev/dsk/c0t4d0s0 /mnt
/mnt: No space left on device
Could not enable logging for /mnt on /dev/dsk/c0t4d0s0.
#
```

ただし、その場合でも、ファイルシステムはマウントされます。次に例を示します。

```
# df -h /mnt
Filesystem                size      used  avail capacity  Mounted on
/dev/dsk/c0t4d0s0         142M      142M      0K    100%      /mnt
#
```

- ロギングが有効にされた UFS ファイルシステムでは、ほとんど空の状態であっても、いくらかのディスク容量がログ用として消費されます。
- 以前の Solaris リリースからこの Solaris リリースにアップグレードする場合、`/etc/vfstab` ファイル内で `logging` オプションが指定されていなかった場合でも、UFS ファイルシステムのロギングは有効になります。ロギングを無効にするには、`/etc/vfstab` ファイル内の UFS ファイルシステムのエントリに、`nologging` オプションを追加します。

## デフォルトのロギングと標準への準拠

ファイルからブロックを解放する UFS ファイルシステムトランザクションが実行されても、それらの解放されたブロックがファイルシステムの解放リストに即座に追加されない可能性があります。この動作は、ロギングを有効にしてマウントされた UFS ファイルシステムシステム上で発生します。

このような動作は、ファイルシステムのパフォーマンスの向上につながりますが、次の標準に準拠した動作ではありません。

- POSIX, Single UNIX<sup>®</sup> Specification
- SPARC<sup>®</sup> Conformance Definition
- SPARC Conformance Definition System V Application Binary Interface
- System V Interface Definition
- X/Open<sup>®</sup> Portability Guide

これらの標準に準拠するには、解放された領域が即座に利用可能になる必要があります。



次のような場合は、UFS ロギングを無効にすることを検討してください。

- ファイルの削除に関して標準に準拠させたい場合
- 満杯状態に近いファイルシステム上で、ファイルを削除した直後にファイルを作成または拡張しようとする問題が発生する場合

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

## SPARC: マルチテラバイトの UFS ファイルシステムのサポート

**Solaris 9 8/03** – この Solaris リリースでは、64 ビットの Solaris カーネルを実行しているシステムで、マルチテラバイトの UFS ファイルシステムをサポートします。

以前は、64 ビットと 32 ビットのどちらのシステムでも、UFS ファイルシステムはおよそ 1T バイトに制限されていました。UFS ファイルシステムのコマンドとユーティリティはすべて、マルチテラバイトの UFS ファイルシステムに対応するように更新されました。

たとえば、`ufsdump` コマンドは大規模な UFS ファイルシステムをダンプできるようにブロックサイズが大きくなりました。

```
# ufsdump 0f /dev/md/rdisk/d97 /dev/md/rdisk/d98
DUMP: Date of this level 0 dump: Tue Jan 07 14:23:36 2003
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/md/rdisk/d98 to /dev/md/rdisk/d97.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Forcing larger tape block size (2048).
DUMP: Writing 32 Kilobyte records
DUMP: Estimated 4390629500 blocks (2143862.06MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
```

1T バイト未満の UFS ファイルシステムは、以前と同じように管理されます。1T バイト未満の UFS ファイルシステムと 1T バイトを超えるファイルシステムとの間に管理面での違いはありません。

最初に `newfs -T` オプションを使用していれば、1T バイト未満の UFS ファイルシステムを作成し、最終的にマルチテラバイトのファイルシステムに拡張されるように指定できます。このオプションを使用すると、マルチテラバイトのファイルシステムに適した値に拡張が可能であるように `i` ノードとフラグメントの密度が設定されます。

32 ビットのカーネルを実行しているシステムで 1T バイト未満の UFS ファイルシステムを作成するときに、`newfs -T` オプションを使用すると、最終的に 64 ビットのカーネルでこのシステムを起動するときに、`growfs` コマンドを使ってこのファイルシステムを拡張できます。詳細については、`newfs(1M)` のマニュアルページを参照してください。

growfs コマンドを使用すると、サービスやデータを失わずに、UFS ファイルシステムをスライスまたはボリュームのサイズまで拡張できます。詳細については、growfs (1M) のマニュアルページを参照してください。

これに関連して、EFI ディスクラベルによるマルチテラバイトボリュームのサポートと、Solaris ボリュームマネージャによるマルチテラバイトボリュームのサポートという新しい2つの機能が追加されました。詳細については、163 ページの「SPARC: EFI ディスクラベルによるマルチテラバイトディスクのサポート」および『Solaris ボリュームマネージャの管理』を参照してください。

## マルチテラバイトの UFS ファイルシステムの機能

マルチテラバイトの UFS ファイルシステムには、次の機能があります。

- 最大 16T バイトの UFS ファイルシステムを作成できる
- 16T バイト未満のファイルシステムを作成し、後で 16T バイトまでサイズを増やすことができる
- マルチテラバイトのファイルシステムを物理ディスク、Solaris ボリュームマネージャの論理ボリューム、および Veritas の VxVM 論理ボリューム上に作成できる
- マルチテラバイトのファイルシステムでは、UFS ログギングが有効になっているとパフォーマンスが向上するというメリットがある。また、ログギングが有効なときは fsck コマンドを実行しなくてもよい場合があるというログギングのメリットもある
- マルチテラバイトの UFS ファイルシステム用のパーティションを作成すると、そのディスクには自動的に EFI ディスクラベルが付く。EFI ディスクラベルの詳細については、163 ページの「SPARC: EFI ディスクラベルによるマルチテラバイトディスクのサポート」を参照

## マルチテラバイトの UFS ファイルシステムの制限事項

マルチテラバイトの UFS ファイルシステムには、次の制限事項があります。

- Solaris x86 システムでは、この機能はサポートされていない
- 32 ビットの Solaris カーネルを実行しているシステムでは、1T バイトを超えるファイルシステムをマウントすることはできない
- 64 ビットの Solaris カーネルを実行しているシステムでは、1T バイトを超えるファイルシステムからブートすることはできない。つまり、マルチテラバイトのファイルシステムにルート (/) ファイルシステムを配置することはできない
- 1T バイトを超える個々のファイルはサポートされていない
- UFS ファイルシステムの 1T バイトあたりの最大ファイル数は 100 万である。たとえば、4T バイトのファイルシステムには、400 万個のファイルを含めることができる

100 万に制限されているのは、fsck コマンドでファイルシステムを検査するのに要する時間を減らすためである。

- マルチテラバイトの UFS ファイルシステムに設定できる最大割り当て数は、2T バイトの 1024 バイトブロックである
- 現時点では、`fssnap` コマンドを使ってマルチテラバイトの UFS ファイルシステムのスナップショットを作成することはできない

## ▼ マルチテラバイトの UFS ファイルシステムを作成する方法

マルチテラバイトの UFS ファイルシステムは、Solaris ボリュームマネージャまたは VxVM ボリューム、あるいは 1T バイトを超える物理ディスクとして提供されたマルチテラバイトの LUN が利用できることを前提にサポートされています。

マルチテラバイトの UFS ファイルシステムを作成する前に、次のいずれかを行う必要があります。

- `format` ユーティリティまたは Solaris インストールユーティリティを使ってマルチテラバイトのディスクパーティションを作成しておく
- Solaris ボリュームマネージャを使ってマルチテラバイトのボリュームを設定しておく

手順 1. スーパーユーザーになります。

2. 論理ボリューム上にマルチテラバイトの **UFS** ファイルシステムを作成します。

たとえば、次のコマンドを実行すると、1.8T バイトボリュームの UFS ファイルシステムが作成されます。

```
# newfs /dev/md/rdsk/d99
newfs: construct a new file system /dev/md/rdsk/d99: (y/n)? y
/dev/md/rdsk/d99: 3859402752 sectors in 628158 cylinders of 48 tracks,
128 sectors
1884474.0MB in 4393 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
3850872736, 3851751456, 3852630176, 3853508896, 3854387616, 3855266336,
3856145056, 3857023776, 3857902496, 3858781216,
#
```

3. 新しく作成したファイルシステムの整合性を検査します。

次に例を示します。

```
# fsck /dev/md/rdsk/d99
** /dev/md/rdsk/d99
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
```

```

** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 2 used, 241173122 free (0 frags, 241173122 blocks, 0.0%
fragmentation)
#

```

4. 新しく作成したファイルシステムをマウントして検査します。  
次に例を示します。

```

# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem                size      used  avail capacity  Mounted on
/dev/md/dsk/d99           1.8T        64M   1.8T      1%      /bigdir

```

## ▼ マルチテラバイトの UFS ファイルシステムを拡張する方法

マルチテラバイトの UFS ファイルシステムを作成した後は、`growfs` コマンドを使ってファイルシステムを拡張できます。たとえば、前の手順でボリュームに作成したファイルシステムを使用すると、別のディスクをこのボリュームに追加できます。その後で、ファイルシステムを拡張します。

- 手順
1. スーパーユーザーになります。
  2. 別のディスクをボリュームに追加します。  
次に例を示します。

```

# metattach d99 c4t5d0s4
d99: component is attached
# metastat
d99: Concat/Stripe
      Size: 5145882624 blocks (2.4 TB)
      Stripe 0:
          Device      Start Block  Dbase  Reloc
          c0t1d0s4      36864      Yes   Yes
      Stripe 1:
          Device      Start Block  Dbase  Reloc
          c3t7d0s4         0       No   Yes
      Stripe 2:
          Device      Start Block  Dbase  Reloc
          c1t1d0s4         0       No   Yes
      Stripe 3:
          Device      Start Block  Dbase  Reloc
          c4t5d0s4         0       No   Yes

```

3. ファイルシステムを拡張します。  
次に例を示します。

```

# growfs -v /dev/md/rdsk/d99
/usr/lib/fs/ufs/mkfs -G /dev/md/rdsk/d99 5145882624
/dev/md/rdsk/d99: 5145882624 sectors in 837546 cylinders of 48 tracks,

```

```

128 sectors
      2512638.0MB in 5857 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
 5137130400, 5138009120, 5138887840, 5139766560, 5140645280, 5141524000,
 5142402720, 5143281440, 5144160160, 5145038880,
#

```

4. 拡張したファイルシステムをマウントして検査します。  
次に例を示します。

```

# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem              size  used  avail capacity  Mounted on
/dev/md/dsk/d99         2.4T   64M   2.4T    1%      /bigdir

```

## ▼ UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法

UFS ファイルシステムを 1T バイトを超えるサイズに拡張するには、次の手順を実行します。この手順は、`newfs -T` オプションを使って UFS ファイルシステムを作成したことを前提としています。

- 手順
1. スーパーユーザーになります。
  2. 現在のディスクまたはボリュームのサイズを特定します。  
たとえば、次のボリュームは 800G バイトです。

```

# metastat d98
d98: Concat/Stripe
  Size: 1677754368 blocks (800 GB)
  Stripe 0:
    Device      Start Block  Dbase  Reloc
    c0t1d0s4      0         No     Yes
  Stripe 1:
    Device      Start Block  Dbase  Reloc
    c3t7d0s4      0         No     Yes

```

3. このボリュームを **1T** バイトよりも大きくします。  
次に例を示します。

```

# metattach d98 c1t1d0s4
d98: component is attached
# metastat d98
d98: Concat/Stripe
  Size: 2516631552 blocks (1.2 TB)
  Stripe 0:

```

| Device    | Start Block | Dbase | Reloc |
|-----------|-------------|-------|-------|
| c0t1d0s4  | 0           | No    | Yes   |
| Stripe 1: |             |       |       |
| Device    | Start Block | Dbase | Reloc |
| c3t7d0s4  | 0           | No    | Yes   |
| Stripe 2: |             |       |       |
| Device    | Start Block | Dbase | Reloc |
| c1t1d0s4  | 0           | No    | Yes   |

4. ディスクまたはボリュームの **UFS** ファイルシステムを **1T** バイトよりも大きくします。

次に例を示します。

```
growfs -v /dev/md/rdisk/d98
/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d98 2516631552
/dev/md/rdisk/d98: 2516631552 sectors in 68268 cylinders of 144 tracks,
256 sectors
1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 921888, 1843744, 2765600, 3687456, 4609312, 5531168, 6453024, 737...
8296736,
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
2507714848, 2508636704, 2509558560, 2510480416, 2511402272, 2512324128,
2513245984, 2514167840, 2515089696, 2516011552,
```

5. 拡張したファイルシステムをマウントして検査します。

次に例を示します。

```
# mount /dev/md/dsk/d98 /datadir
# df -h /datadir
Filesystem          size  used  avail capacity  Mounted on
/dev/md/dsk/d98    1.2T   64M   1.2T     1%    /datadir
```

## マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡

マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡には、次のエラーメッセージと解決法を参考にしてください。

エラーメッセージ (次に例を示す)

```
mount: /dev/rdisk/c0t0d0s0 はこのファイルシステム形式ではありません。
```

原因

Solaris 9 8/03 リリースよりも以前のリリースが動作するシステムに 1T バイトを超える UFS ファイルシステムをマウントしようとしました。

解決法

1T バイトを超える UFS ファイルシステムは、Solaris 9 8/03 以降のリリースが動作するシステムにマウントしてください。

### エラーメッセージ

"ファイルシステムがマルチテラバイト形式に設定されていません。" "そのサイズは 1 テラバイト以上にできません。"

### 原因

`newfs -T` コマンドを使わずに作成したファイルシステムを拡張しようとした。

### 解決法

- 1T バイトより大きくするファイルシステムのデータをバックアップします。
- `newfs` コマンドを使用して、ファイルシステムをマルチテラバイトファイルシステムに再作成します。
- 新しく作成したファイルシステムにバックアップされていたデータを復元します。

---

## ファイルシステム管理作業についての参照先

ファイルシステムの管理に関する手順を調べるには、次の参照先を使用します。

| ファイルシステム管理作業                                  | 参照先           |
|-----------------------------------------------|---------------|
| 新しいファイルシステムを作成する                              | 第 16 章と第 18 章 |
| ローカルファイルとリモートファイルをユーザーが利用できるようにする             | 第 17 章        |
| 新しいディスクデバイスを接続して構成する                          | 第 10 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 22 章        |
| ファイルシステムの矛盾をチェックして訂正する                        | 第 20 章        |

---

## ファイルシステムの概要

ファイルシステムは、ファイルを編成して格納するためのディレクトリ構造です。「ファイルシステム」という用語は、次のような場合に使用されます。

- ディスクベース、ネットワークベース、または仮想ファイルシステムなど、特定のタイプのファイルシステムを指す場合
- ルートディレクトリから始まるファイルツリー全体を指す場合
- ディスクスライスや他の記憶メディアデバイスのデータ構造を指す場合
- ファイルツリー構造のうち、ファイルがアクセスできるように主なファイルツリー上のマウントポイントに接続されている部分を指す場合

通常、その意味は状況に応じて判断できます。

Solaris オペレーティング環境は、各種ファイルシステムへの標準インタフェースを提供する「仮想ファイルシステム」(VFS) アーキテクチャを使用します。VFS アーキテクチャを使用すると、カーネルはファイルの読み取り、書き込み、一覧表示などの基本操作を処理でき、新しいファイルシステムの追加が簡単になります。

---

## ファイルシステムのタイプ

Solaris システムソフトウェアでは、次の 3 種類のファイルシステムがサポートされます。

- ディスクベースのファイルシステム
- ネットワークベースのファイルシステム
- 仮想ファイルシステム

ファイルシステムのタイプを確認するには、[273 ページの「ファイルシステムのタイプを調べる」](#)を参照してください。

## ディスクベースのファイルシステム

ディスクベースのファイルシステムは、ハードディスク、CD-ROM、フロッピーディスクなどの物理メディアに格納されます。ディスクベースのファイルシステムは、さまざまな形式で作成できます。利用できる形式は次のとおりです。



| ディスクベースのファイルシステム | 形式の説明                                                                                                                                                                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UFS              | <p>UNIX ファイルシステム (4.3 Tahoe リリースに組み込まれていた BSD Fast File システムに基づく)。UFS は、Solaris オペレーティングシステムのデフォルトのディスクベースファイルシステムです。</p> <p>UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクのフォーマットとスライスへの分割については、<a href="#">第 10 章</a>を参照してください。</p>                                                                    |
| HSFS             | <p>High Sierra、Rock Ridge、および ISO 9660 のファイルシステム。High Sierra は、初めての CD-ROM ファイルシステムです。ISO 9660 は、High Sierra ファイルシステムの公式の標準バージョンです。HSFS ファイルシステムは CD-ROM 上で使用される読み取り専用ファイルシステムです。Solaris HSFS では、ISO 9660 への Rock Ridge 拡張がサポートされるので、CD-ROM 上でも、すべての UFS ファイルシステムのセマンティクスとファイルタイプを提供します。ただし、書き込み可能ではなく、ハードリンクも提供しません。</p> |
| PCFS             | <p>PC ファイルシステム。DOS ベースのパーソナルコンピュータ用に作成された DOS フォーマットのディスク上のデータとプログラムに読み取りと書き込みのアクセスができます。</p>                                                                                                                                                                                                                             |
| UDF              | <p>UDF (Universal Disk Format) ファイルシステム。DVD (Digital Versatile Disc または Digital Video Disc) と呼ばれる光学式メディアテクノロジーに情報を格納するための業界標準形式。</p>                                                                                                                                                                                      |

ディスクベースの各種ファイルシステムは、次のように特定のメディアのタイプに対応しています。

- UFS とハードディスク
- HSFS と CD-ROM
- PCFS とフロッピーディスク
- UDF と DVD

ただし、上記以外の組み合わせも可能です。たとえば、CD-ROM やフロッピーディスクにも、UFS ファイルシステムを格納できます。

## ネットワークベースのファイルシステム

ネットワークベースのファイルシステムは、ネットワークからアクセスされるファイルシステムです。一般に、ネットワークベースのファイルシステムは 1 つのシステム上 (通常はサーバー上) にあり、他のシステムからネットワーク経由でアクセスされません。

NFS で分散された「リソース」(ファイルやディレクトリ)を管理するには、サーバーからそれらのリソースをエクスポートして個々のクライアントシステムでマウントします。詳細については、[272 ページ](#)の「[NFS 環境](#)」を参照してください。

## 仮想ファイルシステム

仮想ファイルシステムは、特殊なカーネル情報と機能へのアクセスを提供するメモリーベースのファイルシステムです。ほとんどの仮想ファイルシステムは、ディスク領域を使用しません。ただし、CacheFS ファイルシステムは、ディスク上のファイルシステムを使用してキャッシュを保持します。また、一時ファイルシステム (TMPFS) などの一部の仮想ファイルシステムは、ディスク上のスワップ空間を使用します。

## CacheFS ファイルシステム

CacheFS™ ファイルシステムを使用すると、リモートファイルシステムや、CD-ROM ドライブのような低速デバイスのパフォーマンスを改善できます。ファイルシステムをキャッシュすると、リモートファイルシステムや CD-ROM から読み込まれたデータは、ローカルシステム上のキャッシュに格納されます。

NFS や CD-ROM ファイルシステムのパフォーマンスとスケーラビリティを向上させるには、CacheFS ファイルシステムを使用してください。CacheFS ソフトウェアは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティを改善する汎用ファイルシステムキャッシュメカニズムです。

CacheFS ソフトウェアは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ソフトウェアはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。また、CacheFS ファイルシステムと AutoFS サービスを組み合わせると、パフォーマンスとスケーラビリティをさらに向上させることができます。

CacheFS ファイルシステムの詳細については、第 18 章を参照してください。

## UDF (Universal Disk Format) ファイルシステム

UDF ファイルシステムは、「DVD」(Digital Versatile Disc または Digital Video Disc) 光学式メディアに情報を格納するための業界標準形式です。

UDF ファイルシステムは、SPARC と x86 の両方のプラットフォームにおいて、動的に読み込み可能な 32 ビットと 64 ビットのモジュールとして提供されます。また、ファイルシステムを作成、マウント、および検査するシステム管理ユーティリティも同時に提供されます。Solaris の UDF ファイルシステムは、サポートされている ATAPI と SCSI の DVD ドライブ、CD-ROM デバイス、ハードディスク、およびフロッピーディスクドライブで機能します。さらに、Solaris の UDF ファイルシステムは UDF 1.50 仕様に完全に準拠しています。

UDF ファイルシステムには次のような機能があります。

- UDF ファイルシステムが入っている業界標準の CD-ROM や DVD-ROM のメディアにアクセスできます。

- さまざまなプラットフォームやオペレーティングシステムで情報を交換できます。
- UDF 形式に基づく DVD ビデオ仕様を使用することで、相互対話性が豊富な放送品質並みの映像や高品質のサウンドを持つ新しいアプリケーションを実装できます。

次の機能は、UDF ファイルシステムにはありません。

- CD-RW (何度でも書き換えが可能な CD メディア) と DVD-RAM への順次ディスク同時記録方式と増分記録方式での書き込み
- UDF 1.50 仕様の一部ではないディスク割り当て、ACL、トランザクションのロギング、ファイルシステムのロック、およびファイルシステムのスレッドなど UFS 構成要素

次に、UDF ファイルシステムの要件を示します。

- Solaris 7 11/99 以降のリリースが動作していること
- SPARC または x86 プラットフォームがサポートされていること
- CD-ROM または DVD-ROM ドライブがサポートされていること

Solaris で実装された UDF ファイルシステムには、次のような互換性があります。

- 業界標準の読み取り / 書き込み UDF バージョン 1.50 のサポート
- 完全に国際化されたファイルシステムのユーティリティ

## 一時ファイルシステム

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べてアクセス速度が高速です。TMPFS を使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドを軽減でき、システムパフォーマンスを改善できます。たとえば、プログラムをコンパイルすると一時ファイルが作成されます。オペレーティングシステムは、これらのファイルを処理する間に大量のディスク処理やネットワーク処理を行います。TMPFS を使用してこれらの一時ファイルを格納すると、その作成、処理、または削除が大幅に高速になります。

TMPFS ファイルシステムのファイルは、永続的に保存されるわけではありません。ファイルシステムのマウントが解除されるとき、システムがシャットダウンまたはリブートされるときに、一時ファイルシステムのファイルは削除されます。

TMPFS は、Solaris オペレーティングシステム内の /tmp ディレクトリのデフォルトのファイルシステムです。UFS ファイルシステムの場合と同様に、/tmp ディレクトリとの間でファイルをコピーまたは移動できます。

TMPFS ファイルシステムは、一時的な退避場所としてスワップ空間を使用します。TMPFS ファイルシステムがマウントされたシステムのスワップ空間が足りないと、次の 2 つの問題が発生する可能性があります。

- TMPFS ファイルシステムは、通常のファイルシステムと同様に容量不足になる可能性がある。

- TMPFS はスワップ空間を割り当ててファイルのデータを保存するので (必要な場合)、一部のプログラムがスワップ空間不足のために実行できなくなる可能性がある。

TMPFS ファイルシステムの作成方法については、[第 16 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 19 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 16 章](#)を参照してください。

## プロセスファイルシステム

プロセスファイルシステム (PROCFS) はメモリー内にあり、/proc ディレクトリには、アクティブなプロセスのプロセス番号別リストが入っています。/proc ディレクトリの内容は、ps などのコマンドで使用されます。デバッガや他の開発ツールも、ファイルシステムコールを使用して、プロセスのアドレス空間にアクセスできます。



---

注意 - /proc ディレクトリ内のファイルは削除しないでください。/proc ディレクトリからプロセスを削除しても、そのプロセスは強制終了されません。/proc ファイルはディスク容量を消費しないため、このディレクトリからファイルを削除してもあまり意味がありません。

---

/proc ディレクトリは、管理が不要です。

## その他の仮想ファイルシステム

次のタイプの仮想ファイルシステムは、参考のために掲載してあります。管理は不要です。

| 仮想ファイルシステム      | 説明                                                  |
|-----------------|-----------------------------------------------------|
| FIFOFS (先入れ先出し) | プロセスにデータへの共通アクセス権を与える名前付きパイプのファイル                   |
| FDFS (ファイル記述子)  | 開いているファイルに、記述子を使用して名前を明示的に与える                       |
| MNTFS           | ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供する      |
| NAMEFS          | ほとんどの場合、ファイル記述子をファイルの先頭に動的にマウントするために STREAMS に使用される |
| SPECFS (特殊)     | キャラクタ型特殊デバイスとブロック型特殊デバイスへのアクセスを提供する                 |
| SWAPFS          | カーネルがスワッピングに使用するファイルシステム                            |

## 拡張ファイル属性

UFS、NFS、および TMPFS ファイルシステムは、拡張ファイル属性を包含するように機能拡張されました。拡張ファイル属性によりアプリケーション開発者は、ファイルを特定の属性に関連付けることができます。たとえば、ウィンドウシステムファイル管理アプリケーションの開発者は、表示アイコンをファイルに関連付けることができます。拡張ファイル属性は、論理的には、ターゲットファイルに関連付けられている隠しディレクトリ内のファイルとして表されます。

runat コマンドを使用すると、属性を追加したり、拡張属性の名前空間に入っているシェルコマンドを実行したりできます。拡張属性の名前空間とは、指定のファイルに関連付けられている隠し属性ディレクトリのことです。

runat コマンドを使用して属性をファイルに追加するには、最初に属性ファイルを作成する必要があります。

```
$ runat filea cp /tmp/attrdata attr.1
```

次に、runat コマンドを使用して、ファイルの属性をリストに表示します。

```
$ runat filea ls -l
```

詳細については、runat (1) のマニュアルページを参照してください。

ファイル属性の照会、コピー、または検索に使用できる属性対応のオプションを提供することにより、Solaris ファイルシステムコマンドの多くは、ファイルシステム属性をサポートするよう変更されました。詳細については、各ファイルシステムコマンドのマニュアルページを参照してください。

---

## ファイルシステム管理用のコマンド

ほとんどのファイルシステム管理コマンドには、汎用コマンドとファイルシステムに固有のコマンドの2種類があります。可能な場合には、常に汎用コマンドを使用してください。汎用コマンドは、ファイルシステム固有のコマンドを呼び出します。次の表に、ファイルシステム管理用の汎用コマンドを示します。これらのコマンドは、`/usr/sbin` ディレクトリに入っています。

表 15-1 ファイルシステム管理用の汎用コマンド

| コマンド                  | マニュアルページ                   | 説明                                                                       |
|-----------------------|----------------------------|--------------------------------------------------------------------------|
| <code>clri</code>     | <code>clri (1M)</code>     | i ノードをクリアする。                                                             |
| <code>df</code>       | <code>df (1M)</code>       | 空きディスクブロック数とファイル数を出力する。                                                  |
| <code>ff</code>       | <code>ff (1M)</code>       | ファイルシステムのファイル名と統計情報を表示する。                                                |
| <code>fsck</code>     | <code>fsck (1M)</code>     | ファイルシステムの整合性を検査し、検出された損傷を修復する。                                           |
| <code>fsdb</code>     | <code>fsdb (1M)</code>     | ファイルシステムをデバッグする。                                                         |
| <code>fstyp</code>    | <code>fstyp (1M)</code>    | ファイルシステムのタイプを調べる。                                                        |
| <code>labelit</code>  | <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</code>     | <code>mkfs (1M)</code>     | 新しいファイルシステムを作成する。                                                        |
| <code>mount</code>    | <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</code> | <code>mountall (1M)</code> | 仮想ファイルシステムテーブル ( <code>/etc/vfstab</code> ) 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>ncheck</code>   | <code>ncheck (1M)</code>   | パス名とその i ノード番号のリストを生成する。                                                 |

表 15-1 ファイルシステム管理用の汎用コマンド (続き)

| コマンド      | マニュアルページ      | 説明                                                          |
|-----------|---------------|-------------------------------------------------------------|
| umount    | mount (1M)    | ローカルおよびリモートのファイルシステムをマウント解除する。                              |
| umountall | mountall (1M) | 仮想ファイルシステムテーブル (/etc/vfstab) に指定されているすべてのファイルシステムをマウント解除する。 |
| volcopy   | volcopy (1M)  | ファイルシステムのイメージコピーを作成する。                                      |

## ファイルシステムコマンドによるファイルシステムタイプの判断

汎用ファイルシステムコマンドは、次の順序でファイルシステムのタイプを判断します。

1. 指定されている場合は、`-F` オプションで指定されているファイルシステムのタイプから判断します。
2. 特殊デバイスを `/etc/vfstab` ファイルのエントリと突き合わせて判断します (`special` が指定されている場合)。たとえば `fsck` は、まず `fsck device` フィールドと突き合わせて一致するエントリを検索します。一致するエントリが見つからなければ、特殊デバイスフィールドと突き合わせてチェックします。
3. ローカルファイルシステムの場合は `/etc/default/fs` ファイル内に指定されたデフォルトを使用し、リモートファイルシステムの場合は `/etc/dfs/fstypes` ファイル内に指定されたデフォルトを使用して判断します。

## 汎用コマンドと専用コマンドのマニュアルページ

汎用コマンドと専用コマンドについては、『man pages section 1M: System Administration Commands』を参照してください。ファイルシステムの汎用コマンドのマニュアルページには、汎用コマンドオプションに関する情報だけが記載されています。専用コマンドのマニュアルページには、該当するファイルシステムに固有のオプション情報が記載されています。特定のマニュアルページを見つけるには、汎用コマンド名の末尾にアンダースコアとファイルシステムタイプの略称を追加してください。たとえば、UFS ファイルシステムのマウントについてのマニュアルページを参照するには、次のように入力します。

```
$ man mount_ufs
```

## デフォルトの Solaris ファイルシステム

Solaris UFS ファイルシステムは階層構造になっており、ルートディレクトリ (/) から始まり、下位に多数のディレクトリが形成されています。Solaris のインストールプロセスは、デフォルトのディレクトリセットをインストールし、一連の規則を適用して類似するタイプのファイルをグループ化します。次の表に、デフォルトの Solaris ファイルシステムの概要を示します。

表 15-2 デフォルトの Solaris ファイルシステム

| ファイルシステムまたはディレクトリ      | ファイルシステムのタイプ | 説明                                                                                                                                                              |
|------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ルート (/)                | UFS          | 階層ファイルツリーの最上位。ルートディレクトリには、カーネル、デバイスドライバ、システムのブートに使用されるプログラムなど、システム処理に欠かせないディレクトリとファイルが入っている。また、ローカルとリモートのファイルシステムをファイルツリーに接続できるマウントポイントディレクトリも入っている。            |
| /usr                   | UFS          | 他のユーザーと共有できるシステムファイルとディレクトリ。特定のタイプのシステム上でのみ実行できるファイルは、/usr ファイルシステムに入っている (SPARC 実行可能ファイルなど)。どのタイプのシステム上でも使用できるファイル (マニュアルページなど) は、/usr/share ディレクトリに入っている。     |
| /export/home または /home | NFS、UFS      | ユーザーのホームディレクトリのマウントポイント。ホームディレクトリには、そのユーザーの作業ファイルが格納される。デフォルトでは、/home ディレクトリは自動マウントされるファイルシステムである。スタンドアロンシステムでは、/home ディレクトリはローカルディスクスライス上の UFS ファイルシステムの場合がある。 |
| /var                   | UFS          | ローカルシステムの使用中に変化または拡大する可能性のあるシステムファイルとディレクトリ。これには、システムログ、vi と ex のバックアップファイル、および uuwp ファイルが含まれる。                                                                 |
| /opt                   | NFS、UFS      | オプションの Sun 以外のソフトウェア製品のマウントポイント。一部のシステムでは、/opt ディレクトリはローカルディスクスライス上の UFS ファイルシステムの場合がある。                                                                        |
| /tmp                   | TMPFS        | システムがブートされるか、/tmp ファイルシステムがマウント解除されるたびに消去される一時ファイル。                                                                                                             |
| /proc                  | PROCFS       | アクティブなプロセスの番号別リスト。                                                                                                                                              |



表 15-2 デフォルトの Solaris ファイルシステム (続き)

| ファイルシステムまたはディレクトリ | ファイルシステムのタイプ | 説明                                                      |
|-------------------|--------------|---------------------------------------------------------|
| /etc/mnttab       | MNTFS        | ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供するファイルシステム。 |
| /var/run          | TMPFS        | システムのブート後に不要になる一時ファイルを格納するためのファイルシステム。                  |

システムを動作させるには、ルート (/) と /usr のファイルシステムが必要です。/usr ファイルシステムに置かれているもっとも基本的なコマンドの一部 (mount など) は、システムのブート時や、システムがシングルユーザーモードで実行しており、/usr ファイルシステムがマウントされていない場合でも使用できるように、ルート (/) ファイルシステムにも置かれています。ルート (/) と /usr ファイルシステムのデフォルトディレクトリの詳細については、第 21 章を参照してください。

## スワップ空間

Solaris オペレーティングシステムは、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で作成されているため、スワップ空間を割り当て、その使われ方を監視して、必要に応じてスワップ空間を追加する方法を知っておく必要があります。スワップ空間の概要とスワップ空間を追加する手順については、第 19 章を参照してください。

## UFS ファイルシステム

UFS は、Solaris オペレーティング環境内のデフォルトのディスクベースファイルシステムです。ほとんどの場合、ディスクベースのファイルシステムを管理するときには、UFS を管理していることとなります。UFS ファイルシステムの機能は次の通りです。

| UFS の機能       | 説明                                                                                                                                                          |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 状態フラグ         | ファイルシステムの状態を、クリーン、安定、使用中、ロギング処理、または不明として示します。これらのフラグによって、必要のないチェックをファイルシステム上で行わなくて済みます。ファイルシステムが「クリーン」状態、「安定」状態、または「ロギング処理」状態になっていると、ファイルシステムのチェックは実行されません。 |
| 拡張基礎タイプ (EFT) | 32 ビットのユーザーID (UID)、グループID (GID)、およびデバイス番号。                                                                                                                 |
| 大規模ファイルシステム   | 最大 16T バイトまで増やせるファイルシステムに、およそ 1T バイトの大きさのファイルを指定できます。EFI ディスクラベルを使ってディスク上にマルチテラバイトの UFS ファイルシステムを作成できます。                                                    |

UFS ファイルシステム構造の詳細については、[第 21 章](#)を参照してください。

## 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。次にいくつかの推奨事項を示します。

- 作業負荷を異なる入出力システムやディスクドライブ間でできるだけ均等に分散します。/export/home ファイルシステムやスワップ空間を異なるディスク間で均等に分散させます。
- プロジェクトの個々の部分やグループのメンバーを同じファイルシステム内に入れます。
- 1 ディスク当たりのファイルシステム数をできるだけ少なくします。通常、システム(またはブート) ディスク上には、ルート (/)、/usr、スワップ空間の 3 つのファイルシステムがあります。他のディスクには、1 つまたは多くても 2 つのファイルシステム (1 つはなるべく追加スワップ空間) を作成します。多数の小規模なファイルシステムに分割しすぎるよりもファイルシステム数を少なくして余地を設ける方が、ファイルがフラグメントに分割される可能性が小さくなります。容量の大きいテープドライブを使用し、ufsdump コマンドで複数のボリュームを処理できるようにしておけば、大規模なファイルシステムでも簡単にバックアップをとることができます。
- 絶えずきわめて小さいファイルを作成するユーザーがいる場合は、i ノード数を増やして別のファイルシステムを作成することを検討します。ただし、ほとんどのサイトでは、類似するタイプのファイルを同じファイルシステム内で保管する必要はありません。

デフォルトのファイルシステムパラメータや、新しい UFS ファイルシステムを作成する手順については、[第 16 章](#)を参照してください。

## UFS ロギング

UFS ロギングは、1つの完全な UFS 操作を構成する複数のメタデータ変更を、1つのトランザクションにまとめます。一連のトランザクションは、ディスク上のログに記録された後で、実際の UFS ファイルシステムのメタデータに適用されます。

システムはリポート時に、不完全なトランザクションを廃棄しますが、完結している操作のトランザクションは適用します。完結しているトランザクションだけが適用されるために、ファイルシステムの整合性が保たれます。通常であればシステムコールの実行が中断され、UFS ファイルシステムの整合性が確保できないシステムクラッシュ時にも、ファイルシステムの整合性が保たれます。

UFS ロギングには2つの長所があります。

- トランザクションログによりファイルシステムの整合性がすでに保持されている場合、システムがクラッシュしたとき、またはクリーンでないシステム停止が発生したときに、`fsck` コマンドを実行する必要がないことがあります。クリーンでない停止については、346 ページの「`fsck` コマンドでチェックして修復される内容」を参照してください。
- Solaris 9 12/02 以降、UFS ロギングのパフォーマンスは、非ロギングファイルシステムのパフォーマンスレベルを上回るようになりました。このようにパフォーマンスが改善されるのは、ロギングを有効にしたファイルシステムでは、同じデータに対する複数回の更新が1回の更新に変換されるため、オーバーヘッドの大きいディスク操作の実行回数が減るからです。

ログはファイルシステムの空きブロックから割り当てられ、1G バイトのファイルシステムごとに約 1M バイトのサイズ (合計で 64M バイトまで) が割り当てられます。ログは、いっぱいになるとフラッシュされます。また、ファイルシステムがマウント解除されたとき、あるいは `lockfs` コマンドを実行したときにも、ログがフラッシュされます。

UFS ロギングは、すべての UFS ファイルシステムでデフォルトで有効になります。

UFS ロギングを無効にする必要がある場合は、`/etc/vfstab` ファイル内のファイルシステムのエントリに `nologging` オプションを追加するか、ファイルシステムを手動でマウントするときに `nologging` オプションを指定します。

UFS ロギングを有効にする必要がある場合は、`/etc/vfstab` ファイル内で `mount` コマンドに `-o logging` オプションを指定するか、ファイルシステムを手動でマウントするときに `mount` コマンドに `-o logging` オプションを指定します。ログは、ルート (/) ファイルシステムを含む、任意の UFS ファイルシステムで有効にできます。また、`fsdb` コマンドには、UFS ロギングをサポートするための新しいデバッグコマンドが用意されています。

一部のオペレーティングシステムでは、ロギングが有効になっているファイルシステムを「ジャーナル」ファイルシステムと呼びます。

## UFS スナップショット

`fssnap` コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。スナップショットは、バックアップ操作のためのファイルシステムの一時的イメージです。

詳細については、[第 24 章](#)を参照してください。

## UFS 直接入出力

直接入出力の目的は、大容量入出力処理のスピードを速くすることです。大容量入出力処理では、大規模ファイル (256K バイトを超える) を転送するために、大容量のバッファサイズを使用します。

UFS の直接入出力を使用すると、データベースエンジンなど、独自の内部バッファリングを行うアプリケーションにメリットがあります。Solaris 8 1/01 リリースから開始された UFS の直接入出力は、`raw` デバイスのアクセス時に見られる同様の入出力の並行処理に対応するよう改善されました。現在では、パフォーマンスがわずかに低下するだけで、ファイルシステムのネーミングや柔軟性がもたらすメリットを受けることができます。データベースの製造元を調べて、その製品構成オプションで UFS の直接入出力を有効にできるかどうかを確認してください。

`mount` コマンドに `forcedirectio` オプションを使用しても、直接入出力をファイルシステムで有効にできます。直接入出力を有効にしてパフォーマンスが向上するのは、ファイルシステムが大量の連続するデータを転送するときだけです。

`forcedirectio` オプションでファイルシステムをマウントするとき、データはユーザーのアドレス空間とディスクの間で直接伝送されます。直接入出力がファイルシステムで無効な場合、ユーザーのアドレス空間とディスクの間で転送されるデータは、まず、カーネルアドレス空間にバッファされます。

デフォルトでは、UFS ファイルシステムでは直接入出力は行われません。詳細については、`mount_ufs(1M)` のマニュアルページを参照してください。

---

## ファイルシステムのマウントとマウント解除

ファイルシステム上のファイルにアクセスするには、ファイルシステムをマウントする必要があります。ファイルシステムのマウントとは、ファイルシステムを特定のディレクトリ (マウントポイント) に接続し、システムで使用可能にすることです。ルート (/) ファイルシステムは、常にマウントされています。他のファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりできます。

ファイルシステムをマウントすると、そのファイルシステムがマウントされている間は、マウントポイントのディレクトリ内に実際に存在しているファイルやディレクトリは使用できなくなります。これらのファイルはマウント処理の影響を永続的に受けるわけではなく、ファイルシステムをマウント解除すると再び使用できるようになります。通常は存在はするがアクセスできないファイルは混乱の原因になるので、マウントディレクトリを空にしておきます。

たとえば、次の図は、ルート (/) ファイルシステムと、その下の `sbin`、`etc`、`opt` の各サブディレクトリから始まるローカルファイルシステムを示しています。

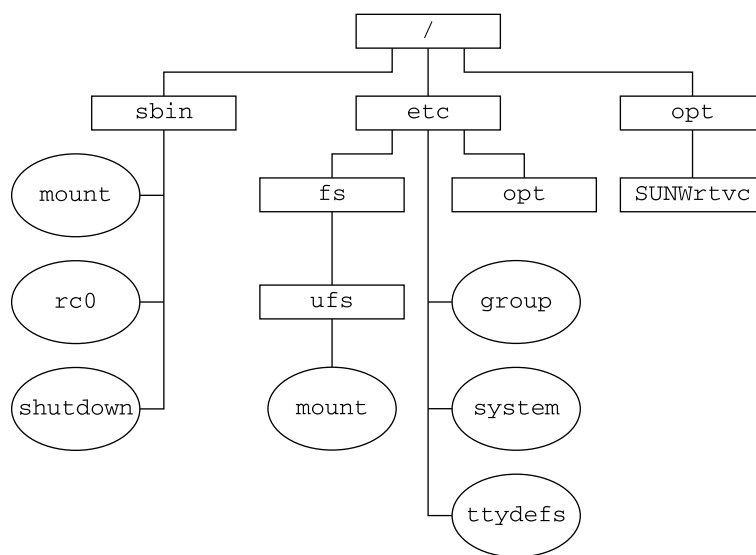
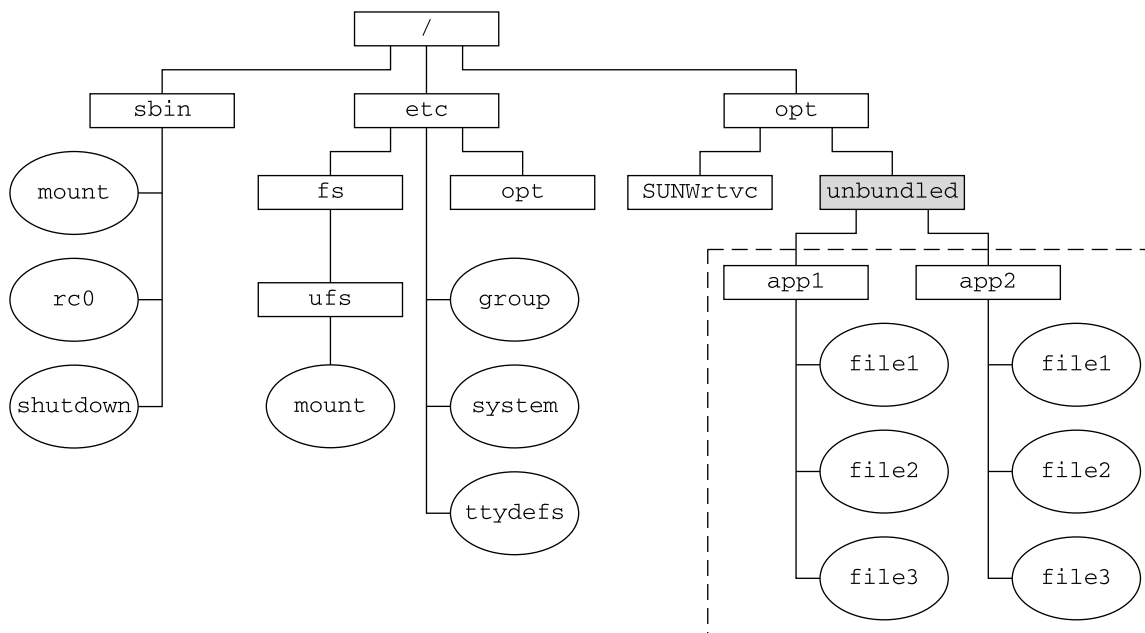


図 15-1 ルート (/) ファイルシステムの例

別パッケージの製品が含まれている `/opt` ファイルシステムからローカルファイルシステムにアクセスするには、次の作業を行います。

- まず、ファイルシステムをマウントするためのマウントポイントとして使用するディレクトリを作成しなければなりません (たとえば、`/opt/unbundled`)。
- マウントポイントを作成すると、`mount` コマンドでファイルシステムをマウントでき、`/opt/unbundled` 内のすべてのファイルとディレクトリにアクセスできるようになります (次の図を参照)。

ファイルシステムのマウント方法については、[第 17 章](#)を参照してください。



□ マウントポイント

[ ] ファイルシステム

図 15-2 ファイルシステムのマウント

## マウントされたファイルシステムテーブル

ファイルシステムをマウントまたはマウント解除すると、現在マウントされているファイルシステムのリストを使用して、`/etc/mnttab` (マウントテーブル) ファイルが変更されます。このファイルの内容は `cat` または `more` コマンドを使用して表示できますが、編集することはできません。次にマウントテーブルファイル `/etc/mnttab` の例を示します。

```

$ more /etc/mnttab
/dev/dsk/c0t0d0s0 / ufs rw,intr,largefiles,onerror=panic,suid,dev=2200000 938557523
/proc /proc proc dev=3180000 938557522
fd /dev/fd fd rw,suid,dev=3240000 938557524
mnttab /etc/mnttab mntfs dev=3340000 938557526
swap /var/run tmpfs dev=1 938557526
swap /tmp tmpfs dev=2 938557529
/dev/dsk/c0t0d0s7 /export/home ufs rw,intr,largefiles,onerror=panic,suid,dev=2200007 ...
$
  
```

## 仮想ファイルシステムテーブル

アクセスするたびにファイルシステムを手動でマウントするのは、時間がかり、またまちがいが起こりやすい作業です。この問題を解決するために、仮想ファイルシステムテーブル (/etc/vfstab ファイル) にファイルシステムのリストとそのマウント方法を指定できます。

/etc/vfstab ファイルは、2つの重要な機能を持っています。

- 1つは、システムブート時に自動的にマウントするファイルシステムを指定できることです。
- もう1つは、マウントポイント名だけでファイルシステムをマウントできることです。これは、/etc/vfstab ファイルにマウントポイントと実際のデバイススライスの名とのマッピングを指定することができるためです。

デフォルトの /etc/vfstab ファイルは、システムをインストールするときに作成され、その内容はシステムソフトウェアをインストールするときに行なった選択によって異なります。ただし、システムの /etc/vfstab ファイルはいつでも編集できます。エントリを追加するときに指定する必要がある主な情報は、ファイルシステムが置かれているデバイス、マウントポイントの名前、ファイルシステムのタイプ、システムブート時に自動的にマウントするかどうか (mountall コマンドを使用する)、およびマウントオプションです。

次の例は、/etc/vfstab ファイルの内容を示しています。コメント行は # で始まります。この例は、2つのディスク (c0t0d0 と c0t3d0) を持つシステムの /etc/vfstab ファイルを示しています。

```
$ more /etc/vfstab
#device      device          mount           FS      fsck  mount  mount
#to mount    to fsck         point          type    pass  at boot options
/dev/dsk/c0t0d0s0 /dev/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 ファイルシステムが自動的にマウントポイント /test にマウントされることを示しています。ルート (/) と /usr では、mount at boot フィールドの値が no に指定されていることに注意してください。これはこの2つのファイルシステムが、mountall コマンドを実行する前にブートシーケンスの一部としてカーネルによってマウントされるからです。

/etc/vfstab の各フィールドの説明、およびこのファイルの編集/使用方法については、[第 17 章](#)を参照してください。

## NFS 環境

NFS は、1つのシステム (通常はサーバー) の「リソース」 (ファイルやディレクトリ) をネットワーク上の他のシステムと共有するための分散型ファイルシステムサービスです。たとえば、Sun 以外のアプリケーションやソースファイルを他のシステム上のユーザーと共有できます。

NFS は、リソースの実際の物理的な位置をユーザーが意識しなくてすむようにします。よく使用されるファイルのコピーをシステムごとに配置しなくても、あるシステムのディスク上にコピーを1つ配置することによって NFS は、他のすべてのシステムがそのコピーにネットワークからアクセスできるようにします。NFS の環境では、リモートファイルシステムは、実際にはローカルシステムと区別が付きません。

システムは、ネットワーク上で共有するリソースがあるときに、NFS サーバーになります。サーバーは、現在共有されているリソースとそのアクセス制限 (読み取り / 書き込み、読み取り専用アクセスなど) のリストを管理します。

リソースを共有する場合は、リモートシステムにマウントできるように、そのリソースを使用可能な状態にします。

リソースを共有するには、次の方法があります。

- `share` コマンドまたは `shareall` コマンドを使用する。
- `/etc/dfs/dfstab` (分散ファイルシステムテーブル) ファイルにエントリを追加し、システムをリポートする。

リソースを共有する方法については、第 17 章を参照してください。NFS の詳細については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の第 13 章「ネットワークファイルシステムの管理 (概要)」を参照してください。

## 自動マウントまたは AutoFS

NFS ファイルシステムリソースをマウントするには、自動マウント (または AutoFS) というクライアント側のサービスを使用します。このサービスによってシステムは、ユーザーがアクセスしたときに自動的に NFS リソースをマウントまたはマウント解除できます。ユーザーがファイルシステム内のファイルを使用している間、ファイルシステムリソースはマウントされたままになります。リソースが一定の時間アクセスされなかった場合、リソースは自動的にマウント解除されます。

次に、AutoFS の特徴を示します。

- システムブート時に NFS リソースをマウントする必要がないために、ブート時間が短くなります。
- NFS リソースをマウントまたはマウント解除するために、スーパーユーザーのパスワードを知っている必要はありません。
- NFS リソースは使用されるときにだけマウントされるために、ネットワークトラフィックが軽減されます。



AutoFS サービスは automount ユーティリティによって初期化されます。このコマンドは、システムのブート時に自動的に実行されます。automountd デーモンは永続的に動作し、必要に応じて NFS ファイルシステムをマウントまたはマウント解除します。デフォルトでは、/home ファイルシステムは automount デーモンによってマウントされます。

AutoFS では、同じファイルシステムを提供するサーバーを複数指定できます。このような方法では、1つのサーバーがダウンしても、AutoFS が他のマシンからファイルシステムをマウントすることができます。

AutoFS の設定と管理方法については、『Solaris のシステム管理 (IP サービス)』を参照してください。

---

## ファイルシステムのタイプを調べる

ファイルシステムのタイプは、次のいずれかを使用して判断できます。

- 仮想ファイルシステムテーブル (/etc/vfstab ファイル) 内の FS type フィールド
- ローカルファイルシステムの /etc/default/fs ファイル
- NFS ファイルシステムの /etc/dfs/fstypes ファイル

## ファイルシステムのタイプを調べる方法

この手順は、ファイルシステムがマウントされているかどうかにかかわらず、使用できます。

grep コマンドを使用して、ファイルシステムのタイプを調べます。

```
$ grep mount-point fs-table
```

*mount-point* ファイルシステムタイプを調べるファイルシステムのマウントポイント名を指定する。たとえば、/var ディレクトリ

*fs-table* ファイルシステムのタイプを調べるファイルシステムテーブルへの絶対パスを示す。ファイルシステムがマウントされている場合、*fs-table* は /etc/mnttab。マウントされていない場合、*fs-table* は /etc/vfstab

マウントポイントの情報が表示されます。

---

注 - ディスクスライスの raw デバイス名がわかる場合、fstyp コマンドで、ファイルシステムのタイプを調べることができます (そのディスクスライスにファイルシステムが含まれている場合)。詳細については、fstyp (1M) のマニュアルページを参照してください。

---

## 例 — ファイルシステムのタイプを調べる

次の例では、/etc/vfstab を使用して、/export ファイルシステムのタイプを調べます。

```
$ grep /export /etc/vfstab
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /export ufs 2 yes -
$
```

次の例では、/etc/mnttab ファイルを使用して、現在マウントされているフロッピーディスク (vold でマウントされたもの) のファイルシステムのタイプを調べます。

```
$ grep /floppy /etc/mnttab
/vol/dev/diskette0/unnamed_floppy /floppy/unnamed_floppy pcfs rw,
nohidden,nofoldcase,dev=16c0009 89103376
$
```

## 第 16 章

---

# UFS、TMPFS、LOFS ファイルシステムの作成 (手順)

---

この章では、UFS ファイルシステム、一時ファイルシステム (TMPFS)、およびループバックファイルシステム (LOFS) の作成方法について説明します。UFS ファイルシステムについては、`newfs (1M)` コマンドを使用してファイルシステムをハードディスク上に作成する方法を示します。TMPFS と LOFS は仮想ファイルシステムであるため、これらのファイルシステムを「実際に使用する」には、ファイルシステムをマウントします。

この章で説明する手順は次のとおりです。

- [277 ページの「UFS ファイルシステムを作成する方法」](#)
- [279 ページの「TMPFS ファイルシステムを作成する方法」](#)
- [280 ページの「LOFS ファイルシステムを作成する方法」](#)

---

注 - UFS と DOS のファイルシステムをリムーバブルメディア上に作成する手順については、[第 1 章](#)を参照してください。

---

---

## UFS ファイルシステムの作成

UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクスライスとは物理的なディスクのサブセットで、連続するブロックからなる 1 つの範囲のことです。スライスはスワップ空間などの raw デバイスとして使用することも、ディスクベースのファイルシステムとして使用することもできます。ディスクのフォーマットとディスクのスライスへの分割の詳細については、[第 10 章](#)を参照してください。

Solaris ボリュームマネージャなどのボリューム管理製品を使用すると、精錬された (単一のスライスや単一のディスク境界を拡張する) 「ボリューム」を作成できます。ボリュームの使用方法については、『*Solaris ボリュームマネージャの管理*』を参照してください。

---

注 – Solaris のデバイス名は、用語「スライス (デバイス名内の文字は s)」を使用して、スライス番号を参照します。スライスは「パーティション」と呼ばれていました。

---

UFS ファイルシステムは、インストール手順の一部として Solaris オペレーティング環境によって自動的に作成されるので、UFS ファイルシステムを作成しなければならないことはほとんどありません。次の場合には、UFS ファイルシステムを作成する (または作成し直す) 必要があります。

- ディスクを追加または交換する場合
- 既存のパーティション構造を変更する場合
- ファイルシステム全体を復元する場合

`newfs` コマンドを使用するのが、UFS ファイルシステムを作成する標準的な方法です。`newfs` コマンドは `mkfs` コマンドの使いやすいフロントエンドであり、新しいファイルシステムを作成します。`newfs` コマンドのパラメータ (デフォルトでは 1 シリンダ当たりのトラック数や 1 トラック当たりのセクター数など) は、新しいファイルシステムが作成されるディスクのラベルから読み取られます。選択したオプションは、`mkfs` コマンドに渡され、ファイルシステムが作成されます。

## newfs コマンドのデフォルトのパラメータ

ディスクスライス上に新しいファイルシステムを作成するには、ほとんどの場合に `newfs` コマンドを使用します。次の表に、`newfs` コマンドで使用するデフォルトのパラメータを示します。

| パラメータ     | デフォルト値                                                                                |
|-----------|---------------------------------------------------------------------------------------|
| ブロックサイズ   | 8K バイト                                                                                |
| フラグメントサイズ | 1K バイト                                                                                |
| 最小空き領域    | ((64M バイト/パーティションサイズ) * 100) で算出した値をもっとも近い整数に切り捨てる。値は、パーティションサイズの 1% から 10% の範囲に制限される |
| 回転待ち      | 0                                                                                     |
| 最適化タイプ    | 時間                                                                                    |
| i ノード数    | 2K バイトのディスク領域ごとに 1 個                                                                  |

## ▼ UFS ファイルシステムを作成する方法

手順 1. 次の前提条件を満たしているかどうかを確認します。

- a. ディスクをフォーマットし、スライスに分割しておかなければならない。  
ディスクのフォーマットとスライスへの分割については、第 10 章を参照してください。
- b. ファイルシステムを格納するスライスのデバイス名を知っていなければならない。  
ディスク番号とディスクスライス番号を調べる方法については、第 11 章を参照してください。
- c. 既存の **UFS** ファイルシステムを作成し直す場合は、そのマウントを解除する。
- d. スーパーユーザーになるか、同等の役割を引き受けていなければならない。

2. ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdisk/device-name
```

- N** newfs コマンドが **mkfs** コマンドに渡すパラメータを表示する。ファイルシステムは実際に作成されない。newfs コマンドをテストするのに好ましい方法。
- b size** ファイルシステムのブロックサイズを 1 ブロックあたり 4096 または 8192 バイトで指定する。デフォルトは 8192 バイト。
- i bytes** i ノード 1 個当たりのバイト数を指定する。デフォルトはディスクのサイズによって異なる。詳細については、newfs (1M) のマニュアルページを参照。
- device-name** 新しいファイルシステムを作成するディスクデバイス名を指定する。
- システムから、確認を促すプロンプトが表示されます。



---

注意 - この手順を実行する前に、スライスのデバイス名が正しく指定されていることを確認してください。間違ったスライスを指定すると、その内容は新しいファイルシステムの作成時に消去されます。そして、システムがパニックを起こす原因となる可能性があります。

---

3. **UFS** ファイルシステムが作成されていることを確認するには、新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

**device-name** 引数は、新しいファイルシステムを格納するディスクデバイスの名前を指定します。

fsck コマンドは、新しいファイルシステムの整合性をチェックして、問題があれば通知し、問題を修復する前にプロンプトを表示します。fsck コマンドの詳細については、第 20 章または fsck(1M) のマニュアルページを参照してください。

### 例 16-1 UFS ファイルシステムの作成

次の例は、/dev/rdisk/c0t1d0s7 上に UFS ファイルシステムを作成する方法を示しています。

```
# newfs /dev/rdisk/c0t1d0s7
/dev/rdisk/c0t1d0s7: 725760 sectors in 720 cylinders of 14 tracks, 72 sectors
      354.4MB in 45 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
613376, 629584, 645792, 662000, 678208, 694416, 710624,
#
```

参照 UFS ファイルシステムをマウントし、使用可能にするには、第 17 章に進みます。

---

## 一時ファイルシステム (TMPFS) の作成

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べて読み書きの速度が高速です。TMPFS ファイルシステムを使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドが軽減されるのでシステムのパフォーマンスを向上できます。TMPFS ファイルシステム内のファイルは、リブートまたはマウント解除すると削除されません。

複数の TMPFS ファイルシステムを作成した場合は、すべてのファイルシステムが同じシステムリソースを使用するという点に注意してください。mount コマンドの `-o size` オプションを使用して TMPFS のサイズを制限しないと、ある TMPFS ファイルシステムで作成されたファイルが、他の TMPFS ファイルシステムのための領域を使い切ってしまう可能性があります。

詳細については、tmpfs(7FS) のマニュアルページを参照してください。

## ▼ TMPFS ファイルシステムを作成する方法

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、**TMPFS** ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir /mount-point
/mount-point は、TMPFS ファイルシステムがマウントされるディレクトリです。
```

3. **TMPFS** ファイルシステムをマウントします。

```
# mount -F tmpfs [-o size=number] swap mount-point
-o size=number    TMPFS ファイルシステムのサイズ制限を M バイト単位で示す。
mount-point       TMPFS ファイルシステムがマウントされるディレクトリを示す。
```

ブート時にシステムが TMPFS ファイルシステムを自動的にマウントするように設定するには、[例 16-3](#) を参照してください。

4. **TMPFS** ファイルシステムが作成されていることを確認します。

```
# mount -v
```

### 例 16-2 TMPFS ファイルシステムを作成する

次の例は、TMPFS ファイルシステム `/export/reports` を作成およびマウントし、そのサイズを 50M バイトに制限する方法を示しています。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50m swap /export/reports
```

### 例 16-3 ブート時に TMPFS ファイルシステムをマウントする

ブート時にシステムが自動的に TMPFS ファイルシステムをマウントするように設定するには、`/etc/vfstab` のエントリを追加します。次の例は、システムのブート時に `/export/test` を TMPFS ファイルシステムとしてマウントする `/etc/vfstab` ファイルのエントリを示しています。 `size=number` オプションを指定していないため、`/export/test` の TMPFS ファイルシステムのサイズは利用できるシステムリソースによってのみ制限されます。

```
swap - /export/test tmpfs - yes -
```

`/etc/vfstab` ファイルの詳細については、[287 ページの「`/etc/vfstab` ファイルのフィールドの説明](#)」を参照してください。

---

## ループバックファイルシステム (LOFS) の作成

LOFS ファイルシステムは、既存のファイルシステムへの代替パスを提供する仮想ファイルシステムです。他のファイルシステムを LOFS ループバックファイルシステムにマウントしても、元のファイルシステムは変化しません。

詳細については、`lofs` (7FS) のマニュアルページを参照してください。



---

注意 - LOFS ファイルシステムは慎重に作成してください。LOFS ファイルシステムは仮想ファイルシステムなので、ユーザーやアプリケーションを混乱させる可能性があります。

---

### ▼ LOFS ファイルシステムを作成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、**LOFS** ファイルシステムとしてマウントするディレクトリを作成します。  

```
# mkdir loopback-directory
```
  3. 新しく作成したディレクトリに対して、適切なアクセス権と所有権を設定します。
  4. 必要に応じて、**LOFS** ファイルシステムをマウントするマウントポイントを作成します。  

```
# mkdir /mount-point
```
  5. **LOFS** ファイルシステムをマウントします。  

```
# mount -F lofs loopback-directory /mount-point
```

|                                 |                                    |
|---------------------------------|------------------------------------|
| <code>loopback-directory</code> | 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 |
| <code>/mount-point</code>       | LOFS ファイルシステムをマウントするディレクトリを指定する    |
  6. **LOFS** ファイルシステムがマウントされていることを確認します。  

```
# mount -v
```



#### 例 16-4 LOFS ファイルシステムを作成およびマウントする

次の例は、新しいソフトウェアを、実際にはインストールしないで、ループバックファイルシステムとして /new/dist ディレクトリに作成、マウント、およびテストする方法を示しています。

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

#### 例 16-5 ブート時に LOFS ファイルシステムをマウントする

ブート時にシステムが自動的に LOFS ファイルシステムをマウントするように設定するには、/etc/vfstab ファイルの最後にエントリを追加します。次の例は、ルート (/) ファイルシステムの LOFS ファイルシステムを /tmp/newroot にマウントする /etc/vfstab ファイルのエントリを示しています。

```
/ - /tmp/newroot lofs - yes -
```

ループバックファイルシステムのエントリは、/etc/vfstab ファイル内の最後のエントリでなければなりません。ループバックファイルシステムの /etc/vfstab エントリが、そこに組み込まれるファイルシステムよりも前にあると、ループバックファイルシステムをマウントできません。

参照 [/etc/vfstab ファイルの詳細については、287 ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。](#)



## 第 17 章

---

# ファイルシステムのマウントとマウント解除 (手順)

---

この章では、ファイルシステムをマウントしたり、マウント解除する方法について説明します。

この章で説明する手順は次のとおりです。

- 289 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」
- 289 ページの「`/etc/vfstab` ファイルにエントリを追加する方法」
- 291 ページの「1つのファイルシステムをマウントする方法 (`/etc/vfstab` ファイル)」
- 292 ページの「UFS ファイルシステムのマウント方法 (`mount` コマンド)」
- 293 ページの「大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (`mount` コマンド)」
- 294 ページの「NFS ファイルシステムのマウント方法 (`mount` コマンド)」
- 295 ページの「x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (`mount` コマンド)」
- 297 ページの「ファイルシステムのマウント解除を確認する方法」
- 297 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」
- 298 ページの「1つのファイルシステムをマウント解除する方法」

---

## ファイルシステムのマウントの概要

ファイルシステムを作成したら、そのファイルシステムをシステムで使用できるようにする必要があります。ファイルシステムを使用できるようにするには、マウントします。マウントしたファイルシステムは、システムのディレクトリツリー内の指定したマウントポイントに接続されます。ルート (`/`) ファイルシステムは、常にマウントされています。

次の表に、ファイルシステムをその用途に応じてマウントするためのガイドラインを示します。

| 必要なマウントの種類                            | 推奨されるマウント方法                                                                                                                                                                                                                                                                                                     |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ローカルまたはリモートのファイルシステムをときどきマウントする       | コマンド行から手動で <code>mount</code> コマンドを入力する。                                                                                                                                                                                                                                                                        |
| ローカルのファイルシステムを頻繁にマウントする               | <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに、自動的にファイルシステムをマウントする。                                                                                                                                                                                                                                |
| リモートのファイルシステムを頻繁にマウントする (ホームディレクトリなど) | <ul style="list-style-type: none"> <li>■ <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザーモードでブートされたときに、自動的にファイルシステムをマウントする。</li> <li>■ AutoFS により、ユーザーがディレクトリにアクセスしたとき、あるいはディレクトリから移動したときに、自動的にファイルシステムをマウントまたはマウント解除する。</li> </ul> <p>パフォーマンスを向上させるために、CacheFS ファイルシステムを使用してリモートのファイルシステムをキャッシュに書き込むこともできる。</p> |

ファイルシステムを含むメディアは、必要に応じてメディアをドライブに挿入し、`volcheck` コマンドを実行することによってマウントできます。リムーバブルメディアのマウント方法については、第 1 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

次の表に、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 17-1 ファイルシステムのマウントとマウント解除に使用するコマンド

| コマンド                  | マニュアルページ                   | 説明                                                                                                                    |
|-----------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>mount</code>    | <code>mount (1M)</code>    | ファイルシステムとリモートリソースをマウントする。                                                                                             |
| <code>mountall</code> | <code>mountall (1M)</code> | <code>/etc/vfstab</code> ファイルに指定されているすべてのファイルシステムをマウントする。システムがマルチユーザーモードになるときに、 <code>mountall</code> コマンドが自動的に実行される。 |

表 17-1 ファイルシステムのマウントとマウント解除に使用するコマンド (続き)

| コマンド      | マニュアルページ      | 説明                                             |
|-----------|---------------|------------------------------------------------|
| umount    | mount (1M)    | ファイルシステムとリモートリソースをマウント解除する。                    |
| umountall | mountall (1M) | /etc/vfstab ファイルに指定されているすべてのファイルシステムをマウント解除する。 |

mount と mountall の各コマンドは、不整合が認められた読み取り / 書き込み用のファイルシステムはマウントしません。mount または mountall コマンドからエラーメッセージが表示される場合は、ファイルシステムをチェックする必要があります。ファイルシステムをチェックする方法については、第 20 章を参照してください。

umount と umountall の各コマンドは、使用中のファイルシステムはマウント解除しません。ファイルシステムは、次の場合に使用中であるとみなされます。

- ユーザーがファイルシステム内のファイルまたはディレクトリにアクセスしている場合
- プログラムがそのファイルシステム上にあるファイルをオープンしている場合
- ファイルシステムが共有されている場合

## 汎用マウントオプション

次の表に、mount -o オプションで指定できる汎用オプションを示します。複数のオプションを指定する場合は、コンマ (空白を入れない) で区切ります。たとえば、-o ro,nosuid のようになります。

各ファイルシステムタイプで指定可能なマウントオプションのリストについては、各マウントコマンドのマニュアルページ (mount\_ufs (1M) など) を参照してください。

表 17-2 -o で指定する汎用マウントオプション

| オプション       | ファイルシステム | 説明                                                                                                   |
|-------------|----------|------------------------------------------------------------------------------------------------------|
| bg   fg     | NFS      | 最初のマウントに失敗すると、バックグラウンド (bg) またはフォアグラウンド (fg) で再試行する。このオプションは重要でない vfstab エントリーには安全である。デフォルトは fg。     |
| hard   soft | NFS      | サーバーが応答しない場合の手続きを指定する。soft オプションは、エラーが返されることを示す。hard オプションは、サーバーが応答するまで再試行要求が継続されることを示す。デフォルトは hard。 |

表 17-2 -o で指定する汎用マウントオプション (続き)

| オプション                     | ファイルシステム | 説明                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| intr   nointr             | NFS      | ハードマウントされたファイルシステムに関する応答を待って実行を停止しているプロセスを、キーボード割り込みで強制終了できるかどうかを指定する。デフォルトは intr (割り込み可能)。                                                                                                                                                                                                                                                                     |
| largefiles   nolargefiles | UFS      | 2G バイトを超えるファイルを持つことができるようにする。largefiles オプションでマウントされたファイルシステムは、2G バイトを超えるファイルを格納できる (必須ではない)。nolargefiles オプションを指定した場合、UFS ファイルシステムは Solaris 2.6 (またはその互換バージョン) が動作しているシステムにはマウントできない。デフォルトは largefiles。                                                                                                                                                        |
| logging   nologging       | UFS      | ファイルシステムのロギングを有効または無効にする。UFS ロギングとは、トランザクション (完全な UFS 処理を構成する変更) をログに保存してから、そのトランザクションを UFS ファイルシステムに適用するプロセスである。ロギングを使用すると、UFS ファイルシステムの整合性を保つことができる。つまり、fsck を省略できることを意味する。fsck を省略すると、システムがクラッシュしたとき、あるいはシステムをクリーンな状態でシャットダウンできなかったとき、システムをリブートする時間を省ける。<br><br>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 logging。 |
| atime   noatime           | UFS      | ファイルのアクセス時間更新を抑制する。ただし、最後にファイルの状態が変わった時間または最後にファイルが変更された時間に対する更新が同時に行われる場合を除く。詳細については、stat(2) のマニュアルページを参照。このオプションによって、アクセス時間が重要でないファイルシステム (Usenet ニュースプールなど) でのディスクに対する動作が減る。デフォルトでは、通常アクセス時間 (atime) が記録される。                                                                                                                                                 |
| remount                   | すべて      | 既にマウントされているファイルシステムに関連付けられているマウントオプションを変更する。通常このオプションは、ro オプション以外のオプションと共に使用できる。ただし、変更できるマウントオプションは、ファイルシステムタイプによって異なる。                                                                                                                                                                                                                                         |

表 17-2 -o で指定する汎用マウントオプション (続き)

| オプション           | ファイルシステム                  | 説明                                                                                     |
|-----------------|---------------------------|----------------------------------------------------------------------------------------|
| retry= <i>n</i> | NFS                       | マウント処理に失敗した場合に再試行する。 <i>n</i> は再試行回数。                                                  |
| ro   rw         | CacheFS、NFS、PCFS、UFS、HSFS | 読み取り / 書き込み (rw) または読み取り専用 (ro) を指定する。このオプションを指定しない場合のデフォルトは rw。HSFS のデフォルトのオプションは ro。 |
| suid   nosuid   | CacheFS、HSFS、NFS、UFS      | setuid 実行を許可または禁止する。デフォルトは、setuid 実行を許可する。                                             |

## /etc/vfstab ファイルのフィールドの説明

/etc/vfstab ファイル内のエントリには、次の表に示すように7つのフィールドがあります。

表 17-3 /etc/vfstab ファイルのフィールドの説明

| フィールド名          | 説明                                                                                                                                                                                                                                                                                                                             |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| device to mount | このフィールドは、次のいずれかを指定する。 <ul style="list-style-type: none"> <li>■ 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)</li> <li>■ リモートファイルシステム用のリソース名 (myserver:/export/home など)。NFS の詳細については、『Solaris のシステム管理 (IP サービス)』を参照。</li> <li>■ 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)</li> <li>■ 仮想ファイルシステムタイプ用のディレクトリ</li> </ul> |
| device to fsck  | 「device to mount」フィールドで指定した UFS ファイルシステムに対応する raw (キャラクタ型) デバイス名 (/dev/rdisk/c0t0d0s0 など)。このフィールドによって、fsck コマンドが使用する raw インタフェースが決まる。読み取り専用ファイルシステムやリモートファイルシステムなど、適用できるデバイスがない場合は、ダッシュ (-) を使用する。                                                                                                                             |
| mount point     | デフォルトのマウントポイントディレクトリ (/usr など) を指定する。                                                                                                                                                                                                                                                                                          |
| FS type         | ファイルシステムのタイプを指定する。                                                                                                                                                                                                                                                                                                             |

表 17-3 /etc/vfstab ファイルのフィールドの説明 (続き)

| フィールド名        | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fsck pass     | <p>fsck コマンドがファイルシステムをチェックするか決めるために使用するパス番号。このフィールドでダッシュ (-) を指定すると、ファイルシステムはチェックされない。</p> <p>このフィールドに 0 が指定されている場合、UFS ファイルシステムはチェックされないが、UFS 以外のファイルシステムはチェックされる。このフィールドに 0 より大きい値が指定されている場合、UFS ファイルシステムは常にチェックされる。</p> <p>このフィールドに 1 が指定されている場合、すべてのファイルシステムは vfstab ファイル内の順番どおりに 1 つずつチェックされる。このフィールドに 1 より大きな値が指定され、さらに preen (修復) オプション (-o p) が指定されている UFS ファイルシステムが複数ある場合、効率を最大限に高めるために、fsck コマンドは複数のディスク上のファイルシステムを自動的に並行してチェックする。それ以外の場合、このフィールドの値は意味を持たない。</p> |
| mount at boot | <p>システムのブート時にファイルシステムが mountall コマンドによって自動的にマウントされるかどうかを yes または no で設定する。このフィールドは AutoFS とは連動していないので注意すること。ルート (/)、/usr、/var のファイルシステムは最初は vfstab ファイルからマウントされない。これらのファイルシステムおよび /proc や /dev/fd などのような仮想ファイルシステムの場合、このフィールドは常に no に設定しなければならない。</p>                                                                                                                                                                                                                      |
| mount options | <p>ファイルシステムのマウントに使用されるオプションを (空白を入れずに) コンマで区切ったリスト。オプションなしを示すにはダッシュ (-) を使用する。汎用マウントオプションについては、表 17-2 を参照。</p>                                                                                                                                                                                                                                                                                                                                                             |

注 - /etc/vfstab ファイル内の各フィールドには必ずエントリが必要です。フィールドに値を指定しない場合は、必ずダッシュ (-) を入力してください。ダッシュを入力しないと、システムが正常にブートしない可能性があります。同様に、フィールドの値に空白文字を使用しないでください。

## ファイルシステムのマウント

次の節では、/etc/vfstab ファイルにエントリを追加するか、コマンド行から mount コマンドを使用してファイルシステムをマウントする方法について説明します。



## 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すでにマウント済みであるかを調べるには、`mount` 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 17-1 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

この例は、`mount` コマンドを使用して、現在マウントされているファイルシステムに関する情報を表示する方法を示しています。

```
$ mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Jan  8 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Jan  8 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Jan  8 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Jan  8 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Jan  8 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Jan  8 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
$
```

## ▼ /etc/vfstab ファイルにエントリを追加する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. `/etc/vfstab` ファイルを編集してエントリを追加します。次のことを確認してください。
  - a. 各フィールドを空白 (空白文字またはタブ) で区切る。
  - b. 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。
  - c. 変更結果を保存する。

/etc/vfstab フィールドのエントリについては、表 17-3 を参照してください。

---

注 - ルート (/) ファイルシステムは、ブートプロセスの過程でカーネルによって読み取り専用としてマウントされます。そのため、remount オプション (および、remount と一緒に使用できるオプション) だけが /etc/vfstab ファイルのルート (/) エントリでは有効です。

---

### 例 17-2 /etc/vfstab ファイルにエントリを追加する

次の例は、ディスクスライス /dev/dsk/c0t3d0s7 を UFS ファイルシステムとして、マウントポイントディレクトリ /files1 にマウントする方法を示しています。「device to fsck」として raw キャラクタ型デバイス /dev/rdisk/c0t3d0s7 を指定します。「fsck pass」の値が 2 なので、ファイルシステムは順不同でチェックされます。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
#
/dev/dsk/c0t3d0s7 /dev/rdisk/c0t3d0s7 /files1 ufs      2     yes    -
```

次の例は、システム pluto 上のディレクトリ /export/man を、NFS ファイルシステムとしてマウントポイント /usr/man にマウントする方法を示しています。ファイルシステムが NFS であるため、「device to fsck」や「fsck pass」は指定されません。この例では、「mount options」は ro (読み取り専用) と soft になっています。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
pluto:/export/man -                /usr/man nfs      -     yes    ro,soft
```

次の例は、ルート (/) ファイルシステムをループバックマウントポイント /tmp/newroot にマウントする方法を示しています。LOFS ファイルシステムをマウントするときは、LOFS ファイルシステム内に入るファイルシステムを先にマウントし、その後で LOFS をマウントします。

```
#device          device          mount   FS      fsck  mount  mount
#to mount        to fsck         point   type    pass  at boot options
#
/                 -                /tmp/newroot lofs -     yes    -
```

## ▼ 1つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. **/etc/vfstab** ファイル内に列挙されているファイルシステムをマウントします。

```
# mount /mount-point
/mount-point は、/etc/vfstab ファイル内の「mount point」または「device
to mount」フィールドのエントリを指定します。通常は、マウントポイントを指
定する方が簡単です。
```

### 例 17-3 1つのファイルシステムをマウントする (/etc/vfstab ファイル)

次の例は、/etc/vfstab ファイル内に列挙されているファイルシステム  
/usr/dist をマウントする方法を示しています。

```
# mount /usr/dist
```

### 例 17-4 すべてのファイルシステムをマウントする (/etc/vfstab ファイル)

次の例は、mountall コマンドを実行したときに、すでにファイルシステムがマウン  
トされている場合に表示されるメッセージを示します。

```
# mountall
/dev/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
```

マウントが実行される前に、「device to fsck」エントリがあるすべてのファイル  
システムがチェックされ、必要であれば修正されます。

次の例は、/etc/vfstab ファイル内に列挙されているすべてのローカルシステムを  
マウントする方法を示しています。

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Jan 8 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Jan 8 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Jan 8 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Jan 8 09:38:30 2004
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Jan 8 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Jan 8 09:38:30 2004
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
```

次の例は、`/etc/vfstab` ファイル内に列挙されているすべてのリモートファイルシステムをマウントする方法を示しています。

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Jan 8 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Jan 8 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Jan 8 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Jan 8 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Jan 8 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Jan 8 09:38:30 2004
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
```

## ▼ UFS ファイルシステムのマウント方法 (mount コマンド)

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. UFS ファイルシステムをマウントします。

```
# mount [-o mount-options] /dev/dsk/device-name /mount-point
```

`-o mount-options` UFS ファイルシステムのマウントに使用できるマウントオプションを指定する。オプションの詳細については、表 17-2 または `mount_ufs(1M)` のマニュアルページを参照。

`/dev/dsk/device-name` ファイルシステムが含まれているディスクスライスのディスクデバイス名 (`/dev/dsk/c0t3d0s7` など) を指定する。ディスクのスライス情報については、189 ページの「ディスクスライス情報を表示する方法」を参照。

`/mount-point` 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 17-5 UFS ファイルシステムをマウントする (mount コマンド)

次の例は、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントする方法を示しています。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

## ▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)

ファイルシステムをマウントするときには、`largefiles` オプションがデフォルトで選択されるため、2G バイトを超えるファイルを作成できます。大規模ファイルを作成した後で、`nolargefiles` オプションを指定してファイルシステムを再マウントするには、あるいは Solaris 2.6 およびその互換バージョンを実行するシステム上にマウントするには、大規模ファイルをすべて削除し、`fsck` コマンドを実行して状態を「`nolargefiles`」にリセットしなければなりません。

以下の手順では、ファイルシステム用のエントリが `/etc/vfstab` ファイルにあるものとします。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。
  3. ファイルシステム内に大規模ファイルが存在しないことを確認します。

```
# cd /mount-point
# find . -xdev -size +20000000 -exec ls -l {} \;
```

`/mount-point` は、大規模ファイルがあるかどうかをチェックするファイルシステムのマウントポイントを指定します。
  4. 大規模ファイルが当該ファイルシステム内に存在する場合は、必要に応じてそのファイルを削除するか、他のファイルシステムに移動します。
  5. ファイルシステムをマウント解除します。

```
# umount /mount-point
```
  6. ファイルシステムの状態をリセットします。

```
# fsck /mount-point
```
  7. `nolargefiles` オプションを指定してファイルシステムを再マウントします。

```
# mount -o nolargefiles /mount-point
```

### 例 17-6 大規模ファイルを持たないファイルシステムをマウントする (mount コマンド)

次の例は、/datab ファイルシステムをチェックし、nolargefiles オプションを指定して再マウントする方法を示しています。

```
# cd /datab
# find . -xdev -size +20000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

## ▼ NFS ファイルシステムのマウント方法 (mount コマンド)

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. リソース (ファイルまたはディレクトリ) がサーバーから使用可能かどうかを確認します。

NFS ファイルシステムをマウントするには、share コマンドを使用し、サーバー上のリソースを使用可能にしておかなければなりません。リソースの共有方法については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「NFS サービスについて」を参照してください。

4. NFS ファイルシステムをマウントします。

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

*-o mount-options* NFS ファイルシステムのマウントに使用できるマウントオプションを指定する。mount の汎用オプションのリストについては、表 17-2 を参照。完全なオプションのリストについては、mount\_nfs (1M) のマニュアルページを参照。

*server:/directory* 共有するリソースを持つサーバーのホスト名と、マウントするファイルまたはディレクトリへのパスを指定する。

*/mount-point* 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 17-7 NFS ファイルシステムをマウントする (mount コマンド)

次の例は、サーバー `pluto` の `/export/packages` ディレクトリを `/mnt` にマウントする方法を示しています。

```
# mount -F nfs pluto:/export/packages /mnt
```

### ▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)

次の手順で、PCFS (DOS) ファイルシステムをハードディスクからマウントします。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. PCFS ファイルシステムをマウントします。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
```

`-o rw | ro` PCFS ファイルシステムを読み取り / 書き込み (rw) または読み取り専用 (ro) にマウントできることを指定する。このオプションを指定しない場合のデフォルトは `rw`

`/dev/dsk/device-name` ディスク全体のデバイス名を指定する (`/dev/dsk/c0t0d0p0` など)

`logical-drive` DOS の論理ドライブ名 (c から z) またはドライブ番号 (1 から 24) を指定する。ドライブ c はドライブ 1 に相当し、ディスク上の基本 DOS スライスを表す。他のすべてのドライブ名やドライブ番号は、拡張 DOS スライス内の DOS 論理ドライブを表す

`/mount-point` ファイルシステムをマウントするディレクトリを指定する

「`device-name`」と「`logical-drive`」は、コロンで区切る必要があります。

**例 17-8** x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする (mount コマンド)

次の例は、基本 DOS スライス内の論理ドライブを /pcfs/c ディレクトリにマウントする方法を示しています。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例は、拡張 DOS スライス内の最初の論理ドライブを /mnt に読み取り専用としてマウントする方法を示しています。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

---

## ファイルシステムのマウント解除

ファイルシステムをマウント解除すると、ファイルシステムがそのマウントポイントから削除され、そのエントリが /etc/mnttab ファイルから削除されます。マウントされているファイルシステム上では、一部のファイルシステム管理作業を実行できません。次の場合には、ファイルシステムをマウント解除する必要があります。

- ファイルシステムが不要になった場合、または新しいソフトウェアが入ったファイルシステムに交換された場合
- fsck コマンドを使用してファイルシステムを検査し、修復する必要がある場合  
fsck コマンドの詳細については、[第 20 章](#)を参照してください。

ファイルシステムの完全バックアップを実行する前に、マウント解除しておくといでしょう。バックアップの実行の詳細については、[第 23 章](#)を参照してください。

---

注 - 各ファイルシステムは、ファイルシステムのシャットダウン手順の一部として自動的にマウント解除されます。

---

umount -f オプションを使用すると、非常時に使用中のファイルシステムを強制的にマウント解除できます。ファイルを開いた状態でファイルシステムをマウント解除すると、データが失われる可能性があるため、非常時以外はこの操作を行わないようにしてください。このオプションは、UFS と NFS のファイルシステムでのみ使用できます。



## ファイルシステムをマウント解除する場合の前提条件

ファイルシステムをマウント解除する場合の前提条件は次のとおりです。

- スーパーユーザーになるか、同等の役割を引き受けていなければならない。
- ファイルシステムは、マウント解除が可能な状態でなければならない。使用中のファイルシステムはマウント解除できません。ユーザーがそのディレクトリ内にいるとき、プログラムがそのファイルシステム上のファイルを開いているとき、または共有されているときには、ファイルシステムは使用中とみなされます。次の方法でファイルシステムをマウント解除が可能な状態にできます。
  - 別のファイルシステム内のディレクトリにカレントディレクトリを変更する。
  - システムからログアウトする。
  - `fuser` コマンドを使用して、そのファイルシステムを使用中のすべてのプロセスを表示し、必要に応じて終了させる。詳細については、[297 ページの「ファイルシステムを使用中のすべてのプロセスを終了させる方法」](#)を参照。  
他のユーザーが使用しているファイルシステムをマウント解除する必要があるときは、各ユーザーに通知してください。
  - ファイルシステムの共有を解除する。ファイルシステムの共有を解除する方法については、`unshare(1M)` のマニュアルページを参照してください。

## ファイルシステムのマウント解除を確認する方法

ファイルシステムをマウント解除したことを確認するには、`mount` コマンドからの出力を調べます。

```
$ mount | grep unmounted-file-system
$
```

### ▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. どのプロセスを終了させるかがわかるように、ファイルシステムを使用中のすべてのプロセスを表示します。

```
# fuser -c [ -u ] /mount-point
```

-c           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。

-u           プロセス ID ごとにユーザーのログイン名が表示される。

/mount-point   プロセスを終了させるファイルシステムの名前を指定する。

3. ファイルシステムを使用しているすべてのプロセスを終了させます。

```
# fuser -c -k /mount-point
ファイルシステムを使用している各プロセスに SIGKILL が送信されます。
```

---

注 - ユーザーのプロセスを終了させるときには、必ず事前に警告してください。

---

4. ファイルシステムを使用しているプロセスがないことを確認します。

```
# fuser -c /mount-point
```

#### 例 17-9 あるファイルシステムを使用中のすべてのプロセスを終了させる

次の例は、/export/home ファイルシステムを使用中のプロセス 4006c を終了させる方法を示しています。

```
# fuser -c /export/home
/export/home:       4006c
# fuser -c -k /export/home
/export/home:       4006c
# fuser -c /export/home
/export/home:
```

## ▼ 1つのファイルシステムをマウント解除する方法

次の手順に従って、ファイルシステム (ルート (/)、/usr、/var を除く) をマウント解除します。

---

注 - ルート(/)、/usr、/var の各ファイルシステムは、システムが機能するのに必要であるため、シャットダウン中でなければマウント解除できません。

---

- 手順
1. [297 ページ](#)の「ファイルシステムをマウント解除する場合の前提条件」の前提条件を満たしているかどうかを確認します。
  2. ファイルシステムをマウント解除します。

```
# umount /mount-point
```

`/mount-point` は、マウント解除するファイルシステムの名前を示します。次のいずれかを指定できます。

- ファイルシステムがマウントされているディレクトリ名
- ファイルシステムのデバイス名パス
- NFS ファイルシステムのリソース
- LOFS ファイルシステムのループバックディレクトリ

#### 例 17-10 1 つのファイルシステムをマウント解除する

次の例は、ローカルのホームディレクトリからファイルシステムをマウント解除する方法を示しています。

```
# umount /export/home
```

次の例は、ローカルディスクの 7 番目のスライス上のファイルシステムをマウント解除する方法を示しています。

```
# umount /dev/dsk/c0t0d0s7
```

次の例は、`/export` ファイルシステムを強制的にマウント解除する方法を示しています。

```
# umount -f /export  
#
```

次の例は、`/etc/vfstab` ファイル内のすべてのファイルシステム (ルート (`/`)、`/proc`、`/var`、`/usr` を除く) をマウント解除する方法を示しています。

```
# umountall
```

使用中のファイルシステムを除く、すべてのファイルシステムがマウント解除されます。



## 第 18 章

# CacheFS ファイルシステムの使用 (手順)

この章では、CacheFS™ ファイルシステムの設定および管理の方法について説明します。

この章で説明する作業マップは、次のとおりです。

- 301 ページの「CacheFS ファイルシステムの概観 (作業マップ)」
- 304 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」
- 310 ページの「CacheFS ファイルシステムの保守 (作業マップ)」
- 315 ページの「CacheFS ファイルシステムのバックアップ (作業マップ)」
- 325 ページの「CacheFS の統計情報の収集 (作業マップ)」

CacheFS に関するエラーの障害追跡については、321 ページの「cachefspack エラーの障害追跡」を参照してください。

## CacheFS ファイルシステムの概観 (作業マップ)

次の作業マップは、CacheFS ファイルシステムを使用するためのすべての作業を確認するために使用してください。このマップに記載された各作業は、CacheFS ファイルシステムの作成とマウント、キャッシュのバックアップと保守などの一連の付加された作業を指しています。

| 作業                             | 説明                                 | 参照先                                        |
|--------------------------------|------------------------------------|--------------------------------------------|
| 1. CacheFS ファイルシステムを作成してマウントする | キャッシュを作成し、ファイルシステムをそのキャッシュにマウントする。 | 304 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」 |

| 作業                                       | 説明                                                                                  | 参照先                                      |
|------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------|
| 2. CacheFS ファイルシステムを保守する                 | キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムを表示および変更する。                               | 310 ページの「CacheFS ファイルシステムの保守 (作業マップ)」    |
| 3. (省略可能) CacheFS ファイルシステムをバックおよびバック解除する | キャッシュをバックしてバックングリストを使用するかどうかを決める。キャッシュをバックすると、キャッシュ内の特定のファイルおよびディレクトリが常に更新されるようになる。 | 315 ページの「CacheFS ファイルシステムのバックング (作業マップ)」 |
| 4. CacheFS の統計情報を収集する                    | キャッシュの性能や適切なキャッシュサイズを決める。                                                           | 325 ページの「CacheFS の統計情報の収集 (作業マップ)」       |

## CacheFS ファイルシステムの概要

CacheFS ファイルシステムは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティを改善する汎用ファイルシステム キャッシュメカニズムです。CacheFS ファイルシステムは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ファイルシステムはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。

## CacheFS ファイルシステムの機能

CacheFS ファイルシステムをクライアントシステム上に作成すると、キャッシュに書き込んだファイルシステムをクライアントがネットワークを経由しなくても、ローカルにアクセスできます。次の図は、CacheFS ファイルシステムの使用に関連する構成要素の関係を示しています。

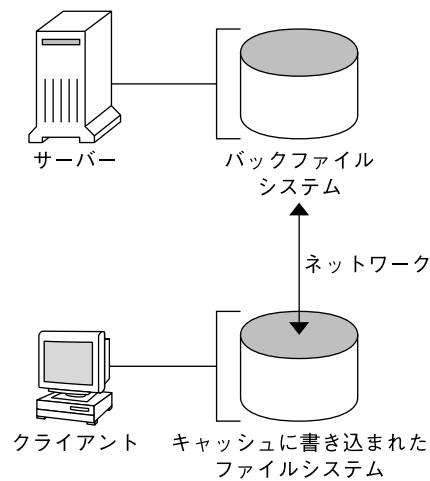


図 18-1 CacheFS ファイルシステムの機能

「バック」ファイルシステムとは、キャッシュにマウントされるように指定したファイルシステムです。通常、これは NFS または HSFS (High Sierra File System) ファイルシステムです。ユーザーがバックファイルシステムの一部であるファイルにアクセスしようとする、そのファイルはキャッシュに書き込まれます。「フロント」ファイルシステムとは、キャッシュにマウントされ、ローカルのマウントポイントからアクセスされるファイルシステムです。フロントファイルシステムのタイプは、UFS でなければなりません。

ユーザーにとって、CacheFS ファイルシステムのファイルに初めてアクセスするときは低速に思われますが、同じファイルを続けて使用していると高速になります。

## CacheFS ファイルシステムの構造と動作

各キャッシュには、キャッシュの構造とその動作を決定する 1 組のパラメータが付いています。各パラメータは、以下の表に示すデフォルト値に設定されています。デフォルト値は、フロントファイルシステム全体をキャッシュに使用するように指定しますが、これはファイルシステムをキャッシュに書き込む場合の推奨方法です。

表 18-1 CacheFS ファイルシステムのパラメータとそのデフォルト値

| CacheFS ファイルシステムのパラメータ | デフォルト値 | 定義                                               |
|------------------------|--------|--------------------------------------------------|
| maxblocks              | 90%    | CacheFS ファイルシステムがフロントファイルシステム内で要求できる最大ブロック数を設定する |
| minblocks              | 0%     | CacheFS ファイルシステムがフロントファイルシステム内で要求できる最小ブロック数を設定する |

表 18-1 CacheFS ファイルシステムのパラメータとそのデフォルト値 (続き)

| CacheFS ファイルシステムのパラメータ | デフォルト値 | 定義                                                                                       |
|------------------------|--------|------------------------------------------------------------------------------------------|
| threshblocks           | 85%    | CacheFS ファイルシステムが minblocks で指定したより多数のブロックを要求する前に、フロントファイルシステム内で使用可能でなければならないブロック数を設定する  |
| maxfiles               | 90%    | CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最大数 (ファイル数) を設定する                        |
| minfiles               | 0%     | CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最小数を設定する                                 |
| threshfiles            | 85%    | CacheFS ファイルシステムが minfiles で指定したより多くのファイルを要求する前に、フロントファイルシステム内で使用可能でなければならない i ノード数を設定する |

通常、これらのパラメータ値を変更する必要はありません。最適のキャッシュ動作が得られるデフォルト値に設定されています。ただし、キャッシュに使用されないフロントファイルシステム内に空き空間があり、それを他のファイルシステムに使用する場合は、maxblocks と maxfiles の値を変更することをお勧めします。この変更を行うには、cfsadmin コマンドを使用します。たとえば、次のようになります。

```
$ cfsadmin -o maxblocks=60
```

## CacheFS ファイルシステムの作成とマウント (作業マップ)

次の表の手順を使用して、CacheFS ファイルシステムを作成およびマウントしてください。

| 作業                         | 説明                                 | 参照先                    |
|----------------------------|------------------------------------|------------------------|
| 1. キャッシュに書き込むファイルシステムを共有する | キャッシュに書き込むファイルシステムが共有されていることを確認する。 | share (1M) のマニュアルページ   |
| 2. キャッシュを作成する              | cfsadmin コマンドを使用してキャッシュを作成する。      | 305 ページの「キャッシュを作成する方法」 |



| 作業                       | 説明                                            | 参照先                                             |
|--------------------------|-----------------------------------------------|-------------------------------------------------|
| 3. ファイルシステムをキャッシュにマウントする | 次のいずれかの方法を使用して、ファイルシステムをキャッシュにマウントする。         |                                                 |
|                          | mount コマンドで、CacheFS ファイルシステムをマウントする           | 306 ページの「CacheFS ファイルシステムのマウント方法 (mount)」       |
|                          | /etc/vfstab ファイルを編集して、CacheFS ファイルシステムをマウントする | 308 ページの「CacheFS ファイルシステムのマウント方法 (/etc/vfstab)」 |
|                          | AutoFS を使用して、キャッシュされたファイルシステムをマウントする          | 309 ページの「CacheFS ファイルシステムのマウント方法 (AutoFS)」      |

## ▼ キャッシュを作成する方法

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. キャッシュを作成します。

```
# cfsadmin -c /cache-directory
```

*cache-directory* は、キャッシュがあるディレクトリの名前を示します。

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

---

注- キャッシュを作成し終わったら、キャッシュディレクトリ内で処理を実行しないでください。何か実行すると、CacheFS ソフトウェア内で矛盾が生じる可能性があります。

---

### 例 18-1 キャッシュを作成する

次の例は、デフォルトのキャッシュパラメータ値を使用して、*/local/mycache* ディレクトリ内にキャッシュを作成する方法を示しています。

```
# mkdir /local
# cfsadmin -c /local/mycache
```

## ファイルシステムをキャッシュにマウントする

キャッシュにマウントされるファイルシステムを指定して、ユーザーがそのファイルシステム内のファイルにローカルにアクセスできるようにします。実際には、各ファイルは、ユーザーがアクセスするまではキャッシュに書き込まれません。

次の表に、CacheFS ファイルシステムをマウントする 3 つの方法を示します。

| CacheFS ファイルシステムのマウント方法 | CacheFS マウント方法の使用頻度                                 |
|-------------------------|-----------------------------------------------------|
| mount コマンドを使用する方法       | 同じファイルにアクセスするには、システムをリブートするたびに mount コマンドを使用する      |
| /etc/vfstab ファイルを編集する方法 | 1 度だけですむ。/etc/vfstab ファイルの内容は、システムをリブートした後も変更されずに残る |
| AutoFS を使用する方法          | 1 度だけですむ。AutoFS のマップは、システムをリブートした後も変更されずに残る         |

いずれかの方法を選択してファイルシステムをマウントしてください。

共有可能なファイルシステムしかマウントできません。ファイルシステムの共有については、share (1M) のマニュアルページを参照してください。

---

注 - CacheFS ファイルシステムでは、ルート (/) と /usr のファイルシステムをキャッシュに書き込むことはできません。

---

## ▼ CacheFS ファイルシステムのマウント方法 (mount)

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. 必要に応じて、マウントポイントを作成します。

```
# mkdir /mount-point
```

マウントポイントはどこからでも作成できますが、UFS ファイルシステムでなければなりません。次の手順のように、mount コマンドで CacheFS オプションを使用すると、作成するマウントポイントが指定したキャッシュディレクトリ内のキャッシュに書き込まれるように指定できます。

3. ファイルシステムをキャッシュにマウントします。

```
# mount -F cachefs -o backfstype=fstype,cachedir=/cache-directory [,options]
/back-filesystem /mount-point
```

*fstype*                   バックファイルシステムのファイルシステムタイプ (NFS または HSFS)

*/cache-directory*       キャッシュがある UFS ディレクトリの名前。これは、305 ページの「[キャッシュを作成する方法](#)」でキャッシュを作成するときの指定と同じ

|                         |                                                                                                                                                                       |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>options</i>          | ファイルシステムをキャッシュにマウントするときに追加できる他のマウントオプション。CacheFS mount オプションの詳細については、 <code>mount_cacheofs(1M)</code> のマニュアルページを参照                                                    |
| <i>/back-filesystem</i> | キャッシュにマウントするバックファイルシステムのマウントポイント。バックファイルシステムが NFS ファイルシステムである場合は、ファイルシステムのマウント元となるサーバーのホスト名と、キャッシュにマウントするファイルシステム名 (コロンで区切る) を指定する必要がある。たとえば、 <i>merlin:/data/abc</i> |
| <i>/mount-point</i>     | ファイルシステムのマウント先となるディレクトリ                                                                                                                                               |

4. 作成したキャッシュが実際にマウントされたかどうかを確認します。

```
# cachefsstat /mount-point
/mount-point は作成した CacheFS ファイルシステムです。
たとえば、次のようになります。

# cachefsstat /docs
/docs
      cache hit rate:    100% (0 hits, 0 misses)
      consistency checks: 1 (1 pass, 0 fail)
      modifies:         0
      garbage collection: 0
```

ファイルシステムがキャッシュにマウントされなかった場合は、次のようなエラーメッセージが表示されます。

```
# cachefsstat /mount-point
cachefsstat: mount-point: not a cacheofs mountpoint

cacheofsstat コマンドの詳細については、325 ページの「CacheFS の統計情報の収集」を参照してください。
```

## 例 18-2 CacheFS ファイルシステムをマウントする (mount)

次の例は、NFS ファイルシステム `merlin:/docs` を `/docs` という CacheFS ファイルシステムとして、`/local/mycache` というキャッシュにマウントする方法を示しています。

```
# mkdir /docs
# mount -F cacheofs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

次の例は、Solaris 9 SPARC CD (HSFS ファイルシステム) を `/cfssrc` という CacheFS ファイルシステムとして使用できるようにする方法を示しています。CD には書き込めないため、引数 `ro` を指定して CacheFS ファイルシステムを読み取り専用にします。この例では、`vo1d` を実行していないものとします。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /sol9
# mount -F cacheofs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/sol9 /dev/dsk/c0t6d0s0 /cfssrc
# ls /cfssrc
```

次の例は、`vold` を実行しながら、Solaris 9 SPARC CD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,  
backpath=/cdrom/sol_9_sparc/s0 /vol/dev/dsk/c0t2d0/sol_9_sparc/s0 /cfssrc
```

次の例は、`vold` を実行しながら、CD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,  
backpath=/cdrom/epson /vol/dev/dsk/c0t2d0/epson /drvrs
```

次の例では、`demandconst` オプションを使用して、NFS CacheFS ファイルシステム `/docs` の整合性チェックを指定します。`/docs` のバックファイルシステムは `merlin:/docs` です。詳細については、312 ページの「CacheFS ファイルシステムの整合性チェック」を参照してください。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

## ▼ CacheFS ファイルシステムのマウント方法 (`/etc/vfstab`)

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. エディタを使用して、マウントするファイルシステムを `/etc/vfstab` ファイル内で指定します。  
下の例を参照してください。  
`/etc/vfstab` ファイルの詳細については、287 ページの「`/etc/vfstab` ファイルのフィールドの説明」を参照してください。
  3. CacheFS ファイルシステムをマウントします。  

```
# mount /mount-point
```

あるいはシステムをリブートします。

### 例 18-3 CacheFS ファイルシステムをマウントする (`/etc/vfstab`)

次の例は、キャッシュされたディレクトリ `/opt/cache` にマウントされるリモートシステム `starbug` からの `/data/abc` ディレクトリ用の `/etc/vfstab` エントリを示しています。

```
#device          device          mount    FS    fsck  mount  mount  
#to mount        to fsck         point    type  pass  at boot options  
#
```

```
starbug:/data/abc /local/abc /opt/cache cachefs 7 yes local-access,bg,  
nosuid,demandconst,backfstype=nfs,cachedir=/opt/cache
```

## ▼ CacheFS ファイルシステムのマウント方法 (AutoFS)

自動マウントマップ内で `-fstype=cachefs` マウントオプションを指定して、AutoFS によってファイルシステムをキャッシュにマウントします。CacheFS のマウントオプション (`backfstype` や `cachedir` など) も、自動マウントマップ内で指定します。

自動マウントマップの詳細については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「`autofs` 管理作業の概要」または `automount (1M)` のマニュアルページを参照してください。

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. エディタを使用して、次の 1 行を `auto_direct` マップに追加します。

```
/mount-point -fstype=cachefs,cachedir=/directory,backfstype=nfs  
server:/file-system
```
  3. エディタを使用して、次の 1 行を `auto_master` マップに追加します。

```
/-  
/- エントリによって、auto_direct マップをチェックするように指示します。
```
  4. システムをリブートします。
  5. エントリが正しく作成されたか確認するには、次のようにキャッシュにマウントしたファイルシステムにカレントディレクトリを変更して内容を表示します。

```
# cd /filesystem  
# ls
```

### 例 18-4 CacheFS ファイルシステムをマウントする (AutoFS)

次の `auto_direct` エントリは、CacheFS ファイルシステムを `/docs` ディレクトリに自動的にマウントします。

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

---

## CacheFS ファイルシステムの保守 (作業マップ)

CacheFS ファイルシステムを設定し終わったら、保守作業を多少行う必要があります。CacheFS ファイルシステムの保守作業を行う必要がある場合は、次の表の手順(オプション)に従ってください。

| 作業                         | 説明                                                                               | 参照先                                     |
|----------------------------|----------------------------------------------------------------------------------|-----------------------------------------|
| 1. CacheFS ファイルシステムの変更     | キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムの動作を変更する。                              | 311 ページの「CacheFS ファイルシステムの変更」           |
| 2. CacheFS ファイルシステム情報の表示   | <code>cfsadmin</code> コマンドを使用して、CacheFS ファイルシステムに関する情報を表示する。                     | 311 ページの「CacheFS ファイルシステムに関する情報を表示する方法」 |
| 3. 整合性チェックの実行              | <code>cfsadmin</code> コマンドを使用して必要に応じて整合性チェックを実行する。                               | 312 ページの「必要に応じてキャッシュの整合性チェックを指定する方法」    |
| 4. CacheFS ファイルシステムの削除     | <code>umount</code> コマンドと <code>cfsadmin</code> コマンドを使用して、CacheFS ファイルシステムを削除する。 | 313 ページの「CacheFS ファイルシステムを削除する方法」       |
| 5. CacheFS ファイルシステムの整合性の検査 | <code>fsck_cacheofs</code> コマンドを使用して、CacheFS ファイルシステムの整合性を検査する。                  | 314 ページの「CacheFS ファイルシステムの整合性をチェックする方法」 |

---

## CacheFS ファイルシステムの保守

この節では、CacheFS ファイルシステムの保守方法について説明します。

`/etc/vfstab` ファイルを使用してファイルシステムをマウントしている場合は、このファイル内でファイルシステムオプションを編集してキャッシュを変更します。AutoFS を使用している場合は、AutoFS マップ内でファイルシステムオプションを編集してキャッシュを変更します。

## CacheFS ファイルシステムの変更

キャッシュ内でファイルシステムを変更する場合は、キャッシュを削除してから作成し直す必要があります。また、ファイルシステムの共有方法とアクセス方法によっては、マシンをシングルユーザーモードでリブートしなければならない場合があります。

次の例では、キャッシュが削除されてから再び作成され、ファイルシステム `/docs` に指定された `demandconst` オプションを使用して再びマウントされます。

```
# shutdown -g30 -y
.
.
.
Type Cntrl-d to proceed with normal startup,
(or give root password for system maintenance):
# enter password:
.
.
.
(ここで、キャッシュがあるファイルシステムで
fsck コマンドを実行するようメッセージが表示される。)

# fsck /local
# mount /local
# cfsadmin -d all /local/mycache
# cfsadmin -c /local/mycache
# init 6
.
.
.
console login:
password:
# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1,demandconst
merlin:/docs /docs
#
```

### ▼ CacheFS ファイルシステムに関する情報を表示する方法

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. 指定したキャッシュにマウントされたすべてのファイルシステムに関する情報を表示します。

```
# cfsadmin -l /cache-directory
/cache-directory は、キャッシュがあるディレクトリの名前です。
```

## 例 18-5 CacheFS ファイルシステムに関する情報を表示する

次の例は、`/local/mycache` キャッシュディレクトリに関する情報を示しています。この例では、`/docs` ファイルシステムが `/local/mycache` にキャッシュされます。最終行には、CacheFS ファイルシステムの名前が表示されます。

```
# cfsadmin -l /local/mycache
cfsadmin: list cache FS information
  maxblocks      90%
  minblocks      0%
  threshblocks   85%
  maxfiles       90%
  minfiles       0%
  threshfiles    85%
  maxfilesize    3MB
merlin: _docs: _docs
#
```

## CacheFS ファイルシステムの整合性チェック

CacheFS ソフトウェアでは、キャッシュされたディレクトリとファイルが最新の状態に保たれることを保証するために、キャッシュに格納されているファイルの整合性を定期的にチェックします。整合性をチェックするために、CacheFS ソフトウェアでは現在の変更時刻を前回の変更時刻と比較します。変更時刻が異なる場合は、そのディレクトリまたはファイルに関するすべてのデータと属性がキャッシュから消去されます。そして、バックファイルシステムから新しいデータと属性が取り出されます。

### 必要に応じて行う整合性チェック

整合性チェックは、`-o demandconst` オプションを使用してマウントされたファイルシステムに対して明示的に要求した場合のみ実行できます。このオプションでファイルシステムをキャッシュにマウントした場合は、`-s` オプションを指定した `cfsadmin` コマンドを実行して整合性チェックを要求します。デフォルトでは、ファイルがアクセスされるたびに、各ファイルの整合性がチェックされます。ファイルがアクセスされなければ、チェックは実行されません。`-o demandconst` オプションを使用すると、整合性チェックによってネットワークがいっぱいになるという事態を回避できます。

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

## ▼ 必要に応じてキャッシュの整合性チェックを指定する方法

手順 1. クライアントシステムでスーパーユーザーになります。



2. ファイルシステムをキャッシュにマウントし、キャッシュの整合性チェックを指定します。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/directory,demandconst
server:/file-system /mount-point
```

3. 特定の CacheFS ファイルシステムに対する整合性チェックを開始します。

```
# cfsadmin -s /mount-point
```

## ▼ CacheFS ファイルシステムを削除する方法

- 手順 1. クライアントシステムでスーパーユーザーになります。

2. CacheFS ファイルシステムをマウント解除します。

```
# umount /mount-point
```

*/mount-point* は、削除する CacheFS ファイルシステムを示します。

3. CacheFS ファイルシステムの名前 (キャッシュ ID) を判別します。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
maxblocks      90%
minblocks      0%
threshblocks   85%
maxfiles       90%
minfiles       0%
threshfiles    85%
maxfilesize    3MB
cache-ID
#
```

4. 指定したキャッシュから CacheFS ファイルシステムを削除します。

```
# cfsadmin -d cache-ID /cache-directory
```

*cache-ID* CacheFS ファイルシステム名。この名前は、`cfsadmin -l` コマンドからの出力の最終行に表示される。詳細については、[311 ページの「CacheFS ファイルシステムに関する情報を表示する方法」](#)を参照。*cache-ID* に `all` を指定すると、特定のキャッシュに書き込まれた CacheFS ファイルシステムをすべて削除できる。

*/cache-directory* キャッシュがあるディレクトリ。

5. ファイルシステムが削除されたことを確認します。

直前に削除したファイルシステムのキャッシュ ID が、`cfsadmin -l` の出力に表示されなくなります。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
```

```
maxblocks    90%
minblocks    0%
threshblocks 85%
maxfiles     90%
minfiles     0%
threshfiles  85%
maxfilesize  3MB
```

#

このコマンド出力に指定されるフィールドについては、`cfsadmin(1M)` のマニュアルページを参照してください。

6. **fsck -F cachefs** コマンドを実行して、キャッシュのリソースアカウントを更新します。  
詳細については、314 ページの「CacheFS ファイルシステムの整合性をチェックする方法」を参照してください。

### 例 18-6 CacheFS ファイルシステムを削除する

次の例は、キャッシュからファイルシステムを削除する方法を示しています。

```
# umount /cfssrc
# cfsadmin -l /cfssrc
# cfsadmin -d _dev_dsk_c0t6d0s0:_cfssrc
# cfsadmin -l
```

## ▼ CacheFS ファイルシステムの整合性をチェックする方法

`fsck` コマンドを使用して、CacheFS ファイルシステムの整合性を確認します。何も操作しなくても、`fsck` コマンドの CacheFS バージョンによって問題が自動的に解決されます。`fsck` コマンドはブート時またはファイルシステムのマウント時に自動的に実行されるため、CacheFS ファイルシステムに対して `fsck` コマンドを手作業で実行する必要はありません。整合性を手作業で確認する場合は、次の手順を使用できます。

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

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. 指定されたキャッシュ内でファイルシステムをチェックします。

```
# fsck -F cachefs [-m -o noclean] /cache-directory
```

-m                    `fsck` コマンドに CacheFS ファイルシステムをチェックさせるが、修復は行わない。

- o noclean      CacheFS ファイルシステムに対するチェックのみを実行させる。修復は行わない。
- /cache-directory      キャッシュがあるディレクトリの名前を指定する。

### 例 18-7 CacheFS ファイルシステムの整合性をチェックする

次の例は、/local/mycache キャッシュに書き込まれているファイルシステムをチェックする方法を示しています。

```
# fsck -F cachefs /local/mycache
#
```

## CacheFS ファイルシステムのバックアップ (作業マップ)

次の作業マップでは、CacheFS ファイルシステムのバックアップに関連する手順について説明します。これらの手順はすべてオプションです。

| 作業                                  | 説明                                                                             | 参照先                                             |
|-------------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------|
| ファイルをキャッシュにバックする                    | キャッシュにロードされるファイルとディレクトリを識別し、それらをバックする。バックアップは、これらのファイルのコピーがキャッシュで利用できることを保証する。 | 316 ページの「キャッシュにファイルをバックする方法」                    |
| バックアップリストを作成する                      | キャッシュにバックするファイルを個々に指定しない場合は、バックアップリストを作成する。                                    | 319 ページの「バックアップリストを作成する方法」                      |
| バックアップリストを使ってファイルをキャッシュにバックする       | キャッシュにバックするファイルが記載されているバックアップリストの名前を指定する。                                      | 319 ページの「バックアップリストを使ってファイルをキャッシュにバックする方法」       |
| キャッシュからファイルまたはバックアップリストのバックアップを解除する | 不要になったファイルをキャッシュから削除する。                                                        | 320 ページの「キャッシュからファイルまたはバックアップリストのバックアップを解除する方法」 |

| 作業                 | 説明                                 | 参照先                            |
|--------------------|------------------------------------|--------------------------------|
| バックされたファイルの情報を表示する | バックしたファイルに関する情報 (バックング状況など) を表示する。 | 317 ページの「バックされたファイルの情報を表示する方法」 |

## CacheFS ファイルシステムのバックング

一般的な使い方では、設定が終わると、CacheFS ソフトウェアは、ユーザーが要求しなくても自動的に適切な動作をします。最近使用されたファイルがキャッシュされます。「バックング」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。

`cachefspack` コマンドを使用すると、キャッシュにロードするファイルとディレクトリを指定できます。このコマンドは、これらのファイルのコピーがキャッシュで利用できることを保証します。

バックングリストには、特定のファイル名やディレクトリ名が入っています。他のバックングリストを入れることも可能です。この機能によって、たくさんの項目をキャッシュにバックする場合に、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。

次のように、`-h` オプションを使用すると、`cachefspack` のすべてのオプションの簡単なヘルプ情報を出力できます。

```
$ cachefspack -h
Must select 1 and only 1 of the following 5 options
-d Display selected filenames
-i Display selected filenames packing status
-p Pack selected filenames
-u Unpack selected filenames
-U Unpack all files in directory 'dir'
-f Specify input file containing rules
-h Print usage information
-r Interpret strings in LIST rules as regular expressions
-s Strip './' from the beginning of a pattern name
-v Verbose option
files - a list of filenames to be packed/unpacked
```

## キャッシュにファイルをバックする方法

ファイルをキャッシュにバックするには、`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示す。このオプションはデフォルト。
- filename** キャッシュにパックするファイルまたはディレクトリの名前を指定する。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。詳細については、cachefspack (1M) のマニュアルページを参照。

## 例 — ファイルをキャッシュにパックする

次の例は、projects ファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects
```

次の例は、3つのファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects updates master_plan
```

次の例は、ディレクトリがキャッシュにパックされることを示しています。

```
$ cachefspack -p /data/abc/bin
```

## パックされたファイルの情報を表示する方法

パックされたファイルの情報を表示するには、cachefspack -i コマンドを使用します。

```
$ cachefspack -i [v] filename
```

**-i** パックされたファイルの情報を表示することを示す。

**-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 18-8** パックされたファイルの情報を表示する

次の例は、doc\_file ファイルが正常にパックされていることを示しています。

```
$ cachefspack -i doc_file
cachefspack: file doc_file marked packed YES, packed YES
```

次の例では、/data/abc ディレクトリに bin サブディレクトリがあります。bin サブディレクトリには、big、medium、および small という3つのファイルがあります。big と small ファイルはパックするように指定されていますが、パックされていません。medium ファイルは正常にパックされています。

```
$ cd /data/abc
$ cachefspack -i bin
```

例 18-8 パックされたファイルの情報を表示する (続き)

```
.  
. .  
cachefspack: file /bin/big marked packed YES, packed NO  
cachefspack: file /bin/medium marked packed YES,  
packed YES  
cachefspack: file /bin/small marked packed YES,  
packed NO  
. .  
. .  
. .
```

-iv オプションを指定した場合は、指定したファイルまたはディレクトリがキャッシュからフラッシュされたかどうかに関する追加情報が表示されます。たとえば、次のようになります。

```
$ cd /data/bin  
FSCACHEPACK-4$ cachefspack -iv bin  
. .  
. .  
cachefspack: file /bin/big marked packed YES, packed NO,  
nocache YES  
cachefspack: file /bin/medium marked packed YES,  
packed YES, nocache NO  
cachefspack: file /bin/small marked packed YES,  
packed NO  
nocache NO  
. .  
. .  
. .
```

上記例の最後の行は、ディレクトリの内容がキャッシュからフラッシュされていないことを示しています。

## パッキングリストの使用

cachefspack コマンドの機能の1つに、パッキングリストを作成するという機能があります。

パッキングリストには、キャッシュにパックするファイルやディレクトリが入っています。ディレクトリがパッキングリストに入っている場合、そのサブディレクトリとファイルもすべてパックされます。

この機能によって、キャッシュにパックする各ファイルをそれぞれ指定するという手間が省けます。

## パッキングリストを作成する方法

パッキングリストを作成するには、vi などのエディタを使用して、ファイルを開きます。パッキングリストファイルの書式は、filesync コマンドで使用する書式と同じです。詳細については、filesync(1) のマニュアルページを参照してください。

パッキングリストには、次の 2 つの機能があります。

- パッキングリスト内のファイルを文字通りのファイル名ではなく、正規表現として識別することができるため、ファイル名を個別に指定する必要がありません。
- 所有するファイルだけがパックされるようにすることで、共有ディレクトリのファイルをパックできます。

これらの機能の使い方については、cachefspack(1M) のマニュアルページを参照してください。

### 例 18-9 パッキングリストを作成する

次の例は、パッキングリストファイルの内容を示しています。

```
BASE /home/ignatz
LIST plans
LIST docs
IGNORE *.ps
```

- BASE 文は、パックする項目が存在するディレクトリのパスを指定しています。
- 2 つの LIST 文は、当刻ディレクトリ中のパックされるファイルを指定しています。
- IGNORE 文は、パックしないファイルタイプ (ここでは .ps) を指定しています。

## パッキングリストを使ってファイルをキャッシュにパックする方法

パッキングリストのファイルをパックするには、次のように cachefspack -f コマンドを使用します。

```
$ cachefspack -f packing-list
-f          パッキングリストを使用することを示す。
packing-list パッキングリスト名を指定する。
```

### 例 18-10 パッキングリストを使ってファイルをキャッシュにパックする

この例では、list.pkg ファイルを cachefspack コマンドのパッキングリストとして使用しています。

```
$ cachefspack -f list.pkg
```

## キャッシュからファイルまたはパッキングリスト のパッキングを解除する

キャッシュからファイルを削除、つまりそのパッキングを解除する場合があります。他よりも優先度が高い一部のファイルまたはディレクトリが存在する場合があるため、重要でないファイルのパッキングを解除する必要があります。たとえば、あるプロジェクトを終了して、そのプロジェクトに関連するファイルをアーカイブしたと仮定します。次は、新しいプロジェクト、つまり新しいファイルのセットで作業することになります。

## キャッシュからファイルまたはパッキングリスト のパッキングを解除する方法

キャッシュからファイルまたはパッキングリストのパッキングを解除するには、`cachefspack` コマンドの `-u` または `-U` オプションを使用します。

```
$ cachefspack -u filename | -U cache-directory
```

`-u`           ファイルまたは複数のファイルのパッキングを解除することを示す。このオプションを使用する場合は、ファイル名を指定しなければならない。

*filename*    キャッシュからパッキングを解除するファイルまたはパッキングリストの名前を示す。

`-U`           キャッシュ中のすべてのファイルのパッキングを解除することを示す。

`cachefspack` コマンドの詳細については、マニュアルページを参照してください。

**例 18-11** キャッシュからファイルまたはパッキングリストのパッキングを解除する

次の例は、キャッシュから `/data/abc/bin/big` ファイルのパッキングが解除されることを示しています。

```
$ cachefspack -u /data/abc/bin/big
```

次の例は、キャッシュからいくつかのファイルのパッキングが解除されることを示しています。

```
$ cd /data/abc/bin/big  
$ cachefspack -u big small medium
```

次の例は、パッキングリスト (ファイルのディレクトリへのパスを含むファイル) のパッキングを解除する方法を示しています。

```
$ cachefspack -uf list.pkg
```

次の例は、キャッシュディレクトリ中のすべてのファイルのパッキングが解除されるように `-U` オプションを使用する方法を示しています。



例 18-11 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 (続き)

```
$ cachefspack -U /local/mycache
```

ファイルシステムを1つもマウントしていないキャッシュのパッキングは解除できません。-U オプションに対して、マウント済みファイルシステムを持っていないキャッシュを指定した場合、次のような出力が表示されます。

```
$ cachefspack -U /local/mycache
cachefspack: Could not unpack cache /local/mycache, no mounted
filesystems in the cache.
```

## cachefspack エラーの障害追跡

cachefspack コマンドを使用すると、次のようなエラーメッセージが表示されることがあります。

```
cachefspack: pathname - ディレクトリをオープンできません。アクセス権がありません。
```

### 原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

### 解決法

適切なアクセス権を取得してください。

```
cachefspack: pathname - ディレクトリをオープンできません。
ファイルまたはディレクトリがありません。
```

### 原因

正しいファイルまたはディレクトリがありません。

### 解決法

入力ミスがないか確認してください。

```
cachefspack: pathname - ディレクトリをオープンできません。
NFS のファイルハンドルが無効です。
```

### 原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

### 解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

```
cachefspack: pathname - ディレクトリをオープンできません。
システムコールに割り込みがかかりました。
```

### 原因

コマンドの実行中に間違って Control + C を押した可能性があります。

#### 解決法

このコマンドを再度実行してください。

cachefspack: *pathname* - ディレクトリをオープンできません。I/O エラー

#### 原因

ハードウェアの障害の可能性があります。

#### 解決法

ハードウェアの接続を確認してください。

cachefspack: ディレクトリのオープンに失敗しました。

#### 原因

正しいファイルまたはディレクトリがありません。ファイル書式中の BASE コマンドの後に指定したパスが、ディレクトリではなくファイルになっている可能性があります。指定するパスはディレクトリでなければなりません。

#### 解決法

入力ミスがないか確認してください。ファイル書式中の BASE コマンドの後に指定したパスを確認してください。ファイルではなくディレクトリが指定されていることを確認してください。

cachefspack: 共有オブジェクトを得られません。

#### 原因

実行可能ファイルが壊れているか、その形式を認識できません。

#### 解決法

実行可能ファイルを交換してください。

cachefspack: *filename* - ファイルを pack できません。アクセス権がありません。

#### 原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

#### 解決法

適切なアクセス権を取得してください。

cachefspack: *filename* - ファイルを pack できません。  
ファイルまたはディレクトリがありません。

#### 原因

正しいファイルまたはディレクトリがありません。

#### 解決法

入力ミスがないか確認してください。

cachefspack: *filename* - ファイルを pack できません。NFS のファイルハンドルが無効です。

#### 原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

#### 解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

cachefspack: *filename* - ファイルを pack できません。  
システムコールに割り込みがかかりました。

#### 原因

コマンドの実行中に間違っ**て Control + C** を押した可能性があります。

#### 解決法

このコマンドを再度実行してください。

cachefspack: *filename* - ファイルを pack できません。I/O エラー

#### 原因

ハードウェアの障害の可能性があります。

#### 解決法

ハードウェアの接続を確認してください。

cachefspack: *filename* - ファイルを pack できません。デバイス上に十分な領域がありません。

#### 原因

キャッシュのディスク容量が不足しています。

#### 解決法

ディスク容量を増やしてキャッシュのサイズを大きくする必要があります。

cachefspack: *filename* - ファイルを unpack できません。アクセス権がありません。

#### 原因

ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

#### 解決法

適切なアクセス権を取得してください。

cachefspack: *filename* - ファイルを unpack できません。  
ファイルまたはディレクトリがありません。

#### 原因

正しいファイルまたはディレクトリがありません。

#### 解決法

入力ミスがないか確認してください。

cachefspack: *filename* - ファイルを unpack できません。NFS のファイルハンドルが無効です。

#### 原因

アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

#### 解決法

サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

cachefspack: *filename* - ファイルを unpack できません。  
システムコールに割り込みがかかりました。

原因

コマンドの実行中に間違って Control + C を押した可能性があります。

解決法

このコマンドを再度実行してください。

cachefspack: *filename* - ファイルを unpack できません。I/O エラー

原因

ハードウェアの障害の可能性があります。

解決法

ハードウェアの接続を確認してください。

cachefspack: 'd'、'i'、'p'、'u' オプションのどれか 1 つを指定できます。

原因

コマンドに対して上記オプションのうち複数のオプションが指定されています。

解決法

オプションは 1 つだけ選択してください。

cachefspack: 環境変数が見つかりません。

原因

構成ファイル中で \$ で指定されている環境変数を設定していません。

解決法

環境変数を適切な場所に定義してください。

cachefspack: LIST コマンドをスキップします - 動作中の base はありません

原因

LIST コマンドが構成ファイル内にありますが、対応する BASE コマンドがありません。

解決法

BASE コマンドを定義してください。

---

## CacheFS の統計情報の収集 (作業マップ)

次の作業マップは、CacheFS の統計情報の収集に関連する手順を示しています。この表に記載されている手順はすべてオプションです。

| 作業           | 説明                                               | 参照先                                 |
|--------------|--------------------------------------------------|-------------------------------------|
| ロギングの設定      | cachefslog コマンドを使用して、CacheFS ファイルシステムのロギングを設定する。 | 327 ページの「CacheFS ロギングを設定する方法」       |
| ログファイルの検索    | cachefslog コマンドを使用してログファイルの位置を特定する。              | 327 ページの「CacheFS ログファイルの場所を調べる方法」   |
| ロギングの停止      | cachefslog コマンドを使用してロギングを停止する。                   | 328 ページの「CacheFS ロギングを停止する方法」       |
| キャッシュサイズの表示  | cachefswssize コマンドを使用してキャッシュサイズを表示する。            | 328 ページの「作業セット (キャッシュ) のサイズを表示する方法」 |
| キャッシュ統計情報の表示 | cachefsstat コマンドを使用して統計情報を表示する。                  | 329 ページの「CacheFS 統計情報を表示する方法」       |

---

## CacheFS の統計情報の収集

CacheFS の統計情報を収集すると、次の作業を行うことができます。

- 適切なキャッシュサイズを判断する。
- キャッシュのパフォーマンスを監視する。

これらの統計情報を使用すると、キャッシュサイズと望ましいパフォーマンスを選択して調整できます。

CacheFS 統計情報コマンドには次のものがあります。

| コマンド          | マニュアルページ          | 説明                                                                                     |
|---------------|-------------------|----------------------------------------------------------------------------------------|
| cachefslog    | cachefslog(1M)    | ログファイルの位置を指定します。また、このコマンドでは、統計情報が現在どこに記録されているかが表示されるので、ロギングを中止できます。                    |
| cachefswssize | cachefswssize(1M) | ログファイルを解釈して推奨キャッシュサイズを表示します。                                                           |
| cachefsstat   | cachefsstat(1M)   | 特定のファイルシステム、またはすべての CacheFS ファイルシステムに関する統計情報を表示します。このコマンドの出力に含まれる情報は、キャッシュから直接取り出されます。 |

注 - CacheFS 統計情報コマンドは、どのディレクトリから実行してもかまいません。ただし、cachefswssize コマンドを実行するには、スーパーユーザーにならなければなりません。

CacheFS 統計の累計は、ログファイルの作成時から始まります。作業時間が終わったら、cachefslog -h コマンドを使用してロギングを停止してください。手順については、328 ページの「CacheFS ロギングを停止する方法」を参照してください。

CacheFS 統計情報コマンドを使用する前に、次の操作を実行する必要があります。

- cfsadmin コマンドを使用してキャッシュを設定する。
- 作成するログファイルに統計情報を収集できるように、適切な時間を決定する。この時間は、一般的な作業間隔に等しくする必要がある。たとえば、1 日、1 週間、1 カ月などにする。
- ログファイルの位置またはパスを選択する。ログファイルが大きくなっても対応できる程度の領域があることを確認する。ログファイルに統計情報を収集できる時間を長くするほど、大きな領域が必要になる。

注 - 次の手順は推奨する順序を示しています。異なる順序で作業してもかまいません。

## ▼ CacheFS ロギングを設定する方法

手順 1. ロギングを設定します。

```
$ cachefslog -f log-file-path /mount-point
```

`-f` ロギングを設定する。

`log-file-path` ログファイルの位置を指定します。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。

`/mount-point` 統計情報を収集するマウントポイント (CacheFS ファイルシステム) を指定する。

2. ログファイルを正しく設定したかどうかを確認します。

```
$ cachefslog /mount-point
```

### 例 18-12 CacheFS ロギングを設定する

次の例は、`/var/tmp/samlog` ログファイルを設定して、`/home/sam` ディレクトリに関する統計情報を収集する方法を示しています。

```
$ cachefslog -f /var/tmp/samlog /home/sam
/var/tmp/samlog: /home/sam
```

## CacheFS ログファイルの場所を調べる方法

オプションを指定せずに `cachefslog` コマンドを使用して、特定のマウントポイントに対するログファイルの場所を調べることもできます。

```
$ cachefslog /mount-point
```

`/mount-point` は、統計情報を表示する CacheFS ファイルシステムを示します。

次の例は、ログファイルが設定されている場合の表示を示します。ログファイルは、`/var/tmp/stufflog` にあります。

```
$ cachefslog /home/stuff
/var/tmp/stufflog: /home/stuff
```

次の例は、指定したファイルシステムのログファイルが設定されていないことを示しています。

```
$ cachefslog /home/zap
not logged: /home/zap
```

## CacheFS ロギングを停止する方法

ロギングを停止するには、`cachefslog -h` オプションを使用します。

```
$ cachefslog -h /mount-point
```

次の例は、`/home/stuff` に対するロギングを停止する方法を示しています。

```
$ cachefslog -h /home/stuff
not logged: /home/stuff
```

上記の例と異なるシステム応答が表示される場合は、ロギングが正常に停止されていません。正しいログファイル名とマウントポイントを指定したかどうかを確認してください。

## ▼ 作業セット (キャッシュ) のサイズを表示する方法

キャッシュサイズを増やすべきかどうかを確認できます。または、特定のマウントポイントに関して前回 `cachefslog` コマンドを使用した後の作業に基づいて、理想的なキャッシュサイズを決定することもできます。

- 手順
1. クライアントシステムでスーパーユーザーになります。
  2. 現在のキャッシュサイズとロギングされた最大キャッシュサイズを表示します。

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

### 例 18-13 作業セット (キャッシュ) のサイズを表示する

次の例で、「end size」とは `cachefswssize` コマンドを実行した時点のキャッシュサイズです。「high water size」とは、ロギングが発生した時間枠内のキャッシュの最大サイズです。

```
# cachefswssize /var/tmp/samlog

/home/sam
    end size: 10688k
    high water size: 10704k

/
    end size: 1736k
    high water size: 1736k

/opt
    end size: 128k
    high water size: 128k
```



```

/nfs/saturn.dist
  end size: 1472k
high water size: 1472k

/data/abc
  end size: 7168k
high water size: 7168k

/nfs/venus.svr4
  end size: 4688k
high water size: 5000k

/data
  end size: 4992k
high water size: 4992k

total for cache
  initial size: 110960k
  end size: 30872k
high water size: 30872k

```

## CacheFS の統計情報の表示

特定の CacheFS ファイルシステムに関する情報を表示できます。次の表は、統計情報の出力時に表示される用語を示しています。

表 18-2 CacheFS 統計情報の用語

| 用語             | 説明                                                                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| キャッシュのヒット率     | キャッシュのヒット率対ミスヒット率の比と、それに続く実際のヒット数とミスヒット数。キャッシュヒットは、ユーザーがファイル操作を実行したいときに、そのファイルが実際にはキャッシュ内にあると発生する。キャッシュのミスヒットは、ファイルがキャッシュにないときに発生する。サーバーにかかる負荷は、キャッシュのミスヒット数、整合性チェック数、および変更数の合計である |
| 整合性チェック        | 実行された整合性チェックの回数、合格回数、不合格回数                                                                                                                                                         |
| 変更数 (modifies) | 変更操作の回数。書き込みや作成など                                                                                                                                                                  |

## CacheFS 統計情報を表示する方法

`cachefsstat` コマンドを使用して統計情報を表示します。この操作はいつでも実行できます。たとえば、ロギングを設定しなくても統計情報を表示できます。

```
$ cachefsstat /mount-point
```

`/mount-point` は、統計情報を表示する CacheFS ファイルシステムを示します。

マウントポイントを指定しなければ、マウントされているすべての CacheFS ファイルシステムに関する統計情報が表示されます。

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

**例 18-14** CacheFS 統計情報を表示する

この例は、キャッシュされたファイルシステム `/home/sam` に関する統計情報の表示方法を示しています。

```
$ cachefsstat /home/sam
    cache hit rate: 73% (1234 hits, 450 misses)
    consistency checks: 700 (650 pass, 50 fail)
        modifies: 321
garbage collection: 0
```

## 第 19 章

---

# 追加スワップ空間の構成 (手順)

---

この章では、Solaris リリースをインストールした後で追加のスワップ空間を構成するためのガイドラインと手順について説明します。

この章で説明する手順は次のとおりです。

- 339 ページの「スワップファイルを作成して使用可能にする方法」
- 340 ページの「不要になったスワップ空間を削除する方法」

この章の内容は次のとおりです。

- 331 ページの「スワップ空間について」
- 334 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」
- 335 ページの「スワップ空間の割り当て方法」
- 336 ページの「スワップ空間の計画」
- 336 ページの「スワップリソースの監視」
- 338 ページの「スワップ空間の追加」

---

## スワップ空間について

システム管理者は、次の事柄を決定する上で SunOS のスワップ機構を理解しておく必要があります。

- スワップ空間の要件
- スワップ空間と TMPFS ファイルシステムとの関係
- スワップ空間に関連するエラーメッセージからの復元

## スワップ空間と仮想メモリー

Solaris ソフトウェアは、いくつかのディスクスライスを、ファイルシステムとして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」スライスと呼びます。スワップスライスは、システムの物理メモリーが不足し現在のプロセスを処理することができないときに、仮想メモリー記憶域として使用されます。

仮想メモリーシステムは、ディスク上のファイルの物理コピーをメモリー内の仮想アドレスに対応付けます。これらのマッピングに関するデータが入った物理メモリーページは、ファイルシステム内の通常ファイルまたはスワップ空間から読み直されます。メモリーをバックアップしているディスク空間に割り当てられる ID はわからないため、スワップ空間から読み直されたメモリーは *anonymous* メモリーと呼ばれます。

Solaris 環境には、「仮想スワップ空間」という概念が導入されています。これは、*anonymous* メモリーページとこれらのページを実際にバックアップする物理記憶域（またはディスク上にとられたバックアップ用のスワップ空間）の間に位置する層です。システムの仮想スワップ空間は、すべての物理（ディスク上にとられたバックアップ用のスワップ空間）スワップ空間と現在使用可能な物理メモリーの一部の合計に等しくなります。

仮想スワップ空間の長所は次のとおりです。

- 仮想スワップ空間が物理（ディスク）記憶域に対応していなくてもかまわないので、大きな物理スワップ空間を確保する必要がなくなる。
- SWAPFS という疑似ファイルシステムが、*anonymous* メモリーページのアドレスを提供する。SWAPFS はメモリーページの割り当てを制御するので、ページに対する処理を柔軟に決定できる。たとえば、ディスク上にとられたバックアップ用のスワップ記憶域のページ要件を変更できる。

## スワップ空間と TMPFS ファイルシステム

Solaris 環境では、TMPFS ファイルシステムは `/etc/vfstab` ファイル内のエントリによって自動的に稼働されます。TMPFS ファイルシステムは、ファイルとそれに関連付けられた情報をディスクではなくメモリー（`/tmp` ディレクトリ）に格納するので、これらのファイルへのアクセスが高速になります。この機能によって、コンパイラや DBMS 製品のように `/tmp` の使用量の大きいアプリケーションの場合は、パフォーマンスが大幅に改善されます。

TMPFS ファイルシステムは、システムのスワップリソースから `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` ディレクトリ内の領域を使い果たすと、スワップ空間も使い果たしたことになります。したがって、`/tmp` ディレクトリの使用量が大きいアプリケーションの場合は、スワップ空間の使用状況を監視しなければ、システムがスワップ空間を使い果たす可能性があります。

TMPFS を使用したいがスワップリソースが限られている場合は、次の方法を使用してください。

- サイズオプション (-o size) を指定して TMPFS ファイルシステムをマウントし、TMPFS が使用できるスワップリソースを制御する。
- コンパイラの TMPDIR 環境変数を使用して、より大きな別のディレクトリを指す。

コンパイラの TMPDIR 変数を使用すると、コンパイラが /tmp ディレクトリを使用するかどうかだけが制御される。この変数は、他のプログラムによる /tmp ディレクトリの使用には影響しない。

## ダンプデバイスとしてのスワップ空間

通常、ダンプデバイスとは、システムクラッシュダンプ情報を格納するために予約されているディスク領域のことです。デフォルトでは、システムのダンプデバイスは適切なスワップパーティションになるように設定されます。可能であれば、スワップパーティションを使用する代わりに、代替パーティションを「専用ダンプデバイス」として設定してください。クラッシュダンプの信頼性を高めたり、システム障害が発生した後のリポート時間を短縮したりできます。専用ダンプデバイスの設定は、dumpadm コマンドで行えます。詳細については、『Solaris のシステム管理 (上級編)』の第 28 章「システムクラッシュ情報の管理 (手順)」を参照してください。

Solaris ボリュームマネージャなどのボリュームマネージャを使用してディスクを管理している場合は、専用ダンプデバイスを Solaris ボリュームマネージャの管理下に置かないように設定してください。スワップ領域は、Solaris ボリュームマネージャの管理下に保管することをお勧めします。ただし、使いやすさとパフォーマンスの理由から、Solaris ボリュームマネージャの管理下で動作しないディスクを専用ダンプデバイスとして設定してください。

## スワップ空間と動的再構成

動的再構成時に CPU やシステムボードで障害が発生した場合に対応できるよう、十分なスワップ空間を確保することをお勧めします。スワップ空間が不足していると、CPU やシステムボードで障害が発生した際に、ホストまたはドメインはより少ないメモリーでリポートしなければなりません。

この追加スワップ空間を利用可能にしなかった場合、1 つ以上のアプリケーションがメモリー不足により起動できなくなる可能性があります。その場合、ユーザーが手動で介入し、追加スワップ空間を追加するか、それらのアプリケーションのメモリー使用量を調整する必要があります。

リポート時のメモリー破損に備えて十分な追加スワップが確保されていた場合、メモリーを大量に消費するアプリケーションのすべてが、通常どおりに起動します。したがって、スワップの多発によりシステムの動作は多少遅くなるにしても、ユーザーは通常どおりにシステムを利用できます。

詳細については、使用するハードウェアの動的構成マニュアルを参照してください。

---

## スワップ空間の追加が必要かどうかを調べる方法

スワップ空間の追加が必要かどうかを調べるには、`swap -l` コマンドを使用します。

たとえば、次の `swap -l` コマンドの出力は、このシステムのスワップ空間がほぼ完全に使い果たされたか、割り当て率が 100% に達していることを示しています。

```
% swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1    16 1638608    88
```

システムのスワップ空間の割り当て率が 100% に達すると、アプリケーションのメモリーページが一時的にロックしてしまいます。アプリケーションエラーが発生しない場合でも、システムパフォーマンスが低下する可能性があります。

システムにスワップ空間を追加する方法については、[339 ページの「スワップファイルを作成して使用可能にする方法」](#)を参照してください。

## スワップ関連のエラーメッセージ

次のメッセージは、アプリケーションがさらに `anonymous` メモリ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

次のメッセージは、ファイルに書き込むときにページを割り当てることができない場合に表示されます。この問題は、TMPFS が許容限度を超えて書き込もうとしたときや、現在実行されているプログラムが大量のメモリーを使用している場合に発生することがあります。

```
directory: File system full, swap space limit exceeded
```

次のメッセージは、TMPFS が新しいファイルやディレクトリの作成中に物理メモリーを使い果たしたことを意味します。

```
directory: File system full, memory allocation failed
```

TMPFS 関連のエラーメッセージから回復する方法については、tmpfs(7FS) のマニュアルページを参照してください。

---

## スワップ空間の割り当て方法

最初に、スワップ空間は Solaris インストールプロセスの一部として割り当てられます。インストールプログラムによるディスクスライスの自動レイアウトを選択し、スワップファイルのサイズを手作業で変更しなければ、Solaris インストールプログラムはデフォルトのスワップ領域 (512M バイト) を割り当てます。

Solaris 9 リリースよりインストールプログラムは利用可能な最初のディスクシリンダ (通常はシリンダ 0) にスワップ空間を割り当てるようになりました。この配置によって、デフォルトのディスクレイアウト時にはルート (/) ファイルシステムに最大のスワップ空間を割り当てることができ、アップグレード時にはルート (/) ファイルシステムを拡張できます。

スワップ空間の割り当て方法については、336 ページの「スワップ空間の計画」を参照してください。

スワップファイルを作成すると、スワップ空間をシステムに追加できます。スワップファイルの作成方法については、338 ページの「スワップ空間の追加」を参照してください。

## /etc/vfstab ファイル

システムをインストールすると、スワップスライスとスワップファイルが /etc/vfstab ファイル内に列挙されます。スワップスライスとスワップファイルは、システムのブート時に /sbin/swapadd スクリプトによって有効になります。

/etc/vfstab ファイル内のスワップデバイスエントリには、次の情報が入っています。

- スワップスライスまたはスワップファイルのフルパス名
- スワップのファイルシステムタイプ

スワップファイルが入っているファイルシステムは、スワップファイルが有効になる前にマウントしておかなければなりません。このため、/etc/vfstab ファイル内で、ファイルシステムをマウントするエントリが、スワップファイルを有効にするエントリより前に入っていることを確認してください。

---

## スワップ空間の計画

スワップ空間のサイズを決定する上でもっとも重要な要素は、システムのソフトウェアアプリケーションの要件です。たとえば、コンピュータ支援設計シミュレータ、データベース管理製品、トランザクションモニター、地質分析システムなどの大型アプリケーションは、200 ~ 1000M バイトのスワップ空間を消費することがあります。

アプリケーションのスワップ空間の要件については、アプリケーションベンダーに問い合わせてください。

アプリケーションベンダーに問い合わせてもスワップ空間の要件を判断できない場合は、次のシステムタイプ別のガイドラインに従ってスワップ空間を割り当ててください。

| システムタイプ                             | スワップ空間のサイズ | 専用ダンプデバイスのサイズ |
|-------------------------------------|------------|---------------|
| 約 4G バイトの物理メモリーを備えたワークステーション        | 1G バイト     | 1G バイト        |
| 約 8G バイトの物理メモリーを備えた中型のサーバー          | 2G バイト     | 2G バイト        |
| 約 16 ~ 128G バイトの物理メモリーを備えたハイエンドサーバー | 4G バイト     | 4G バイト        |

一般的なガイドラインのほかに、次の場合のスワップ空間やディスク領域の割り当てでも検討してください。

- 専用ダンプデバイス
- 大型アプリケーション (コンパイラなど) が /tmp ディレクトリを使用するかどうかを決定する。次に、TMPFS によって使用される追加のスワップ空間を割り当てる。TMPFS については、332 ページの「スワップ空間と TMPFS ファイルシステム」を参照

---

## スワップリソースの監視

/usr/sbin/swap コマンドを使用してスワップ領域を管理します。2つのオプション -l と -s は、スワップリソースに関する情報を表示します。

swap -l コマンドを使用すると、システムのスワップ領域を確認できます。有効になっているスワップデバイスやファイルは、swapfile カラムの下に表示されます。



```
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1      16 1638608 1600528
```

swap -s コマンドを使用すると、スワップリソースを監視できます。

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

used 値と available 値の合計は、システム上の合計スワップ空間に等しくなります。これには、物理メモリーの一部とスワップデバイス (またはファイル) が含まれません。

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。

---

次の表に、swap -s コマンドの出力とその説明を示します。

表 19-1 swap -s コマンドの出力

| キーワード           | 説明                                                                             |
|-----------------|--------------------------------------------------------------------------------|
| bytes allocated | 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 |
| reserved        | 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           |
| used            | 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                                    |

表 19-1 swap -s コマンドの出力 (続き)

| キーワード     | 説明                                            |
|-----------|-----------------------------------------------|
| available | 後から予約や割り当てに使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数 |

## スワップ空間の追加

システム構成を変更して新しいソフトウェアパッケージをインストールした後に、スワップ空間を追加しなければならないことがあります。その場合に望ましいのは、mkfile コマンドと swap コマンドを使用して、既存の UFS または NFS ファイルシステムの一部を補助スワップ空間として指定する方法です。次の節で説明するように、これらのコマンドを使用すると、ディスクをパーティションに分割し直さなくても、スワップ空間を追加できます。

スワップ空間を追加するもう 1 つの方法は、既存のディスクをパーティションに分割し直すか、別のディスクを追加することです。ディスクをパーティションに分割し直す方法については、[第 10 章](#)を参照してください。

## スワップファイルの作成

次の手順でスワップファイルを作成します。

- mkfile コマンドを使用してスワップファイルを作成する。
- 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。スティッキビットが設定され、ファイルは 0 が埋め込まれます。スワップファイルのサイズは、バイト数 (デフォルト) として指定するか、接尾辞 k、b、m を使用して、それぞれ K バイト数、ブロック数、M バイト数として指定できます。

次の表に、mkfile コマンドのオプションを示します。

表 19-2 mkfile コマンドのオプション

| オプション | 説明                                                    |
|-------|-------------------------------------------------------|
| -n    | 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。 |
| -v    | 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

手順 1. スーパーユーザーになります。

root の権限がなくてもスワップファイルを作成できます。しかし、スワップファイルが誤って上書きされないように、root を所有者にしておくでしょう。

2. 必要に応じて、スワップファイル用のディレクトリを作成します。

3. スワップファイルを作成します。

```
# mkfile nnn[k|b|m] filename
```

指定したサイズ「*nnn*」(K バイト、バイト、または M バイト) とファイル名でスワップファイルが作成されます。

4. スワップファイルを有効にします。

```
# /usr/sbin/swap -a /path/filename
```

絶対パス名を使用してスワップファイルを指定しなければなりません。スワップファイルが追加され、ファイルシステムがマウント解除されるか、またはシステムがリブートされるまで使用可能になります。プロセスまたはプログラムがスワップファイルにスワップしているときは、ファイルシステムのマウントを解除できないことに注意してください。

5. 次のように、ファイルのフルパス名を指定し、ファイルシステムのタイプとして **swap** を指定して、スワップファイルのエントリを **/etc/vfstab** ファイルに追加します。

```
/path/filename - - swap - no -
```

6. スワップファイルが追加されたことを確認するには、次のように入力します。

```
$ /usr/sbin/swap -l
```

### 例 19-1 スワップファイルを作成して使用可能にする

次の例は、`/files/swapfiles` という 100M バイトのスワップファイルを作成する方法を示しています。

```
# mkdir /files
# mkfile 100m /files/swapfile
# swap -a /files/swapfile
# vi /etc/vfstab
(エントリがスワップファイル用に追加される):
/files/swapfile - - swap - no -
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1  16 1638608 1600528
/files/swapfile   -      16 204784   204784
```

---

## スワップファイルの削除

不要になったスワップ空間は、削除することができます。

### ▼ 不要になったスワップ空間を削除する方法

手順 1. スーパーユーザーになります。

2. スワップ空間を削除します。

```
# /usr/sbin/swap -d /path/filename
```

スワップファイル名が削除されるので、スワッピングに使用できなくなります。ファイルそのものは削除されません。

3. `/etc/vfstab` ファイルを編集して、スワップファイルのエントリを削除します。

4. 他の目的に使用できるようにディスク領域を復元します。

```
# rm /path/filename
```

スワップ空間がファイルの場合は削除します。また、スワップ空間が別のスライスに入っていて、不要なことがわかっている場合は、新しいファイルシステムを作成してマウントします。

ファイルシステムのマウントについては、[第 17 章](#)を参照してください。

5. 削除したスワップファイルが使用できなくなっていることを確認します。

```
# swap -l
```

## 例 19-2 不要になったスワップ空間を削除する

次の例は、`/files/swapfile` スワップファイルを削除する方法を示しています。

```
# swap -d /files/swapfile
# (スワップエントリを /etc/vfstab ファイルから削除する)
# rm /files/swapfile
# swap -l
swapfile                dev  swaplo  blocks  free
/dev/dsk/c0t0d0s1      136,1      16 1638608 1600528
```



## 第 20 章

# UFS ファイルシステムの整合性 チェック (手順)

---

この章では、UFS ファイルシステムの整合性チェックに関する概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 353 ページの「代替ブートデバイスからルート (/) ファイルシステムまたは /usr ファイルシステムをチェックする方法」
- 355 ページの「ルート (/) 以外、または /usr 以外のファイルシステムをチェックする方法」
- 357 ページの「UFS ファイルシステムを修復する方法」
- 359 ページの「不正なスーパーブロックを復元する方法」

この章の内容は次のとおりです。

- 344 ページの「ファイルシステムの整合性」
- 344 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」
- 346 ページの「fsck コマンドでチェックして修復される内容」
- 352 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」
- 359 ページの「不正なスーパーブロックの復元」
- 361 ページの「fsck コマンドの構文とオプション」

fsck のエラーメッセージについては、『Solaris のシステム管理 (上級編)』の第 32 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

この章で参照される UFS ファイルシステム構造の内容については、第 21 章を参照してください。

---

## ファイルシステムの整合性

UFS ファイルシステムは、一連の内部テーブルを基にして使用済み i ノード、使用可能ブロックを特定します。これらの内部テーブルがディスク上のデータと正しく同期していないと、整合性が失われ、ファイルシステムの修復が必要になります。

次のような原因でオペレーティングシステムが異常終了すると、ファイルシステムの整合性が失われることがあります。

- 電源障害
- 不注意によるシステム電源の切断
- 正しいシャットダウン手順以外の方法によるシステム電源の切断
- カーネル内のソフトウェアエラー

ファイルシステムの整合性が失われることは重大ですが、頻繁に起きるものではありません。システムをブートすると、ファイルシステムの整合性チェックが (fsck コマンドを使用して) 自動的に実行されます。ほとんどの場合は、このファイルシステムのチェックによって問題が修復されます。

fsck コマンドは、ファイルシステム上に配置されているが参照不可能なファイルとディレクトリを lost+found ディレクトリに入れます。参照不可能なファイルとディレクトリの名前として i ノード番号が割り当てられます。lost+found ディレクトリが存在しない場合は、fsck コマンドによって作成されます。lost+found ディレクトリ内の領域が足りない場合は、fsck コマンドによってそのサイズが拡張されます。

i ノードについては、[373 ページの「i ノード」](#)を参照してください。

---

## ファイルシステムの状態はどのように記録されるか

fsck コマンドは、スーパーブロックに格納された状態フラグを使用して、ファイルシステムの状態を記録します。また、このフラグを使用して、ファイルシステムの整合性をチェックする必要があるか判断します。このフラグは、ブート時に /sbin/rcS スクリプトによって使用されるか、fsck -m コマンドによって使用されます。fsck -m コマンドの結果を無視する場合は、状態フラグの設定に関係なく、すべてのファイルシステムをチェックできます。

スーパーブロックについては、[372 ページの「スーパーブロック」](#)を参照してください。

次の表に状態フラグの値とその説明を示します。



表 20-1 ファイルシステムの状態フラグの値

| 状態フラグの値  | 説明                                                                                                                                                                                                        |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FSACTIVE | このファイルシステムがマウント済みであり、メモリー内のデータが変更されたことを示す。この状態フラグを持つマウント済みファイルシステムでは、システムの電源が切断されると、ユーザーデータまたはメタデータが失われる。                                                                                                 |
| FSBAD    | このファイルシステムに不整合なファイルシステムデータが含まれていることを示す。                                                                                                                                                                   |
| FSCLEAN  | このファイルシステムが正常にマウント解除されており、破損していないことを示す。                                                                                                                                                                   |
| FSLOG    | このファイルシステムのロギングが有効になっていることを示す。このフラグが設定されたファイルシステムは、マウントされている場合も、マウント解除されている場合もある。ロギングが有効になっているファイルシステムに設定できるフラグは、FSLOG、FSBAD のいずれかだけである。ロギングが無効になっているファイルシステムに設定できるのは、FSACTIVE、FSSTABLE、FSCLEAN のいずれかである。 |
| FSSTABLE | このファイルシステムがマウント済みであり、アイドル状態であることを示す。この状態フラグを持つマウント済みファイルシステムでは、システムの電源が切断されても、ユーザーデータやメタデータが失われることはない。                                                                                                    |

次の表に、fsck コマンドを使用して、初期状態に基づいて状態フラグを変更する方法を示します。

表 20-2 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 コマンドでチェックして修復される内容

この節では、ファイルシステムの通常の処理中に発生する問題、原因、fsck コマンド (チェックおよび修復ユーティリティ) で検出される問題、およびそれらの修正方法について説明します。

### 不整合が発生する原因

就業日には毎日多数のファイルが作成、変更、または削除されます。ファイルが変更されるたびに、オペレーティングシステムは一連のファイルシステムの更新処理を実行します。これらの更新処理がディスクに確実に書き込まれると、ファイルシステムの整合性が保たれます。

ユーザープログラムが書き込みなどの、ファイルシステムを変更する処理を実行すると、書き込まれるデータはまずカーネルのインコアバッファーにコピーされます。一般に、ディスクの更新は非同期に処理されます。このため、ユーザープロセスは、書き込みシステムコールが値を返した後すぐに処理を続けることができますが、実際のデータの書き込みは、ずいぶん後に実行されることもあります。したがって、ディスク上にあるファイルシステムは、インコア情報で表されるファイルシステムの状態から常に遅延することになります。

別の目的にバッファーが必要になったり、カーネルが fsflush デーモンを自動的に (30 秒間隔で) 実行すると、インコア情報を反映するようにディスク情報が更新されます。システムがインコア情報を書き込まずに停止すると、ディスク上のファイルシステムの整合性がなくなります。

ファイルシステムの整合性は、さまざまな原因で失われることがあります。もっとも一般的な原因は、オペレータのエラーとハードウェア障害です。

システムを正しくシャットダウンしなかったり、マウントされているファイルシステムが正しくオフラインにされないと、「クリーンでない停止」が原因で問題が発生することがあります。クリーンでないシャットダウンを防ぐには、システムをシャットダウンしたり、ディスクをドライブから物理的に取り出したり、ディスクをオフライン状態にしたりする前に、ファイルシステムの現在の状態をディスクに書き込まなければなりません。つまり、「同期」させなければなりません。

また、ハードウェアの欠陥や、ディスクまたはコントローラのファームウェアの問題が原因で整合性が失われることもあります。ディスクドライブ上ではいつでもブロックが損傷する可能性があり、ディスクコントローラが正常に機能しなくなる可能性があります。

## 整合性がチェックされる UFS 構成要素

この節では、UFS ファイルシステムの構成要素、つまりスーパーブロック、シリンダグループブロック、i ノード、間接ブロック、データブロックに `fsck` コマンドが適用する整合性チェックの種類について説明します。

UFS ファイルシステム構造については、372 ページの「[UFS ファイルシステムのシリンダグループの構造](#)」を参照してください。

### スーパーブロックのチェック

スーパーブロックには集計情報が格納されており、UFS ファイルシステム内でもっとも破損しがちな構成要素です。ファイルシステムの i ノードやデータブロックが変更されるたびに、スーパーブロックも変更されます。CPU が停止した場合、直前のコマンドが `sync` コマンドでなければ、スーパーブロックはほぼ確実に破損しています。

スーパーブロックの不整合は、次の面からチェックされます。

- ファイルシステムのサイズ
- i ノード数
- 空きブロック数
- 空き i ノード数

### ファイルシステムのサイズと i ノードリストのサイズのチェック

ファイルシステムのサイズは、スーパーブロックと i ノードリストに使用されるブロック数よりも大きくなければなりません。i ノード数は、ファイルシステムの最大許容数よりも小さくなくてはなりません。i ノードは、ファイルに関するすべての情報を表します。ファイルシステムのサイズとレイアウト情報は、`fsck` コマンドにとってもっとも重要な情報部分です。これらのサイズはファイルシステムの作成時に静的に決められるため、実際にチェックする方法はありません。ただし、`fsck` コマンドを使用してサイズが妥当な範囲内にあるかどうかはチェックできます。ファイルシステムの他のすべてのチェックを行うには、これらのサイズが正確でなければなりません。`fsck` コマンドが一次スーパーブロックの静的パラメータ内に不正な情報を検出すると、オペレータに代替スーパーブロックの位置を指定するように促します。

UFS ファイルシステム構造の詳細については、372 ページの「[UFS ファイルシステムのシリンダグループの構造](#)」を参照してください。

### 空きブロック数のチェック

空きブロック数は、シリンダグループのブロックマップに格納されます。`fsck` コマンドは、空きマーク付きのすべてのブロックがファイルによって使用されていないかどうかをチェックします。すべてのブロックをチェックし終わると、`fsck` コマンドは空きブロック数と i ノードによって使用されるブロック数の合計がファイルシステム内の合計ブロック数に等しくなるかどうかをチェックします。ブロックマップ内に間違いがあると、`fsck` コマンドはブロックが割り当てられている状態のままで構築し直します。

スーパーブロック内の集計情報には、ファイルシステム内の空きブロックの合計数のカウントが入っています。fsck コマンドは、このブロック数をファイルシステム内で見つかった空きブロック数と比較します。数が一致しなければ、fsck コマンドはスーパーブロック内の空きブロック数を実際の空きブロック数で置き換えます。

## 空き *i* ノード数のチェック

スーパーブロック内の集計情報には、ファイルシステム内の空き *i* ノード数が入っています。fsck コマンドは、この *i* ノード数をファイルシステム内で見つかった空き *i* ノード数と比較します。数が一致しなければ、fsck はスーパーブロック内の空き *i* ノード数を実際の空き *i* ノード数で置き換えます。

## *i* ノード

*i* ノードリストは、*i* ノード 2 から順番にチェックされます (*i* ノード 0 と *i* ノード 1 は予約済み)。各 *i* ノードの不整合は、次の面からチェックされます。

- 形式とタイプ
- リンク数
- 重複ブロック
- 不正なブロック番号
- *i* ノードのサイズ

## *i* ノードの形式とタイプ

各 *i* ノードには、そのタイプと状態を記述するモードのワードが入っています。*i* ノードには、次の 9 つのタイプがあります。

- 通常ファイル
- ディレクトリ
- ブロック型特殊ファイル
- キャラクタ型特殊ファイル
- FIFO (名前付きパイプ)
- シンボリックリンク
- シャドウ (ACL で使用される)
- 属性ディレクトリ
- ソケット

*i* ノードの状態は、次の 3 つに分かれています。

- 割り当て済み
- 未割り当て
- 不完全に割り当て済み

ファイルシステムが作成されると、一定数の *i* ノードが確保されますが、必要になるまでは割り当てられません。割り当て済みの *i* ノードとは、ファイルを指す *i* ノードです。未割り当ての *i* ノードは、ファイルを指さないで空のはずです。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 コマンドが実行できる唯一の修正動作は、その i ノードを消去することです。

## リンク数のチェック

各 i ノードには、そこにリンクされているディレクトリエントリ数が入っています。fsck コマンドは、ルートディレクトリから順番にディレクトリ構造全体を検査し、i ノードごとに実際のリンク数を計算して、各 i ノードのリンク数を検査します。

i ノードに格納されているリンク数が fsck コマンドによって判断された実際のリンク数と一致しない場合は、次の 3 つの状況が考えられます。

- 格納されたリンク数が 0 でなく、実際のリンク数が 0 の場合  
この状況は、i ノードにリンクされているディレクトリエントリが存在しない場合に発生することがあります。この場合、fsck コマンドはリンクされていないファイルを lost+found ディレクトリに入れます。
- 格納されたリンク数が 0 でなく、実際のリンク数も 0 でないが、2 つのリンク数が等しくない場合  
この状況は、ディレクトリエントリが追加または削除されたが、i ノードが更新されていない場合に発生することがあります。この場合、fsck コマンドは格納されたリンク数を実際のリンク数で置き換えます。
- 格納されたリンク数が 0 で実際のリンク数が 0 でない場合  
この場合、fsck コマンドは i ノードのリンク数を実際のリンク数に変更します。

## 重複ブロックのチェック

各 i ノードには、それが使用するすべてのブロックのリスト、またはリストを指すポインタ (間接ブロック) が入っています。間接ブロックは i ノードによって所有されるので、間接ブロックの整合性が失われると、それを所有する i ノードが直接影響を受けます。

fsck コマンドは、i ノードから使用される各ブロック番号を、割り当て済みブロックのリストと比較します。別の i ノードからすでにブロック番号が使用されていると、そのブロック番号は重複ブロックのリストに入れられます。それ以外の場合は、割り当て済みブロックのリストが更新され、ブロック番号が追加されます。

重複ブロックがあると、fsck コマンドは再び i ノードリストを調べて、各重複ブロックを使用する他の i ノードを検索します。i ノード内に大量の重複ブロックが入っている場合は、ファイルシステムに間接ブロックが正しく書き込まれていない可能性があります。どの i ノードにエラーがあるかを正確に判断することはできません。fsck コマンドは、保持する i ノードと消去する i ノードを選択するように促すプロンプトを表示します。

## 不正なブロック番号のチェック

fsck コマンドは、i ノードから使用される各ブロック番号をチェックして、その値がファイルシステム内の最初のデータブロック番号よりも大きく、最後のデータブロック番号より小さいかどうかを調べます。ブロック番号がこの範囲に含まれない場合は、不正なブロック番号と見なされます。

間接ブロックがファイルシステムに正しく書き込まれていないことが原因で、i ノード内に不正なブロック番号が発見されることがあります。fsck コマンドはその i ノードの消去を促すプロンプトを表示します。

## i ノードサイズのチェック

各 i ノードには、参照するデータブロック数が入っています。実際のデータブロック数は、割り当て済みのデータブロック数と間接ブロック数の合計です。fsck コマンドはデータブロック数を計算し、そのブロック数を i ノードから使用されるブロック数と比較します。i ノードに不正なブロック数が入っていると、fsck コマンドはその修正を促すプロンプトを表示します。

各 i ノードには、64 ビットのサイズフィールドがあります。このフィールドは、i ノードに関連付けられたファイル内の文字数 (データバイト数) を示します。i ノードのサイズフィールドに整合性があるかどうかは、サイズフィールド内の文字数を使用して、i ノードに関連付けるべきブロック数を計算し、その結果を i ノードから使用される実際のブロック数と比較して概算でチェックされます。

## 間接ブロック

間接ブロックは i ノードによって所有されます。したがって、間接ブロック内の整合性が失われると、それを所有する i ノードが影響を受けます。不整合は、次の面からチェックされます。

- すでに別の i ノードから使用されているブロック
- ファイルシステムの範囲に含まれないブロック番号

また、間接ブロックに対しても整合性チェックが実行されます。

## データブロック

i ノードは、3 種類のデータブロックを直接または間接に参照できます。参照されるブロックは、すべて同じ種類でなければなりません。次の 3 種類のデータブロックがあります。

- プレーンデータブロック
- シンボリックリンクデータブロック
- ディレクトリデータブロック

プレーンデータブロックには、ファイルに格納される情報が入っています。シンボリックリンクデータブロックには、シンボリックリンクに格納されるパス名が入っています。ディレクトリデータブロックには、ディレクトリエントリが入っています。fsck コマンドはディレクトリデータブロックの妥当性しかチェックできません。

ディレクトリは、*i* ノードの `mode` フィールド内のエントリによって通常ファイルと区別されます。ディレクトリに関連付けられたデータブロックには、ディレクトリエントリが入っています。ディレクトリデータブロックの不整合は、次の面からチェックされます。

- 未割り当ての *i* ノードを指すディレクトリ内の *i* ノード番号
- ファイルシステム内の *i* ノード番号より大きいディレクトリ内の *i* ノード番号
- 「`.`」と「`..`」ディレクトリには許されないディレクトリ内の *i* ノード番号
- ファイルシステムから切り離されたディレクトリ

## 未割り当てディレクトリのチェック

ディレクトリデータブロック内の *i* ノード番号が未割り当て *i* ノードを指す場合、`fsck` コマンドはそのディレクトリエントリを削除します。この状況は、新しいディレクトリエントリが入っているデータブロックが変更されて書き出されたが、*i* ノードが書き込まれていない場合に発生します。また、警告なしに CPU が停止された場合にも発生します。

## 不正な *i* ノード番号のチェック

ディレクトリエントリの *i* ノード番号が *i* ノードリストの最後を超える位置を指す場合、`fsck` コマンドはそのディレクトリエントリを削除します。この状況は、不正なデータがディレクトリのデータブロックに書き込まれると発生します。

## 不正な「`.`」と「`..`」エントリ

「`.`」ディレクトリの *i* ノード番号は、ディレクトリデータブロックの最初のエントリでなければなりません。また、それ自体を参照しなければなりません。つまり、その値はディレクトリデータブロックの *i* ノード番号に等しくなければなりません。

「`..`」ディレクトリの *i* ノード番号は、ディレクトリデータブロックの第 2 のエントリでなければなりません。その値は、親ディレクトリの *i* ノード番号 (または、ディレクトリがルートディレクトリの場合は、それ自体の *i* ノード番号) に等しくなければなりません。

「`.`」と「`..`」ディレクトリの *i* ノード番号が不正であれば、`fsck` コマンドは正しい値に置き換えます。ディレクトリへのハードリンクが複数個ある場合は、最初に見つかったハードリンクが「`..`」が指す実際の親であると見なされます。この場合、`fsck` コマンドは他の名前を削除するように促すプロンプトを表示します。

## 切り離されたディレクトリ

`fsck` コマンドは、ファイルシステム全体で参照関係をチェックします。ファイルシステムにリンクされていないディレクトリが見つかったら、`fsck` コマンドはそのディレクトリをファイルシステムの `lost+found` ディレクトリにリンクします。この状況は、*i* ノードがファイルシステムに書き込まれたが、それに対応するディレクトリデータブロックが書き込まれていない場合にも発生することがあります。

## 通常データブロック

通常ファイルに関連付けられたデータブロックには、ファイルの内容が入っていません。fsck コマンドは、通常ファイルのデータブロックの内容が有効かどうかはチェックしません。

## fsck 要約メッセージ

fsck コマンドを対話式で実行して正常に終了すると、次のようなメッセージが表示されます。

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)
#
```

fsck 出力の最後の行は、ファイルシステムについて次のような情報を記述します。

|                 |                                           |
|-----------------|-------------------------------------------|
| # files         | 使用中の i ノード数                               |
| # used          | 使用中のフラグメント数                               |
| # free          | 未使用のフラグメント数                               |
| # frags         | 未使用の非ブロックフラグメント数                          |
| # blocks        | 未使用の完全ブロック数                               |
| % fragmentation | 断片化の比率。ファイルシステム内の空きフラグメント × 100 / 全フラグメント |

フラグメントについては、[376 ページの「フラグメントサイズ」](#)を参照してください。

---

## UFS ファイルシステムを対話式で チェックして修復する

次の場合には、ファイルシステムを対話式でチェックする必要があります。

- マウントできない場合
- 使用中に不整合が発生する場合



使用中のファイルシステムの整合性が失われると、コンソールウィンドウやシステムメッセージファイルにエラーメッセージが出力されたり、システムがクラッシュしたりすることがあります。たとえば、システムメッセージファイル `/var/adm/messages` に次のようなメッセージが出力されることがあります。

```
Sep  5 13:42:40 hostname ufs: [ID 879645 kern.notice] NOTICE: /: unexpected
free inode 630916, run fsck(1M)
```

`hostname` は、エラーを報告したシステムです。

`fsck` コマンドを使用する前に、`fsck` のエラーメッセージの解決方法について、[361 ページの「fsck コマンドの構文とオプション」](#) および『Solaris のシステム管理 (上級編)』の第 32 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

`fsck` コマンドを実行して UFS ファイルシステムをチェックするときには、次の点を考慮してください。

- `fsck` を使用してチェックするファイルシステムは、非アクティブでなければならない。ディスクにまだフラッシュされていないファイルシステムの変更がある場合、または `fsck` によるチェックが進行中にファイルシステムを変更した場合、それらの変更はファイルシステムの破壊と解釈されることがあり、検出された問題の信頼性が低下する可能性がある。
- `fsck` を使用して修復するファイルシステムは、非アクティブでなければならない。ディスクにまだフラッシュされていないファイルシステムの変更がある場合、または `fsck` による修復が進行中にファイルシステムを変更した場合、ファイルシステムが破壊したり、システムがクラッシュすることがある。
- `fsck` を使用する前に対象のファイルシステムをマウント解除して非アクティブにすること。これは、ファイルシステムのすべてのデータ構造について、できるだけ整合性を確保するためである。ただし、ルート (`/`) ファイルシステムと `/usr` ファイルシステムだけは、アクティブにする。これらのファイルシステムは、`fsck` を実行するためにマウントする必要がある。
- ルート (`/`) ファイルシステムまたは `/usr` ファイルシステムを修復する必要がある場合には、可能であれば代替デバイスからシステムをブートすると、マウント解除されて非アクティブになる。

ルート (`/`) ファイルシステムまたは `/usr` ファイルシステムに対して `fsck` を実行する手順については、[353 ページの「代替ブートデバイスからルート \(`/`\) ファイルシステムまたは `/usr` ファイルシステムをチェックする方法](#) を参照してください。

## ▼ 代替ブートデバイスからルート (`/`) ファイルシステムまたは `/usr` ファイルシステムをチェックする方法

この手順では、ローカル CD またはネットワークブートサーバーを利用でき、代替デバイスからシステムをブートできることを前提とします。

不正なスーパーブロックの復元については、359 ページの「不正なスーパーブロックを復元する方法」を参照してください。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. ルート(/) ファイルシステムがミラー化されているシステムの場合のみ: まずルート(/) ミラーを切り離して、その後で代替デバイスからブートしてください。接続した状態で操作すると、ファイルシステムが破壊する可能性があります。  
ルート(/) ミラーの切り離しについては、『Solaris ボリュームマネージャの管理』の「サブミラーに関する作業」を参照してください。
  3. チェックの必要なルート(/) ファイルシステムまたは /usr ファイルシステムのデバイス (/dev/dsk/c0t0d0s0 など) を特定します。  
このデバイス名は、代替デバイスからブートした後で指定する必要があります。代替デバイスからブートした後でデバイスを特定するのは簡単ではありません。
  4. チェックの必要なルート(/) ファイルシステムまたは /usr ファイルシステムのあるシステムを、ローカル CD またはネットワークなどの代替デバイスからブートします。このとき、これらのファイルシステム上でほかの動作が存在しないように、シングルユーザーモードでブートしてください。  
たとえば、次のように入力します。

```
# init 0
ok boot net -s
.
.
#
```

5. 手順3 で特定したルート(/) ファイルシステムまたは /usr ファイルシステムが含まれるデバイスをチェックします。  
チェックまたは修復するファイルシステムのハードウェアが変更されている場合には、そのデバイス名も変更されている可能性があります。fsck -n メッセージの「Last Mounted on ...」を利用して、そのファイルシステムのデバイスが予期したデバイスであることを確認してください。  
たとえば、チェックするルートファイルシステムが /dev/dsk/c0t0d0s0 の場合は、次のように入力されます。

```
# fsck -n /dev/rdisk/c0t0d0s0
** /dev/rdisk/c0t0d0s0 (NO WRITE)
** Last Mounted on /
.
.
.
fsck /dev/rdisk/c0t0d0s0
** /dev/rdisk/c0t0d0s0
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
```

・  
・  
・

6. 報告された **fsck** エラーをすべて修正します。

1 つまたは複数の UFS ファイルシステムを対話式でチェックしながら、エラーメッセージのプロンプトに応答する方法については、『Solaris のシステム管理 (上級編)』の第 32 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

7. 「**FILE SYSTEM STATE NOT SET TO OKAY**」や「**FILE SYSTEM MODIFIED**」などのメッセージが表示される場合は、必要に応じて **fsck** コマンドをもう一度実行してください。

**fsck** コマンドは、一度の実行ではすべてのエラーを修正できないことがあります。

**fsck** コマンドを何度か実行してもすべての問題を修復できない場合は、358 ページの「**fsck** コマンドで修復できない UFS ファイルシステムの修正」を参照してください。

8. 修復したファイルシステムをマウントして、**lost+found** ディレクトリにファイルが存在するかどうかを確認します。

**fsck** コマンドによって **lost+found** ディレクトリに入れられた各ファイルの名前は、その *i* ノード番号を使用して変更されます。可能であれば、ファイル名を変更し、ファイルが含まれるべき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内の語句を探したり、**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。

どうしても特定できないファイルまたはディレクトリについては、**lost+found** ディレクトリからそれらを削除して、不要な領域を解放してください。

9. システムをマルチユーザーモードに戻します。

```
# init 6
```

代替デバイスからシングルユーザーモードでブートしている場合には、Control + D を押すと、Solaris のインストール処理が開始されます。

10. ルート (/) ファイルシステムがミラー化されているシステムの場合のみ: ルート (/) ミラーを再接続します。

## ▼ ルート (/) 以外、または /usr 以外のファイルシステムをチェックする方法

この手順では、チェックするファイルシステムがマウント解除されていることを前提としています。

不正なスーパーブロックの復元については、359 ページの「不正なスーパーブロックを復元する方法」を参照してください。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 最初にローカルファイルシステムをマウント解除し、ファイルシステム上でほかの動作が存在しないようにします。

`fsck` コマンドの引数として、マウントポイントディレクトリや `/dev/dsk/device-name` を指定します。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。

たとえば、次のように入力します。

```
# umount /export/home
# fsck /dev/rdisk/c0t0d0s7
** /dev/dsk/c0t0d0s7
** Last Mounted on /export/home
.
.
.
```

3. 報告された `fsck` エラーをすべて修正します。  
1 つまたは複数の UFS ファイルシステムを対話式でチェックしながら、エラーメッセージのプロンプトに回答する方法については、『Solaris のシステム管理 (上級編)』の第 32 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

4. 「**FILE SYSTEM STATE NOT SET TO OKAY**」や「**FILE SYSTEM MODIFIED**」というメッセージが表示される場合は、必要に応じて `fsck` コマンドをもう一度実行してください。

`fsck` コマンドは、一度の実行ではすべてのエラーを修正できないことがあります。

`fsck` コマンドを何度か実行してもすべての問題を修復できない場合は、358 ページの「`fsck` コマンドで修復できない UFS ファイルシステムの修正」を参照してください。

5. 修復したファイルシステムをマウントして、`lost+found` ディレクトリにファイルが存在するかどうかを確認します。

`fsck` コマンドによって `lost+found` ディレクトリに入れられた各ファイルの名前は、その `i` ノード番号を使用して変更されます。可能であれば、ファイル名を変更し、ファイルが含まれるべき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内の語句を探したり、`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。

どうしても特定できないファイルまたはディレクトリについては、`lost+found` ディレクトリからそれらを削除して、不要な領域を解放してください。

6. `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。

### 例 20-1 ルート (/) 以外、または /usr 以外のファイルシステムを対話式でチェックする

次の例は、/dev/rdsk/c0t0d0s6 ファイルシステムをチェックし、不正なブロック数を訂正する方法を示しています。この例では、ファイルシステムがマウント解除されていることを前提としています。

```
# fsck /dev/rdsk/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/rdsk/c0t0d0s6 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
```

## UFS ファイルシステムの修復

fsck -o p コマンド (p は preen (修復) を表す) は、UFS ファイルシステムをチェックし、通常は予期しないシステムのシャットダウンによって発生する問題を自動的に修正します。オペレータの介入が必要な問題が発見されると、このコマンドは即座に終了します。このコマンドによって、ファイルシステムを並列にチェックすることも可能です。

システムがクリーンな状態でシャットダウンしなかった後のファイルシステムの修復にも、fsck -o p コマンドを実行することができます。このモードでは、fsck コマンドはクリーンフラグを調べずに完全チェックを実行します。これらの処理は、fsck コマンドを対話式で実行した場合の処理のサブセットです。

### ▼ UFS ファイルシステムを修復する方法

この手順では、ファイルシステムがマウント解除されているか、非アクティブであることを前提としています。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. UFS ファイルシステムをマウント解除します。

```
# umount /mount-point
```

3. 修復オプションを指定して UFS ファイルシステムをチェックします。

```
# fsck -o p /dev/rdsk/device-name
```

fsck コマンドの引数として */mount-point* または */dev/rdsk/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 20-2 UFS ファイルシステムを修復する

次の例は、*/export/home* ファイルシステムの修復方法を示します。

```
# fsck -o p /export/home
```

## fsck コマンドで修復できない UFS ファイルシステムの修正

fsck コマンドを何度か実行しているとき、問題の修正を進めていく過程で、以前の実行で検出されていた問題が再度発生することがあります。このような場合には、問題が報告されなくなるまで fsck コマンドを繰り返し実行して、すべてのエラーを検出および修正する必要があります。fsck コマンドはすべての問題を修正するまで動作を続けるわけではないので、手作業で再実行しなければなりません。

fsck コマンドで表示される情報に注目してください。問題を解決する上で参考になります。たとえば、メッセージは損傷したディレクトリを指している場合があります。そのディレクトリを削除すると、fsck コマンドが問題なく実行されるようになる場合もあります。

それでも fsck コマンドでファイルシステムを修復できない場合は、ff、clri、および ncheck コマンドを使用し、間違いを指定して修正します。これらのコマンドの使用方法については、fsdb(1M)、ff(1M)、clri(1M)、ncheck(1M) の各マニュアルページを参照してください。最終的には、ファイルシステムを作成し直し、その内容をバックアップメディアから復元せざるを得ない場合があります。

ファイルシステム全体を復元する方法については、[第 25 章](#)を参照してください。

ファイルシステムを完全に修復できないが、読み取り専用としてマウントできる場合は、cp、tar、または cpio コマンドを使用して、データのすべてまたは一部をファイルシステムから取り出してください。

問題の原因がハードウェア上のディスクエラーであれば、ファイルシステムを作成し直して復元する前に、ディスクをフォーマットし直して再びスライスに分割しなければならない場合があります。ディスクデバイスを交換する前に、デバイスのケーブルおよびコネクタが正常に機能するかどうかをチェックしてください。一般に、ハードウェアエラーが発生すると、さまざまなコマンドで同じエラーが繰り返し表示されます。format コマンドはディスク上の不良ブロックを使用しないようにします。ただ

し、ディスクの損傷が致命的な場合、フォーマットし直した後も問題が解決されないことがあります。format コマンドの使用方法については、format (1M) のマニュアルページを参照してください。新しいディスクのインストール方法については、第 12 章または 第 13 章を参照してください。

---

## 不正なスーパーブロックの復元

ファイルシステムのスーパーブロック内のデータが破壊された場合は、復元しなければなりません。スーパーブロックが不正なときには、fsck コマンドからメッセージが表示されます。幸い、スーパーブロックのコピーがファイルシステム内に格納されています。fsck -o b コマンドを使用すると、スーパーブロックをそのいずれかのコピーで置き換えることができます。

スーパーブロックの詳細については、372 ページの「スーパーブロック」を参照してください。

ルート (/) ファイルシステム内のスーパーブロックが損傷し、修復できない場合は、次のどちらかの操作を実行します。

- システムをインストールし直す。
- ネットワークまたはローカルの CD からブートし、以下の手順を実行する。それらの手順で失敗する場合は、newfs コマンドを使ってルート (/) ファイルシステムを作成し直し、バックアップコピーから復元する。

### ▼ 不正なスーパーブロックを復元する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 不正なスーパーブロックがルート (/) または /usr ファイルシステム内にあるかどうかを調べ、次のどちらかの操作を実行します。
    - a. 不正なスーパーブロックがルート (/) または /usr ファイルシステム内にある場合は、システムをいったん停止し、ネットワークまたはローカル接続された CD からブートします。

ローカル接続された CD からブートする場合は、次のコマンドを使用します。

```
ok boot cdrom -s
```

ブートサーバーまたはインストールサーバーがすでに設定済みのネットワークからブートする場合は、次のコマンドを使用します。

```
ok boot net -s
```

システムを停止する方法については、『Solaris のシステム管理 (基本編)』の第 10 章「システムのブート (手順)」または『Solaris のシステム管理 (基本編)』の第 11 章「システムのブート (手順)」を参照してください。

- b. 不正なスーパーブロックがルート (/) または /usr ファイルシステム内にある場合は、損傷したファイルシステム以外のディレクトリに移動し、ファイルシステムをマウント解除します。

```
# umount /mount-point
```



---

注意 - 次の手順では、必ず `newfs -N` オプションを使用してください。-N オプションを指定しない場合は、そのファイルシステムのデータがすべて破壊され、空のファイルシステムに置き換わります。

---

3. `newfs -N` コマンドを使用して、スーパーブロックの値を表示します。

```
# newfs -N /dev/rdsk/device-name
```

このコマンドの出力には、`newfs` コマンドによってファイルシステムが作成されたときに、スーパーブロックのコピーとして使用されることになったブロック番号が表示されます。カスタマイズされたファイルシステムを作成する方法については、375 ページの「カスタムファイルシステムパラメータ」を参照してください。

4. `fsck` コマンドを使用して、代替スーパーブロックを指定します。

```
# fsck -F ufs -o b=block-number /dev/rdsk/device-name
```

`fsck` コマンドは、指定された代替スーパーブロックを使用して、一次スーパーブロックを復元します。いつでも代替ブロックとして 32 を試すことができます。また、`newfs -N` コマンドで表示された代替ブロックを使用することもできます。

### 例 20-3 不正なスーパーブロックを復元する

次の例は、スーパーブロックのコピー 5264 を復元する方法を示しています。

```
# newfs -N /dev/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,
93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
140528, 145760, 150992, 156224, 161456,
# fsck -F ufs -o b=5264 /dev/rdsk/c0t3d0s7
Alternate superblock location: 5264.
** /dev/rdsk/c0t3d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
```



```
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)
/dev/rdisk/c0t3d0s7 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
#
```

---

## fsck コマンドの構文とオプション

fsck コマンドは、ファイルシステム内の不整合をチェックして修復します。オプションを指定しないで fsck コマンドを実行した場合は、修復が行われる前に確認を求めるプロンプトが表示されます。このコマンドには、次の4つのオプションがあります。

| コマンドとオプション | 説明                                                                         |
|------------|----------------------------------------------------------------------------|
| fsck -m    | ファイルシステムがマウント可能かどうかのチェックを行う                                                |
| fsck -y    | すべての修復に対して yes の応答が指定されたものとして処理を行う                                         |
| fsck -n    | すべての修復に対して no の応答が指定されたものとして処理を行う                                          |
| fsck -o p  | 確認を求めるプロンプトを表示することなく、ファイルシステムを修復し、想定される(軽微な)不整合箇所をすべて修正するが、重大な問題が見つかりと終了する |



## 第 21 章

# UFS ファイルシステム (参照情報)

この章の内容は次のとおりです。

- 363 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」
- 371 ページの「プラットフォームに依存するディレクトリ」
- 372 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」
- 375 ページの「カスタムファイルシステムパラメータ」
- 379 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 21-3 を参照してください。

次の表に、ルート (/) ファイルシステムに含まれているデフォルトのディレクトリの説明を示します。

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ

| ディレクトリ   | 説明                                |
|----------|-----------------------------------|
| /        | ファイルシステムの名前空間全体のルート               |
| /dev     | 論理デバイスファイルの一次位置                   |
| /dev/cfg | 物理 <code>ap_id</code> へのシンボリックリンク |

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ        | 説明                                                           |
|---------------|--------------------------------------------------------------|
| /dev/cua      | uucp 用のデバイスファイル                                              |
| /dev/dsk      | ブロックディスクデバイス                                                 |
| /dev/fbs      | フレームバッファのデバイスファイル                                            |
| /dev/fd       | ファイル記述子                                                      |
| /dev/md       | ボリューム管理デバイス名                                                 |
| /dev/printers | USB プリンタデバイスファイル                                             |
| /dev/pty      | pty スレーブデバイス                                                 |
| /dev/rdisk    | raw ディスクデバイス                                                 |
| /dev/rmt      | raw テープデバイス                                                  |
| /dev/sad      | STREAMS Administrative Driver のエントリポイント                      |
| /dev/sound    | オーディオデバイスとオーディオデバイス制御ファイル                                    |
| /dev/swap     | デフォルトのスワップデバイス                                               |
| /dev/term     | シリアルデバイス                                                     |
| /devices      | 物理デバイスファイル                                                   |
| /etc          | ホスト固有のシステム管理構成ファイルとデータベース                                    |
| /etc/acct     | アカウントの構成情報                                                   |
| /etc/apache   | Apache の構成ファイル                                               |
| /etc/cron.d   | cron の構成情報                                                   |
| /etc/default  | 各種プログラムのデフォルト情報                                              |
| /etc/dfs      | エクスポートされるファイルシステムの構成情報                                       |
| /etc/dhcp     | DHCP (Dynamic Host Configuration Protocol) の構成ファイル           |
| /etc/dmi      | Solstice Enterprise Agents の構成ファイル                           |
| /etc/fn       | フェデレーテッドネーミングサービスと x.500 のサポートファイル                           |
| /etc/fs       | ファイルシステムタイプ別に編成されたバイナリ                                       |
| /etc/ftpd     | ftpd の構成ファイル                                                 |
| /etc/gss      | GSS (Generic Security Service) アプリケーションプログラミングインタフェースの構成ファイル |

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ       | 説明                                                   |
|--------------|------------------------------------------------------|
| /etc/gtk     | GNOME (GNU Network Object Model Environment) の構成ファイル |
| /etc/inet    | インターネットサービスの構成ファイル                                   |
| /etc/init.d  | 実行レベルを変更するためのスクリプト                                   |
| /etc/iplanet | iPlanet (Sun ONE) の構成ファイル                            |
| /etc/krb5    | Kerberos の構成ファイル                                     |
| /etc/lib     | /usr が利用できないときに必要な動的リンクライブラリ                         |
| /etc/l1c2    | 論理リンク制御 (l1c2) ドライバの構成ファイル                           |
| /etc/lp      | プリンタサブシステムの構成情報                                      |
| /etc/lu      | Solaris Live Upgrade の構成ファイル                         |
| /etc/lvm     | Solaris ボリュームマネージャの構成ファイル                            |
| /etc/mail    | メールサブシステムの構成情報                                       |
| /etc/nca     | Solaris NCA (Network Cache and Accelerator) の構成ファイル  |
| /etc/net     | TI (トランスポート独立) ネットワークサービスの構成情報                       |
| /etc/nfs     | NFS サーバーロギングの構成ファイル                                  |
| /etc/openwin | OpenWindows の構成ファイル                                  |
| /etc/opt     | オプションパッケージの構成情報                                      |
| /etc/ppp     | Solaris PPP の構成ファイル                                  |
| /etc/rc0.d   | 実行レベル 0 を開始または停止したときに起動されるスクリプト                      |
| /etc/rc1.d   | 実行レベル 1 を開始または停止したときに起動されるスクリプト                      |
| /etc/rc2.d   | 実行レベル 2 を開始または停止したときに起動されるスクリプト                      |
| /etc/rc3.d   | 実行レベル 3 を開始または停止したときに起動されるスクリプト                      |
| /etc/rcS.d   | システムをシングルユーザーモードにするためのスクリプト                          |
| /etc/rcm     | 再構成マネージャ (RCM) のカスタムスクリプト用のディレクトリ                    |
| /etc/rpcsec  | NIS+ 認証の構成ファイルが含まれていることがある                           |

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ         | 説明                                                                                                                                                            |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /etc/saf       | サービスアクセス機能ファイル (FIFO など)                                                                                                                                      |
| /etc/security  | BSM (Basic Security Module) の構成ファイル                                                                                                                           |
| /etc/sfw       | Samba の構成ファイル                                                                                                                                                 |
| /etc/skel      | 新規ユーザーアカウントのデフォルトプロファイルスクリプト                                                                                                                                  |
| /etc/smartcard | Solaris SmartCards の構成ファイル                                                                                                                                    |
| /etc/snmp      | Solstice Enterprise Agents の構成ファイル                                                                                                                            |
| /etc/ssh       | ssh (secure shell) の構成ファイル                                                                                                                                    |
| /etc/sysevent  | syseventd の構成ファイル                                                                                                                                             |
| /etc/tm        | 商標ファイル。内容はブート時に表示される                                                                                                                                          |
| /etc/usb       | USB の構成情報                                                                                                                                                     |
| /etc/uucp      | uucp 構成情報                                                                                                                                                     |
| /etc/wrsm      | WRSM (WCI Remote Shared Memory) の構成情報                                                                                                                         |
| /export        | 共有ファイルシステム (ユーザーのホームディレクトリやクライアントファイルシステムなど) 用のデフォルトのディレクトリ                                                                                                   |
| /home          | スタンドアロンシステム上にあるユーザーのホームディレクトリ用のデフォルトのディレクトリまたはマウントポイント。AutoFS の動作中、このディレクトリには新しいエンタリを作成できない                                                                   |
| /kernel        | プラットフォームに依存しない読み込み可能なカーネルモジュールのディレクトリ。ブートプロセスの一部として必要。プラットフォームに依存しないコアカーネル /kernel/genunix の汎用部分を含む。/platform ディレクトリと /usr/platform ディレクトリの構造については、表 21-3 を参照 |
| /mnt           | ファイルシステムの一般的な一次マウントポイント                                                                                                                                       |
| /opt           | 追加アプリケーションパッケージ用のデフォルトディレクトリまたはマウントポイント                                                                                                                       |
| /platform      | サポートされているプラットフォームのファイル。詳細については、表 21-3 を参照                                                                                                                     |
| /proc          | プロセス情報                                                                                                                                                        |
| /sbin          | ブートプロセスと手作業によるシステム障害の回復に使用される重要な実行可能プログラム                                                                                                                     |

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ        | 説明                                                         |
|---------------|------------------------------------------------------------|
| /tmp          | 一時ファイル。内容はブートシーケンス中に消去される                                  |
| /usr          | /usr ファイルシステムのマウントポイント。詳細については、表 21-2 を参照                  |
| /var          | 常に変化するファイル (一時ファイル、ログファイル、状態ファイルなど) 用のディレクトリ               |
| /var/adm      | システムのログファイルとアカウントングファイル                                    |
| /var/apache   | Apache Web サーバー用のスクリプト、アイコン、ログ、キャッシュページ                    |
| /var/audit    | BSM (Basic Security Module) の監査ファイル                        |
| /var/crash    | カーネルクラッシュダンプのデフォルトの格納場所                                    |
| /var/cron     | cron のログファイル                                               |
| /var/dmi      | Solstice Enterprise Agents のデスクトップ管理インタフェースの実行時構成要素        |
| /var/dt       | dtlogin の構成ファイル                                            |
| /var/inet     | IPv6 ルーターの状態ファイル                                           |
| /var/krb5     | Kerberos のデータベースとログファイル                                    |
| /var/ld       | 実行時リンカーの構成ファイル                                             |
| /var/ldap     | LDAP クライアントの構成ファイル                                         |
| /var/log      | システムログファイル                                                 |
| /var/lp       | ラインプリンタサブシステムのログ情報                                         |
| /var/mail     | ユーザーのメールが保管されるディレクトリ                                       |
| /var/news     | コミュニティサービスメッセージ。これらのメッセージは USENET 方式のニュースとは異なる             |
| /var/nfs      | NFS サーバーのログファイル                                            |
| /var/nis      | NIS+ データベース                                                |
| /var/ntp      | NTP (Network Time Protocol) サーバーの状態ディレクトリ                  |
| /var/opt      | ソフトウェアパッケージ関連の各種ファイルのサブツリーのルート                             |
| /var/preserve | vi と ex のバックアップファイル                                        |
| /var/run      | 一時的な (つまり、システムをリポート後に残る必要がない) システムファイル。TMPFS マウントされたディレクトリ |

表 21-1 ルート (/) ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ                  | 説明                                    |
|-------------------------|---------------------------------------|
| /var/sadm               | ソフトウェアパッケージ管理ユーティリティで管理されるデータベース      |
| /var/saf                | saf (サービスアクセス機能) のログファイルとアカウントリングファイル |
| /var/samba              | Samba のログファイルとロックファイル                 |
| /var/snmp               | SNMP の状態と構成情報                         |
| /var/spool              | スプール化された一時ファイルのディレクトリ                 |
| /var/spool/clientmqueue | Sendmail のクライアントファイル                  |
| /var/spool/cron         | cron と at のスプールファイル                   |
| /var/spool/locks        | スプールロックファイル                           |
| /var/spool/lp           | ラインプリンタのスプールファイル                      |
| /var/spool/mqueue       | 配信用に待ち行列に入れられたメール                     |
| /var/spool/pkg          | スプール化されたパッケージ                         |
| /var/spool/print        | LP 印刷サービスのクライアント側要求格納域                |
| /var/spool/samba        | Samba の印刷待ち行列                         |
| /var/spool/uucp         | 待ち行列に入っている uucp のジョブ                  |
| /var/spool/uucppublic   | uucp によって格納されるファイル                    |
| /var/statmon            | ネットワーク状態監視ファイル                        |
| /var/tmp                | 一時ファイルのディレクトリ。ブートシーケンス中には消去されない       |
| /var/uucp               | uucp のログファイルと状態ファイル                   |
| /var/yp                 | NIS データベース                            |

次の表に、/usr ファイルシステムに含まれているデフォルトのディレクトリの説明を示します。

表 21-2 /usr ファイルシステムのデフォルトディレクトリ

| ディレクトリ | 説明                             |
|--------|--------------------------------|
| 4lib   | SunOS 4.1 バイナリ互換パッケージライブラリ     |
| 5bin   | /usr/bin ディレクトリへのシンボリックリンク     |
| x      | /usr/openwin ディレクトリへのシンボリックリンク |



表 21-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ  | 説明                                                                    |
|---------|-----------------------------------------------------------------------|
| adm     | /var/adm ディレクトリへのシンボリックリンク                                            |
| apache  | Apache の実行可能プログラム、ロード可能モジュール、マニュアル                                    |
| aset    | ASET (Automated Security Enhancement Tools) のプログラムとファイル用のディレクトリ       |
| bin     | 標準的なシステムコマンド用のディレクトリ                                                  |
| ccs     | C 言語処理系のプログラムとライブラリ                                                   |
| demo    | デモのプログラムとデータ                                                          |
| dict    | /usr/share/lib/dict ディレクトリへのシンボリックリンク。UNIX の spell プログラムが使用する辞書が入っている |
| dt      | CDE ソフトウェア用のディレクトリまたはマウントポイント                                         |
| games   | 空のディレクトリ。SunOS 4.0-4.1 ソフトウェアで使用されていた                                 |
| include | C プログラム用などのヘッダーファイル                                                   |
| iplanet | Directory Server の実行可能プログラム、ロード可能モジュール、マニュアル                          |
| j2se    | Java 2 SDK の実行可能プログラム、ロード可能モジュール、マニュアル                                |
| java*   | Java のプログラムとライブラリが入っているディレクトリ                                         |
| kernel  | その他のカーネルモジュール                                                         |
| kvm     | 廃止または互換性がなくなる可能性あり                                                    |
| lib     | 各種プログラムのライブラリ、アーキテクチャ依存データベース、またはユーザーが直接呼び出さないバイナリ                    |
| local   | サイトのローカルコマンド                                                          |
| mail    | /var/mail ディレクトリへのシンボリックリンク                                           |
| man     | /usr/share/man ディレクトリへのシンボリックリンク                                      |
| net     | ネットワークリスナーサービス用のディレクトリ                                                |
| news    | /var/news ディレクトリへのシンボリックリンク                                           |
| oasys   | FMLI (Form and Menu Language Interpreter) 実行環境用のファイル                  |
| old     | 段階的に使用されなくなっているプログラム                                                  |

表 21-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ                 | 説明                                                   |
|------------------------|------------------------------------------------------|
| openwin                | OpenWindows ソフトウェアのディレクトリまたはマウントポイント                 |
| perl5                  | perl 5 のプログラムとマニュアル                                  |
| platform               | サポートされているプラットフォームのファイル。詳細については、表 21-3 を参照            |
| preserve               | /var/preserve ディレクトリへのシンボリックリンク                      |
| proc                   | proc ツール用のディレクトリ                                     |
| pub                    | オンラインマニュアルページと文字処理用のファイル                             |
| sadm                   | システム管理に関連する各種ファイルとディレクトリ                             |
| sbin                   | システム管理用の実行可能プログラム                                    |
| sbin/install.d         | JumpStart のカスタムのスクリプトと実行可能プログラム                      |
| sbin/static            | /usr/bin と /usr/sbin から選択したプログラムの静的リンクバージョン          |
| sbin/sparcv7 と sparcv9 | SPARC システムの 32 ビットバージョンと 64 ビットバージョンのコマンド            |
| sbin/i86               | x86 アーキテクチャ固有のコマンド                                   |
| sfw                    | GNU と公開されているソースの実行可能プログラム、ライブラリ、マニュアル                |
| share                  | アーキテクチャに依存しない共有可能ファイル                                |
| share/admserv5.1       | iPlanet Console and Administration Server 5.0 のマニュアル |
| share/audio            | オーディオファイルのサンプル                                       |
| share/ds5              | Sun ONE Directory Server 5.1 のマニュアル                  |
| share/lib              | アーキテクチャに依存しないデータベース                                  |
| share/man              | Solaris のマニュアルページ                                    |
| share/src              | カーネル、ライブラリ、ユーティリティのソースコード                            |
| snadm                  | システム管理とネットワーク管理に関するプログラムとライブラリ                       |
| spool                  | /var/spool ディレクトリへのシンボリックリンク                         |
| src                    | /usr/share/src ディレクトリへのシンボリックリンク                     |
| tmp                    | /usr/var/tmp ディレクトリへのシンボリックリンク                       |
| ucb                    | UCB 互換パッケージのバイナリ                                     |

表 21-2 /usr ファイルシステムのデフォルトディレクトリ (続き)

| ディレクトリ     | 説明                                                     |
|------------|--------------------------------------------------------|
| ucbinclude | UCB 互換パッケージのヘッダーファイル                                   |
| ucbplib    | UCB 互換パッケージのライブラリ                                      |
| 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

次の表に、/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 21-3 /platform と /usr/platform ディレクトリ

| ディレクトリ               | 説明                                                                                                                           |
|----------------------|------------------------------------------------------------------------------------------------------------------------------|
| /platform            | ルート (/) ファイルシステムに存在すべき一連のディレクトリが、サポートされるプラットフォームごとに 1 ディレクトリずつ入っている。                                                         |
| /platform/*/kernel   | プラットフォームに依存するカーネル構成要素が入っている。プラットフォームに依存するコアカーネルであるファイル <code>unix</code> も含む。詳細については、 <code>kernel (1M)</code> のマニュアルページを参照。 |
| /usr/platform        | ルート (/) ファイルシステムに存在する必要がない、プラットフォームに依存するオブジェクトが入っている。                                                                        |
| /usr/platform/*/lib  | /usr/lib ディレクトリ中のオブジェクトに類似した、プラットフォームに依存するオブジェクトが入っている。                                                                      |
| /usr/platform/*/sbin | /usr/sbin ディレクトリ中のオブジェクトに類似した、プラットフォームに依存するオブジェクトが入っている。                                                                     |

---

## UFS ファイルシステムのシリンダグループの構造

UFS ファイルシステムを作成すると、ディスクスライスは、1 つまたは複数の連続するディスクシリンダから構成される「シリンダグループ」に分割されます。シリンダグループはさらにアドレス指定可能なブロックに分割され、このブロックによって、シリンダグループ内のファイルの構造が編成され、制御されます。各種のブロックは、ファイルシステム内で特定の機能を持っています。UFS ファイルシステムには、次の 4 種類のブロックがあります。

| ブロックの種類       | 格納されている情報の種類        |
|---------------|---------------------|
| ブートブロック       | システムのブート時に使用される情報   |
| スーパーブロック      | ファイルシステムに関する大部分の情報  |
| i ノード         | ファイルに関する名前以外のすべての情報 |
| 記憶域またはデータブロック | 各ファイルのデータ           |

次の節では、これらのブロックの編成と機能について説明します。

### ブートブロック

ブートブロックには、システムのブート時に使用されるオブジェクトが格納されます。ファイルシステムがブートに使用されなければ、ブートブロックは空白のままです。ブートブロックは最初のシリンダグループ (シリンダグループ 0) にのみ置かれ、スライス内の最初の 8K バイトです。

### スーパーブロック

スーパーブロックには、次のようなファイルシステムに関する大部分の情報が格納されます。

- ファイルシステムのサイズと状態
- ラベル。ファイルシステム名とボリューム名を含む
- ファイルシステムの論理ブロックのサイズ
- 最終更新日時
- シリンダグループのサイズ
- シリンダグループ内のデータブロック数
- 集計データブロック

- ファイルシステムの状態
- 最後のマウントポイントのパス名

スーパーブロックには重要なデータが入っているので、ファイルシステムの作成時には複数のスーパーブロックが作成されます。

集計情報ブロックは、スーパーブロック内に保管されます。複製されませんが、通常はシリンダグループ 0 内で最初のスーパーブロックといっしょにグループ化されます。集計ブロックには、ファイルシステムの使用時に発生した変化が記録されます。さらに、ファイルシステム内の i ノード数、ディレクトリ数、フラグメント数、および記憶ブロック数が表示されます。

## i ノード

i ノードには、ファイルに関して名前以外のすべての情報が入っており、ディレクトリ内に保管されます。i ノードは 128 バイトです。i ノード情報はシリンダ情報ブロック内に保管され、次の情報が入っています。

- ファイルのタイプ
  - 通常ファイル
  - ディレクトリ
  - ブロック型特殊ファイル
  - キャラクタ型特殊ファイル
  - FIFO (名前付きパイプとも呼ぶ)
  - シンボリックリンク
  - ソケット
  - その他の i ノード - 属性ディレクトリとシャドウファイル (ACL 用)
- ファイルのモード (読み取り権 - 書き込み権 - 実行権のセット)
- ファイルへのハードリンク数
- ファイルの所有者のユーザー ID
- ファイルが属するグループ ID
- ファイル内のバイト数
- 15 個のディスクブロックアドレスの配列
- ファイルの最終アクセス日時
- ファイルの最終変更日時
- ファイルの作成日時

15 個のディスクアドレス (0 から 14 まで) の配列は、ファイルの内容が格納されるデータブロックを指します。最初の 12 個は直接アドレスです。つまり、ファイルの内容のうち最初の 12 個の論理記憶ブロックを直接指します。ファイルが論理ブロック 12 個分より大きい場合は、13 番目のアドレスは間接ブロックを指します。間接ブロックには、ファイルの内容ではなく直接ブロックのアドレスが入っています。14 番目のアドレスは、二重間接ブロックを指します。二重間接ブロックには、間接ブロックのアドレスが入っています。15 番目のアドレスは三重間接アドレス用です。次の図に、i ノードから始まるこのアドレスブロックチェーンを示します。

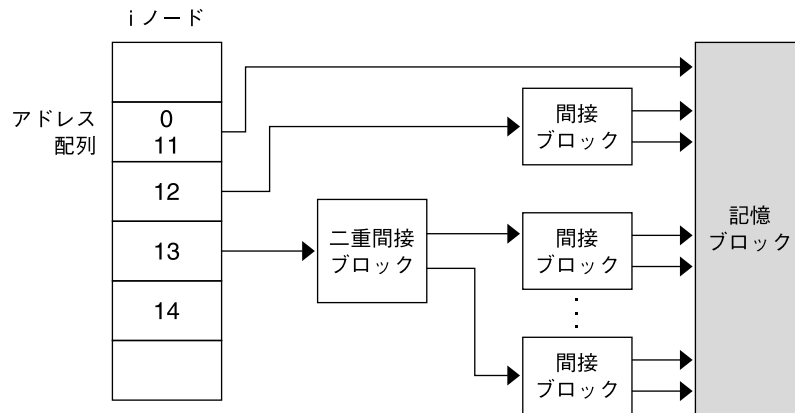


図 21-1 UFS ファイルシステムのアドレスチェーン

## データブロック

ファイルシステムに割り当てられた残りの領域には、データブロック (記憶ブロックともいう) が入っています。これらのデータブロックのサイズは、ファイルシステムの作成時に決定されます。デフォルトでは、データブロックは2つのサイズ、つまり8K バイトの論理ブロックサイズと1K バイトのフラグメントサイズで割り当てられます。

通常ファイルの場合、データブロックにはファイルの内容が入っています。ディレクトリの場合、データブロックにはディレクトリ内のファイルの i ノード番号とファイル名を示すエントリが入っています。

## 空きブロック

現在、i ノード、間接アドレスブロック、または記憶ブロックとして使用されていないブロックには、シリンダグループマップ内で空きを示すマークが付けられます。また、このマップはフラグメントを追跡し、断片化によるディスクパフォーマンスの低下を防止します。

UFS ファイルシステムの内容の概念を理解しやすいように、次の図に、一般的な UFS システム内の一連のシリンダグループを示します。

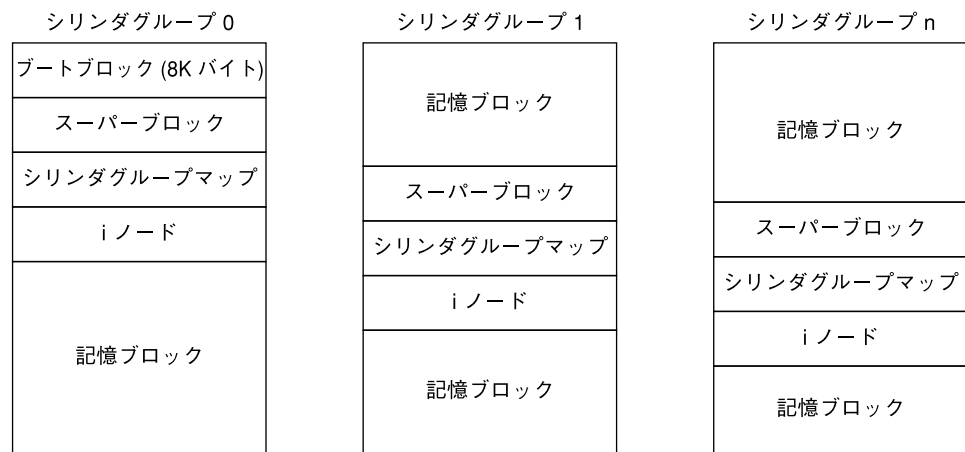


図 21-2 一般的な UFS ファイルシステム

## カスタムファイルシステムパラメータ

`newfs` コマンドによって割り当てられるデフォルトのファイルシステムパラメータを変更しようとする前に、各パラメータについて理解しておく必要があります。この節では、次の各パラメータについて説明します。

- 論理ブロックサイズ
- フラグメントサイズ
- 最小空き領域
- 回転待ち
- 最適化タイプ
- i ノード数 (ファイルの数)

### 論理ブロックサイズ

論理ブロックサイズは、UNIX カーネルがファイルの読み書きに使用するブロックのサイズです。一般に、論理ブロックサイズは物理ブロックサイズとは異なります。物理ブロックサイズは、通常は 512K バイトで、ディスクコントローラが読み書きできる最小ブロックのサイズです。

論理ブロックサイズは、デフォルトでシステムのページサイズに設定されます。UFS ファイルシステムの場合、このデフォルト論理ブロックサイズは 8192 バイト (8K バイト) です。UFS ファイルシステムでは、ブロックサイズとして 4096 バイトまたは 8192 バイト (4K または 8K バイト) がサポートされます。論理ブロックの推奨サイズは 8K バイトです。

---

**SPARC** のみ - sun4u プラットフォームで指定できるブロックサイズは 8192 バイトだけです。

---

システムに最善の論理ブロックサイズを選択するには、必要なパフォーマンスと使用可能容量を検討してください。ほとんどの UFS システムでは、8K バイトのファイルシステムが最高のパフォーマンスを発揮し、ディスクパフォーマンスと一次メモリーやディスク上の領域の使用量が適切なバランスに保たれます。

原則として、効率を高めるには、ほとんどのファイルがきわめて大きいファイルシステムには大きめの論理ブロックサイズを使用します。ほとんどのファイルがきわめて小さいファイルシステムには、小さめの論理ブロックサイズを使用します。ファイルシステム上で `quot -c file-system` コマンドを使用すると、ファイルの分散に関する詳細なレポートをブロックサイズ別に表示できます。

ただし、通常は、ファイルシステムの作成時に設定されたページサイズが最適です。

## フラグメントサイズ

ファイルが作成または拡張されると、論理ブロック全体または「フラグメント」と呼ばれる部分のディスク容量が割り当てられます。ファイルのためにディスク容量が必要になると、まずブロック全体が割り当てられ、次に残りの部分にブロックのうち 1 つまたは複数のフラグメントが割り当てられます。小型ファイルの場合、割り当てはフラグメントから始まります。

ブロック全体ではなく、そのフラグメントを割り当てることができるので、ブロック内の未使用のホールによって生じるディスク容量の「断片化」が減少し、容量の節約になります。

UFS ファイルシステムを作成するときに、「フラグメントサイズ」を定義します。デフォルトのフラグメントサイズは 1K バイトです。各ブロックは、1 個、2 個、4 個、または 8 個のフラグメントに分割できます。この場合、フラグメントサイズは 8192 バイトから 512 バイト (4K バイトのファイルシステムのみ) までです。実際には、下限はディスクのセクターサイズ、通常は 512 バイトに連動します。

マルチテラバイトのファイルシステムの場合、フラグメントサイズはファイルシステムのブロックサイズに等しくなります。

---

注 - フラグメントサイズの上限は論理ブロックサイズに等しくなります。この場合、フラグメントは存在しないこととなります。容量よりも速度を重視する場合、きわめて大型のファイルがあるファイルシステムには、この構成が最適なことがあります。

---



フラグメントサイズを選択するときには、処理時間と容量をトレードオフしてください。フラグメントサイズが小さければ容量の節約になりますが、割り当てには時間がかかります。原則として、格納効率を高めるには、ほとんどのファイルが大型のファイルシステムには、大きめのフラグメントサイズを使用します。ほとんどのファイルが小型のファイルシステムには、小さめのフラグメントサイズを使用します。

## 最小空き容量

「最小空き容量」とは、ファイルシステムの作成時に予約分として保持されるディスク容量の割合です。デフォルトの予約分は、 $((64\text{M バイト}/\text{パーティションサイズ}) * 100)$  で算出し、その値はもっとも近い整数に切り捨てられ、ディスク容量の 1% から 10% の範囲に制限されます。

ファイルシステム内の空き容量が少なくなるほど、アクセス速度が低下するので、空き容量は重要です。十分な空き容量があれば、UFS ファイルシステムは効率よく動作します。ファイルシステムがいっぱいになって、使用可能なユーザー領域を使い果たすと、スーパーユーザー以外は予約済みの空き容量にアクセスできなくなります。

`df` などのコマンドは、最小空き容量として割り当て済みの分を差し引いて、ユーザーに使用可能な容量をパーセントで表示します。コマンドでファイルシステム内のディスク容量の 100 パーセント以上が使用中であると表示される場合は、予約分の一部がスーパーユーザーによって使用されています。

ユーザーに割り当てを適用する場合に、各ユーザーが使用可能な容量には予約分の空き容量は含まれません。`tunefs` コマンドを使用すると、既存のファイルシステムの最小空き容量の値を変更できます。

## 回転待ち

このパラメータはもう使用しません。入力値に関わらず、値は常に 0 に設定されます。

## 最適化のタイプ

「最適化のタイプ」パラメータの設定には、「*space*」と「*time*」があります。

- *space* – 最適化のタイプに *space* を選択すると、断片化を最小限度に抑え、ディスクの使用状況が最適化されるようにディスクブロックが割り当てられる。
- *time* – 最適化のタイプに *time* を選択すると、配置はあまり重視されず、できるだけ高速になるようにディスクブロックが割り当てられる。十分な空き領域があれば、それほど細かく断片化しなくても、比較的簡単にディスクブロックを効率よく割り当てることができる。デフォルトは *time*。

既存のファイルシステムの場合は、`tunefs` コマンドを使用して最適化タイプのパラメータ値を変更できる。

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

## i ノード数 (ファイルの数)

i ノード 1 個あたりのバイト数によって、ファイルシステム内における i ノードの密度が決まります。ファイルシステムの合計サイズをこの値で割ると、作成すべき i ノードの個数が得られます。i ノードが割り当てられたら、ファイルシステムを作成し直さないかぎり、その数は変更できません。

i ノード 1 個あたりのデフォルトのバイト数は 2048 バイト (2K バイト) で、これは各ファイルの平均サイズが 2K バイト以上であることを想定しています。ファイルシステムが 1G バイトを超える場合、次の公式が使用されます。

| ファイルシステムのサイズ                                 | i ノードごとのバイト数 |
|----------------------------------------------|--------------|
| 1G バイト以下                                     | 2048         |
| 2G バイト未満                                     | 4096         |
| 3G バイト未満                                     | 6144         |
| 3G バイト以上 1T バイト以下                            | 8192         |
| 1T バイト超または <code>-t</code> オプションを指定して作成された場合 | 1048576      |

多数のシンボリックリンクを持つファイルシステムでは、平均ファイルサイズが小さくなる場合があります。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。i ノード数が少ないために i ノードが不足するよりも、多すぎる方が好ましいことを留意してください。i ノード数が少なすぎると、実際には空のディスクスライス上でも最大ファイル数に達してしまうことがあります。

## UFS ファイルとファイルシステムの最大サイズ

UFS ファイルシステムの最大サイズは、およそ 16T バイトで、使用できる領域は、そこから約 1 パーセントのオーバーヘッドを引いたサイズです。「空白」ファイルの論理サイズは 1T バイトです。ただし、ファイルに格納できる実際のデータ容量は、1T バイトから約 1 パーセントのファイルシステムオーバーヘッドを引いたサイズです。

## UFS サブディレクトリの最大数

UFS ファイルシステム内の 1 ディレクトリあたりのサブディレクトリの最大数は 32,767 です。この制限はあらかじめ定義されたものであり、変更できません。

---

## カスタマイズされたファイルシステムを作成するためのコマンド

この節では、カスタマイズされたファイルシステムの作成に使用する次の2つのコマンドについて説明します。

- `newfs`
- `mkfs`

### `newfs` コマンドの構文、オプション、引数

`newfs` コマンドは、ファイルシステムの作成に使用する `mkfs` コマンドの簡便バージョンです。

構文は次のとおりです。

```
/usr/sbin/newfs [-NTv] [mkfs_options] raw_device
```

次の表に、`newfs` コマンドのオプションと引数を示します。

表 21-4 `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>-T</code>           | ファイルシステムのサイズが最終的に 1T バイトを超えることが可能となるように、ファイルシステムのパラメータが設定される。このオプションを指定すると、 <code>fragsize</code> が <code>bsize</code> と同じ値に設定され、 <code>-i</code> オプションによる指定がなければ、 <code>nbpi</code> は 1M バイトに設定される。 <code>-f</code> オプションまたは <code>-i</code> オプションを使って指定された <code>fragsize</code> 値または <code>nbpi</code> 値がこのオプションの値と矛盾した場合、指定された <code>fragsize</code> 値または <code>nbpi</code> 値は無視される。 |
| <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 ( <code>-a apc</code> から <code>-t ntrack</code> まで) を使用して <code>mkfs</code> コマンドのパラメータが設定される。各オプションは、空白で区切る。                                                                                                                                                                                                                                                                    |

表 21-4 newfs コマンドのオプションと引数 (続き)

| オプション               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a <i>apc</i>       | <p>不良ブロックを配置するために予約される 1 ディスクシリンダの代替セクター数 (SCSI デバイスのみ)。デフォルトは 0 (ゼロ)。</p> <p>このオプションは、EFI ラベルのディスクには適用されず、無視される。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| -b <i>bsize</i>     | <p>ファイルシステムの論理ブロックサイズ。4096 バイトまたは 8192 バイト。デフォルトは 8192 バイト。sun4u アーキテクチャでは、4096 バイトのブロックサイズはサポートされない。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| -c <i>cgsize</i>    | <p>1 シリンダグループあたりのシリンダ数。値の範囲は 16 から 256 まで。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。デフォルト値は、常に 16 から 256 までの値。</p> <p>デフォルト値以外の値を使用したい場合は <code>mkfs</code> コマンドを直接使用する。</p> <p>このオプションは、EFI ラベルのディスクには適用されず、無視される。</p>                                                                                                                                                                                                                                                                                                                                                                                       |
| -C <i>maxcontig</i> | <p>特定のファイルに属し、連続して割り当てられる論理ブロックの最大数。デフォルトは次のようにして計算される。</p> $\text{maxcontig} = \text{ディスクドライブの最大転送サイズ} / \text{ディスクブロックサイズ}$ <p>ディスクドライブの最大転送サイズが不明である場合、<code>maxcontig</code> のデフォルト値は次のようにして計算される。</p> <p><code>maxphys</code> が <code>ufs_maxmaxphys</code> (通常は 1M バイト) よりも小さい場合、<code>maxcontig</code> は <code>maxphys</code> と同じ値に設定される。そうでない場合、<code>maxcontig</code> は <code>ufs_maxmaxphys</code> と同じ値に設定される。</p> <p><code>maxcontig</code> には任意の正の整数を設定できる。</p> <p>実際の値は、指定された値とハードウェアがサポートする値の、いずれか小さいほうになる。</p> <p>このパラメータを後で変更するには、<code>tunefs</code> コマンドを使用する。詳細については、<code>tunefs(1M)</code> のマニュアルページを参照。</p> |
| -d <i>gap</i>       | <p>回転待ち。このオプションはもう使用しません。入力値に関わらず、値は常に 0 に設定されます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

表 21-4 newfs コマンドのオプションと引数 (続き)

| オプション              | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -f <i>fragsize</i> | <p>ファイルに割り当て可能な最小のディスク容量 (バイト数)。 <i>bsize</i> を <i>fragsize</i> で割った数は、2 の累乗でなければならない。</p> <p>ここで、 <i>bsize</i> / <i>fragsize</i> は 1、2、4、8 のいずれかである。</p> <p>つまり、 <i>fragsize</i> に指定できるのは、論理ブロックサイズが 4096 の場合は 512、1024、2048、4096 のいずれか、論理ブロックサイズが 8192 の場合は 1024、2048、4096、8192 のいずれかである。デフォルト値は 1024。</p> <p>1T バイトを超えるファイルシステム、または -T オプションを指定して作成されたファイルシステムの場合、 <i>fragsize</i> は強制的にブロックサイズ (<i>bsize</i>) と同じ値に設定される。</p> |
| -i <i>nbpi</i>     | <p>i ノード 1 個当たりのバイト数。この値はファイルシステム内における i ノードの密度を表す。この値でファイルシステムの合計サイズを割ると、作成すべき i ノードの個数が得られる。</p> <p>この値は、ファイルシステム内のファイルの予想平均サイズに応じたものにする必要がある。必要とされる i ノードが少ない場合は大きな値を指定し、作成すべき i ノードが多い場合は小さな値を指定する。</p> <p>詳細については、378 ページの「i ノード数 (ファイルの数)」を参照。</p>                                                                                                                                                                          |
| -m <i>free</i>     | <p>ファイルシステムで管理される空きディスク領域の最小許容率 (0% から 99%)。一般のユーザーはこの領域を使用できない。ファイルシステムがこのしきい値に達した場合、スーパーユーザーだけがファイルシステムへの書き込みを続行できる。</p> <p>デフォルトの予約分は、((64M バイト / パーティションサイズ) * 100) で算出した値はもっとも近い整数に切り捨てられ、ディスク容量の 1% から 10% の範囲に制限される。</p> <p>このパラメータは、ファイルシステムを作成した後で、<code>tune2fs</code> コマンドを使用して変更できる。</p>                                                                                                                              |
| -n <i>nrpos</i>    | <p>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。</p> <p>このオプションは、EFI ラベルのディスクには適用されず、無視される。</p>                                                                                                                                                                                                                                                                                                                                              |
| -o <i>opt</i>      | <p>最適化のタイプ (<code>space</code>、<code>time</code> のいずれか)。ブロックの割り当てにかかる時間を最小化する (<code>time</code>) か、領域の断片化を最小化する (<code>space</code>) ように、ファイルシステムに指示できる。デフォルトは <code>time</code>。</p>                                                                                                                                                                                                                                            |

表 21-4 newfs コマンドのオプションと引数 (続き)

| オプション      | 説明                                                                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------|
| -r rpm     | 1分当たりのディスクの回転数。この設定はドライバまたはデバイスに固有である。<br><br>このパラメータは、mkfs コマンドに渡される前に1秒当たりの回転数に変換される。<br><br>このオプションは、EFI ラベルのディスクには適用されず、無視される。 |
| -s size    | ファイルシステムのセクター数。デフォルトでは、パーティション全体が使用される。                                                                                            |
| -t ntrack  | 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取得される。<br><br>このオプションは、EFI ラベルのディスクには適用されず、無視される。                                              |
| raw_device | /dev ディレクトリ内に格納された raw 特殊デバイスデバイス名 (/dev/rdisk/c0t0d0s6 など)。ここで指定されたデバイス上に、ファイルシステムが作成される。この引数は必須。                                 |

例 21-1 newfs コマンドのオプションと引数

次の例は、-N オプションを使用して、バックアップスーパーブロックなどのファイルシステム情報を表示する方法を示しています。

```
# newfs -N /dev/rdisk/c0t0d0s0
/dev/rdisk/c0t0d0s0: 37260 sectors in 115 cylinders of 9 tracks, 36 sectors
                    19.1MB in 8 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
superblock backups (for fsck -b #) at:
    32, 5264, 10496, 15728, 20960, 26192, 31424, 36656,
#
```

## 汎用 mkfs コマンド

汎用 mkfs コマンドは、ファイルシステム専用の mkfs コマンドを呼び出して、指定したディスクスライス上で指定したタイプのファイルシステムを作成させます。mkfs コマンドは各種のファイルシステムに対応していますが、実際には UFS、UDFS、PCFS の各ファイルシステムの作成に使用します。他のタイプのファイルシステムを作成するには、ファイルシステム専用の mkfs コマンドのソフトウェアを作成する必要があります。通常、mkfs コマンドは直接実行しません。mkfs コマンドは、newfs コマンドによって呼び出されます。

汎用 mkfs コマンドは、/usr/sbin ディレクトリに入っています。引数とオプションについては、mkfs (1M) のマニュアルページを参照してください。

## 第 22 章

---

# ファイルシステムのバックアップと復元 (概要)

---

この章では、`ufsdump` コマンドと `ufsrestore` コマンドを使用して、ファイルシステムのバックアップおよび復元を実行する際のガイドラインと計画の作成について説明します。

この章の内容は次のとおりです。

- 383 ページの「バックアップと復元についての参照先」
- 384 ページの「ファイルシステムのバックアップと復元とは」
- 385 ページの「ファイルシステムをバックアップする理由」
- 385 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」
- 388 ページの「バックアップタイプの選択」
- 388 ページの「テープデバイスの選択」
- 389 ページの「ファイルシステムのバックアップおよび復元の概要 (作業マップ)」
- 390 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」
- 394 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

| バックアップまたは復元処理                                 | 参照先    |
|-----------------------------------------------|--------|
| <code>ufsdump</code> コマンドを使用したファイルシステムのバックアップ | 第 23 章 |
| <code>fssnap</code> コマンドを使用した UFS スナップショットの作成 | 第 24 章 |
| <code>ufsrestore</code> コマンドを使用したファイルシステムの復元  | 第 25 章 |

---

cpio、dd、pax、および cpio コマンドを使用したファイルおよびディレクトリのコピー [第 27 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リムーバブルメディア (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リムーバブルメディアから作業ディレクトリにコピーすることを意味します。

この章では、UFS ファイルシステムのバックアップおよび復元に使用する `ufsdump` および `ufsrestore` コマンドについて説明します。他のコマンドを使用してファイルやファイルシステムをコピーし、ファイルの共有や移動を行うこともできます。次の表に、個々のファイルやファイルシステムを他のメディアにコピーするすべてのコマンドを示します。

表 22-1 ファイルとファイルシステムのバックアップおよび復元用コマンド

| 作業                                             | コマンド                                                            | 参照先                                                       |
|------------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------|
| 1 つまたは複数のファイルシステムをローカルまたはリモートのテープデバイスにバックアップする | <code>ufsdump</code>                                            | <a href="#">第 23 章</a> または <a href="#">第 26 章</a>         |
| ファイルシステムの読み取り専用コピーを作成する                        | <code>fssnap</code>                                             | <a href="#">第 24 章</a>                                    |
| ネットワーク上のシステムのファイルシステム全体をバックアップサーバーからバックアップする   | Solstice Backup ソフトウェア                                          | 『 <i>Solstice Backup 5.1 管理者ガイド</i> 』                     |
| NIS+ マスターサーバーをバックアップ、復元する                      | <code>nisbackup</code> および <code>nisrestore</code> コマンド         | 『 <i>Solaris のシステム管理 (ネーミングとディレクトリサービス: FNS、NIS+ 編)</i> 』 |
| テープまたはフロッピーディスク上でファイルをコピー、表示、検索する              | <code>tar</code> 、 <code>cpio</code> 、または <code>pax</code> コマンド | <a href="#">第 27 章</a>                                    |
| マスターディスクをクローンディスクにコピーする                        | <code>dd</code>                                                 | <a href="#">第 27 章</a>                                    |



表 22-1 ファイルとファイルシステムのバックアップおよび復元用コマンド (続き)

| 作業                                              | 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を、リムーバブルメディアから作業ディレクトリに復元する | ufsrestore コマンド | 第 25 章 |

## ファイルシステムをバックアップする理由

ファイルのバックアップは、もっとも重要なシステム管理作業の 1 つです。次のような問題によるデータの消失に備えて、定期的にバックアップを実行する必要があります。

- システムのクラッシュ
- 不注意によるファイルの削除
- ハードウェア障害
- 天災 (火災、台風、地震など)
- システムの再インストールやアップグレード時に発生する問題

## バックアップを作成するファイルシステムの計画

頻繁に更新されるファイルシステムなど、ユーザーにとって重要なファイルシステムはバックアップしておく必要があります。次の表に、スタンドアロンシステムとサーバー用にバックアップを作成するファイルシステムの一般的なガイドラインを示します。

表 22-2 ファイルシステムをスタンドアロン用にバックアップする

| バックアップするファイルシステム           | 説明                                                                                                         | バックアップ間隔                             |
|----------------------------|------------------------------------------------------------------------------------------------------------|--------------------------------------|
| ルート (/) - スライス 0           | このファイルシステムにはカーネルが格納されているが、/var ディレクトリが格納されていることもある。/var ディレクトリには、メールやアカウントングファイルなど、頻繁に変更されるファイルが保管される。     | 定期的 (毎週、毎日など)                        |
| /usr - スライス 6、/opt         | 一般に、新しいソフトウェアおよび新しいコマンドをインストールすると、/usr と /opt ファイルシステムが影響を受ける。/opt ディレクトリは、ルート (/) の一部であるか、独自のファイルシステムである。 | 随時                                   |
| /export/home - スライス 7      | このファイルシステムには、スタンドアロンシステム上のユーザー全員のディレクトリおよびサブディレクトリが含まれる。                                                   | ルート (/) や /usr よりも頻繁に、サイトのニーズによっては毎日 |
| /export、/var、または他のファイルシステム | Solaris ソフトウェアのインストール中に、これらのファイルシステムを作成した可能性がある。                                                           | サイトの必要に応じて                           |

表 22-3 ファイルシステムをサーバー用にバックアップする

| バックアップするファイルシステム       | 説明                                                           | バックアップ間隔                                                                                                                                                                                                                                                                                                               |
|------------------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ルート (/) - スライス 0       | このファイルシステムには、カーネルおよび実行可能プログラムが含まれる。                          | <p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>ネットワーク上でユーザーとシステムの追加および削除を頻繁に実行する場合、このファイルシステム内の構成ファイルを変更する必要がある。この場合、週に 1 度から月に 1 度の間隔で、ルート (/) ファイルシステムの完全バックアップを実行する必要がある。</p> <p>サイトでユーザーのメールをメールサーバー上の /var/mail ディレクトリに保管しておき、後でクライアントシステムがマウントする場合は、ルート (/) ディレクトリを毎日バックアップした方がよい。別のファイルシステムの場合は、/var を毎日バックアップした方がよい。</p> |
| /export - スライス 3       | このファイルシステムには、ディスクレスクライアント用のカーネルおよび実行可能プログラムが含まれる。            | <p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>このファイルシステム内の情報はスライス 0 にあるサーバーのルートディレクトリと同様なので、ファイルシステムは頻繁に変化することはない。サイトからメールをクライアントシステムに送信していない場合、このファイルシステムのバックアップは随時実行すればよい。送信している場合は、/export のバックアップをさらに頻繁に実行する必要がある。</p>                                                                                                      |
| /usr - スライス 6、<br>/opt | オプションのファイルシステムで、一般的にシステムソフトウェア以外のソフトウェアを格納するために使用される。        | <p>サイトの必要に応じて毎日 1 度ないし月に 1 度。</p> <p>通常、これらのファイルシステムは静的であるため、週または月に 1 度バックアップするだけでかまわない。</p>                                                                                                                                                                                                                           |
| /export/home - スライス 7  | このファイルシステムには、システムの全ユーザーのディレクトリが含まれる。このファイルシステム内のファイルは、変更が多い。 | 毎日または毎週。                                                                                                                                                                                                                                                                                                               |

---

## バックアップタイプの選択

ufsdump コマンドを使用すると、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。fssnap コマンドを使用すると、ファイルシステムの一時イメージを作成できます。次の表に、各バックアップタイプの手順の違いを示します。

表 22-4 バックアップタイプの相違点

| バックアップのタイプ | 結果                                             | 長所                        | 短所                                                                                                                |
|------------|------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------|
| 完全 (フル)    | ファイルシステムやディレクトリ全体をコマンドをコピーする。                  | すべてのデータが1カ所に格納される。        | 大量のバックアップテープが必要であり、書き込みに時間がかかる。ドライブはテープ上でファイルが入っている位置に順番に移動しなければならないので、個々のファイルの検索時間が長くなる。複数のテープを検索しなければならない場合もある。 |
| スナップショット   | ファイルシステムの一時的イメージを作成する。                         | システムはマルチユーザモードで動作可能。      | スナップショットの作成中は、システムのパフォーマンスが低下する可能性がある。                                                                            |
| 増分         | 指定されたファイルシステム内で、前回のバックアップ以降に変更されたファイルのみをコピーする。 | ファイルシステム内の小さな変化を簡単に検索できる。 | どの増分テープにファイルが入っているかを探すのに時間がかかることがある。最後の完全バックアップに戻らなければならない場合もある。                                                  |

---

## テープデバイスの選択

次の表に、バックアップ処理中にファイルシステムを格納するための典型的なテープデバイスを示します。記憶容量は、ドライブのタイプとテープに書き込むデータのタイプによって異なります。テープデバイスの詳細については、[第 28 章](#)を参照してください。

表 22-5 ファイルシステムのバックアップに使用する一般的なメディア

| バックアップメディア                        | 記憶容量               |
|-----------------------------------|--------------------|
| 1/2 インチのリールテープ                    | 140M バイト (6250bpi) |
| 2.5G バイト、1/4 インチのカートリッジ (QIC) テープ | 2.5G バイト           |
| DDS3 4mm カートリッジテープ (DAT)          | 12 - 24G バイト       |
| 14G バイト、8mm カートリッジテープ             | 14G バイト            |
| DLT 7000 1/2 インチ カートリッジテープ        | 35 - 70G バイト       |

## ファイルシステムのバックアップおよび復元の概要 (作業マップ)

この作業マップを使用して、ファイルシステムのバックアップおよび復元作業をすべて確認できます。各作業は、実行するバックアップタイプの判別などの一連の追加作業を指し示します。

| 作業                     | 説明                                                                            | 参照先                              |
|------------------------|-------------------------------------------------------------------------------|----------------------------------|
| 1. バックアップするファイルシステムの識別 | 毎日、毎週、または毎月、バックアップの必要なファイルシステムを識別します。                                         | 385 ページの「バックアップを作成するファイルシステムの計画」 |
| 2. バックアップタイプの決定        | サイトに適したファイルシステムのバックアップタイプを決定します。                                              | 388 ページの「バックアップタイプの選択」           |
| 3. バックアップの作成           | 次のいずれかを実行します。                                                                 |                                  |
|                        | ファイルシステムの完全バックアップおよび増分バックアップを実行する場合は、 <code>ufsdump</code> コマンドを使用します。        | 第 23 章                           |
|                        | アクティブかつマウント中のファイルシステムのスナップショットを作成する場合は、 <code>fssnap</code> コマンドの使用を考慮してください。 | 第 24 章                           |
|                        | アクティブかつマウント中のファイルシステムのスナップショットを作成する場合は、 <code>fssnap</code> コマンドの使用を考慮してください。 | 第 27 章                           |

| 作業                                     | 説明                                                                                                        | 参照先                              |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------|----------------------------------|
| 4. (省略可能) ファイルシステムの復元                  | ファイルまたはファイルシステムのバックアップに使用するコマンドに基づいて、復元方法を選択します。                                                          |                                  |
|                                        | ufsdump コマンドを使用して作成したファイルシステムのバックアップを復元します。                                                               | 第 25 章                           |
|                                        | tar、cpio、または pax コマンドを使用して作成したファイルシステムを復元します。                                                             | 第 27 章                           |
| 5. (省略可能) ルート (/) または /usr ファイルシステムの復元 | ルート (/) または /usr ファイルシステムの復元は、非クリティカルなファイルシステムの復元よりも複雑です。これらのファイルシステムの復元中にローカル CD またはネットワークから起動する必要があります。 | 432 ページの「ルート (/) と /usr を復元する方法」 |

## バックアップスケジュールを設定する際のガイドライン

「バックアップスケジュール」とは、ufsdump コマンドを実行するように設定するスケジュールです。この節では、バックアップスケジュールの作成時に検討すべき要素に関するガイドラインを説明します。また、バックアップスケジュールの例もいくつか紹介します。

バックアップスケジュールは、以下の点を考慮に入れて作成します。

- バックアップに使用するテープの数を最小限に抑える
- バックアップの実行に使用できる時間
- 損傷したファイルシステムの完全復元に使用できる時間
- 不注意に削除した個々のファイルの検索に使用できる時間

### バックアップ頻度

バックアップに費やす時間とメディアの数を最小限度に抑える必要がない場合は、完全バックアップを毎日実行してもかまいません。しかし、多くのサイトの場合、このようなバックアップ方法は現実的ではないので、ほとんどの場合は増分バックアップが使用されます。その場合は、サイトが過去 4 週間分のバックアップからファイルを

十分復元できるようにしてください。そのためには、少なくとも1週分ごとに1組ずつ、合計4組のテープが必要です。各組のテープを毎月使い回すことになります。また、少なくとも1年分の月別のバックアップを保存する必要があります。そして、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。

## バックアップの用語と定義

次の表に、バックアップの用語および定義を示します。

| 用語       | 定義                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| スナップショット | ファイルシステムの一時イメージを作成する。                                                                                                                                                                                                                                                                                                                                                               |
| 完全バックアップ | ファイルシステムまたはディレクトリの完全なコピーを作成する。                                                                                                                                                                                                                                                                                                                                                      |
| 増分バックアップ | 指定したファイルシステムのファイルのうち、前回のバックアップ以降に変更されたファイルだけをコピーする。増分バックアップには、次の種類がある。 <ul style="list-style-type: none"><li>■ 日単位累積 - 月曜日に1日分のファイル変更をコピーする。次に、月曜日のバックアップを火曜日、水曜日のように、各曜日のファイル変更で上書きしていく。</li><li>■ 日単位増分 - 1日分のファイル変更をコピーする。つまり、月曜日、火曜日のように、各曜日の変更の差分テープを記録していく。</li><li>■ 週単位累積 - その週に変更されたファイルをコピーする。前の週のファイル変更も残しておく。</li><li>■ 週単位増分 - 前の週のバックアップ以降に変更されたファイルをコピーする。</li></ul> |

## バックアップスケジュールに関するその他の推奨事項

次の表に、バックアップスケジュールに関するその他の推奨事項を示します。

表 22-6 バックアップスケジュールに関するその他の推奨事項

| 必要なファイル復元操作                                    | バックアップ間隔                                                                                                            | コメント                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 各ファイルの別バージョン (ワード処理に使用するファイルシステムなど) を復元する必要がある | 作業日ごとに日単位増分バックアップを実行する<br><br>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、毎日異なるテープを使用する必要がある。このように運用しないと、必要なバージョンのファイルを復元できなくなることがあるためである。<br><br>たとえば、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。ユーザーが火曜日のバージョンを必要とする場合には、火曜日 (または水曜日) のバックアップテープを保存しておかなければ、それを復元できない。また、火曜日と水曜日には存在したファイルが木曜日に削除されても、金曜日の下のレベルのバックアップには表示されない。 |
| ファイルシステム全体を短時間で復元する必要がある                       | 下位のレベルのバックアップを頻繁に実行する                                                                                               | —                                                                                                                                                                                                                                                                                                                                                                                     |
| 同じサーバー上で多数のファイルシステムのバックアップを作成する                | ファイルシステムごとにスケジュールをずらすことを検討する                                                                                        | この方法では、すべてのレベル 0 のバックアップを同じ日に実行しないことになる                                                                                                                                                                                                                                                                                                                                               |
| テープの本数を最小限に抑える                                 | 1 週間に実行する増分バックアップのレベルを上げる<br><br>週末に実行するバックアップのレベルを上げる。日単位と週単位の増分バックアップを同じテープに入れる<br><br>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 | 毎日の変更のみが各日単位テープに保存される<br><br>変更は (月単位ではなく) 週単位でしか週単位テープに保存されない<br><br>この操作には、 <code>ufsdump</code> の <code>'no rewind'</code> オプションを使用する。たとえば、 <code>/dev/rmt/0n</code> のように指定する                                                                                                                                                                                                      |



## ダンプレベルを使用して増分バックアップを作成する

ufsdump コマンドで指定するダンプレベル (0-9) によって、どのファイルのバックアップが作成されるかが決まります。ダンプレベル 0 を指定すると、完全バックアップが作成されます。増分バックアップのスケジュール設定にレベル 1 から 9 までの番号が使用されますが、特に意味が定義されているわけではありません。レベル 1 から 9 は、累積バックアップまたは個別バックアップのスケジュール設定に使用する番号の範囲にすぎません。レベル 1 から 9 までが意味するのは、大小による相互関係だけです。下位ダンプレベル番号を使用した場合は、常に完全バックアップまたは累積バックアップが再開されます。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。

### 例—日単位累積バックアップのダンプレベル

日単位の累積増分バックアップを実行するのが、もっとも一般に使用されるバックアップスケジュールで、ほとんどの場合に推奨できます。次の例で、月曜日から木曜日までレベル 9 のダンプを使用し、金曜日にはレベル 5 のダンプを使用してプロセスを再開するスケジュールを示します。

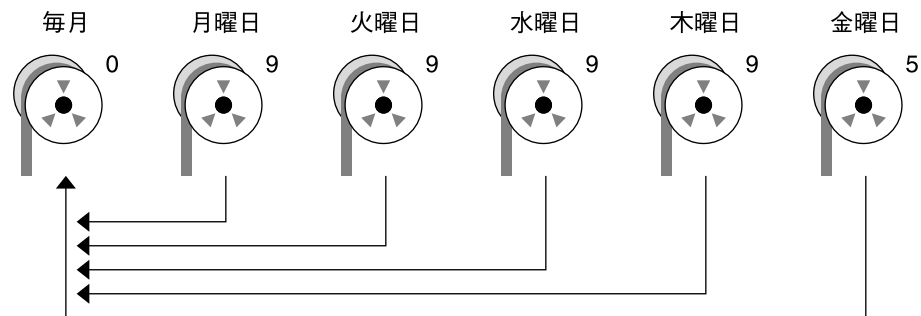


図 22-1 増分バックアップ: 日単位累積

前の例では、1 から 9 までの範囲内で他の番号を使用しても同じ結果が得られます。ポイントは、月曜日から木曜日まで同じ番号を使用し、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。

### 例—日単位増分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。このようなバックアップは、日単位増分バックアップと呼ばれます。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。金曜日はレベル番号が小さいので、プロセスが再開されます。

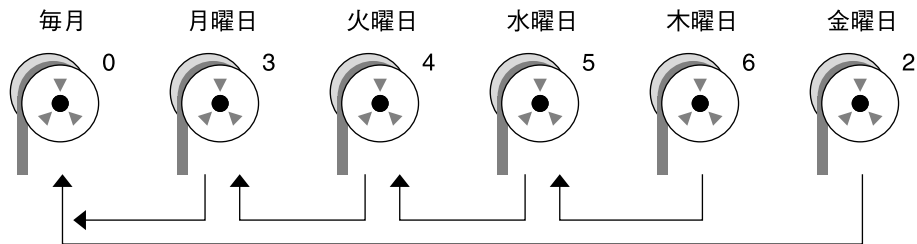


図 22-2 増分バックアップ: 日単位増分

前の例では、連番 6、7、8、9 の次に 2 を使用しても、5、6、7、8 の次に 3 を使用してもかまいません。番号自体の意味は定義されていません。この例が示しているように、連続した番号を並べることによって、それらの番号に意味が生まれます。

## バックアップスケジュールの例

この節では、バックアップスケジュールのサンプルを示します。どのスケジュールも、完全バックアップ(ダンプレベル 0)から始めることと、`-u` オプションを使用して `/etc/dumpdates` ファイルに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 22-7 に、もっとも一般的に使用される増分バックアップスケジュールを示します。これは、ほとんどの場合に推奨できるスケジュールです。このスケジュールで実行される処理は次のとおりです。

- 前週の終わりの下位レベルのバックアップ以降に変更があったすべてのファイルが毎日保存される。
- 月～金のレベル 9 のバックアップの場合は、直前のレベル 0 またはレベル 5 のバックアップがもっと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 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累積される。
- 毎週金曜日のレベル 5 のバックアップの場合、もっとも近い下位レベルのバックアップは、月初めに実行されたレベル 0 のバックアップである。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 22-7 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 |

次の表に、日単位累積、週単位累積スケジュールを使用して、テープの内容が2週間でのどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 22-8 日単位累積、週単位累積スケジュールのテープの内容

|     | 月   | 火     | 水       | 木         | 金                     |
|-----|-----|-------|---------|-----------|-----------------------|
| 第1週 | a b | a b c | a b c d | a b c d e | a b c d e f           |
| 第2週 | g   | g h   | g h i   | g h i j   | a b c d e f g h i j k |

## 日単位累積、週単位累積スケジュールのテープ要件

このスケジュールでは、日単位テープを再利用したい場合には、6本のテープが必要になります。ただし、曜日ごとに4本の日単位テープを別々に使用する場合は、次に示すように9本のテープが必要になります。

- レベル0のバックアップ用に1本
- 金曜日に4本
- 日単位テープ用に1本または4本

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。

- レベル0用に1本
- 前週の金曜日に1本
- 前週の金曜日以降の最新の日単位テープ1本

## 例 — 日単位累積、週単位増分バックアップ

次の表に、各曜日のテープに月曜日(第1週の場合は初期レベル0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累積されるスケジュールを示します。また、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ります。

表 22-9 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 |

次の表に、日単位累積、週単位増分バックアップスケジュールの場合、テープの内容が2週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 22-10 日単位累積、週単位増分バックアップスケジュールのテープの内容

|     | 月   | 火     | 水       | 木         | 金           |
|-----|-----|-------|---------|-----------|-------------|
| 第1週 | a b | a b c | a b c d | a b c d e | a b c d e f |
| 第2週 | g   | g h   | g h i   | g h i j   | g h i j k   |

## 日単位累積、週単位増分バックアップスケジュールのテープ要件

このスケジュールでは、日単位テープを再利用したい場合には、6本のテープが必要になります。ただし、曜日ごとに4本の日単位テープを別々に使用する場合は、次に示すように9本のテープが必要になります。

- レベル0のバックアップ用に1本
- 金曜日に4本
- 日単位テープ用に1本または4本

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。

- レベル0用に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

## 例 — 日単位増分、週単位累積バックアップ

次の表に、各曜日のテープには前日以降に変更があったファイルのみが入るスケジュールを示します。また、すべての金曜日のテープには、月初めの初期レベル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入ります。

表 22-11 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1週   |     | 3 | 4 | 5 | 6 | 2 |
| 第2週   |     | 3 | 4 | 5 | 6 | 2 |
| 第3週   |     | 3 | 4 | 5 | 6 | 2 |
| 第4週   |     | 3 | 4 | 5 | 6 | 2 |

次の表に、日単位増分、週単位累積スケジュールの場合、テープの内容が2週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 22-12 日単位増分、週単位累積バックアップスケジュールのテープの内容

|     | 月     | 火   | 水     | 木   | 金                                        |
|-----|-------|-----|-------|-----|------------------------------------------|
| 第1週 | a b   | c d | e f g | h i | a b c d e f g h i                        |
| 第2週 | j k l | m   | n o   | p q | a b c d e f g h i j k<br>l m n o p q r s |

## 日単位増分、週単位累積スケジュールのテープ要件

このスケジュールでは、少なくとも9本のテープが必要になります。これは日単位テープを毎週再利用することが前提となっていますが、これはお勧めできません。週単位テープを1カ月保存する場合は、21本のテープが必要になります。可能であれば、こちらの方法をお勧めします。内訳は、レベル0に1本、金曜日用に4本、日単位テープ4本または16本です。

- レベル0のバックアップ用に1本
- 金曜日のバックアップ用に4本
- 日単位のテープを4本または16本

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。

- レベル0用に1本
- 前週の金曜日のテープ
- 前週の金曜日以降のその週のすべての日単位テープ

## 例—サーバーの月単位バックアップスケジュール

次の表に、ユーザーがプログラム開発や文書作成のようなファイル集約型の作業を実行する小型ネットワーク上の、使用頻度の高いファイルサーバーのバックアップ方針の例を示します。この例は、バックアップ期間が日曜日に始まり、1週7日間を4週間行うものと想定しています。

表 22-13 例—サーバーの月単位バックアップスケジュール

| ディレクトリ       | 日付      | ダンプレベル | テープ名           |
|--------------|---------|--------|----------------|
| ルート (/)      | 第 1 日曜日 | 0      | <i>n</i> 本のテープ |
| /usr         | 第 1 日曜日 | 0      | <i>n</i> 本のテープ |
| /export      | 第 1 日曜日 | 0      | <i>n</i> 本のテープ |
| /export/home | 第 1 日曜日 | 0      | <i>n</i> 本のテープ |
|              | 第 1 月曜日 | 9      | A              |
|              | 第 1 火曜日 | 9      | B              |
|              | 第 1 水曜日 | 5      | C              |
|              | 第 1 木曜日 | 9      | D              |
|              | 第 1 金曜日 | 9      | E              |
|              | 第 1 土曜日 | 5      | F              |
| ルート (/)      | 第 2 日曜日 | 0      | <i>n</i> 本のテープ |
| /usr         | 第 2 日曜日 | 0      | <i>n</i> 本のテープ |
| /export      | 第 2 日曜日 | 0      | <i>n</i> 本のテープ |
| /export/home | 第 2 日曜日 | 0      | <i>n</i> 本のテープ |
|              | 第 2 月曜日 | 9      | G              |
|              | 第 2 火曜日 | 9      | H              |
|              | 第 2 水曜日 | 5      | I              |
|              | 第 2 木曜日 | 9      | J              |
|              | 第 2 金曜日 | 9      | K              |
|              | 第 2 土曜日 | 5      | L              |
| ルート (/)      | 第 3 日曜日 | 0      | <i>n</i> 本のテープ |
| /usr         | 第 3 日曜日 | 0      | <i>n</i> 本のテープ |
| /export      | 第 3 日曜日 | 0      | <i>n</i> 本のテープ |
| /export/home | 第 3 日曜日 | 0      | <i>n</i> 本のテープ |
|              | 第 3 月曜日 | 9      | M              |
|              | 第 3 火曜日 | 9      | N              |
|              | 第 3 水曜日 | 5      | O              |
|              | 第 3 木曜日 | 9      | P              |

表 22-13 例—サーバーの月単位バックアップスケジュール (続き)

| ディレクトリ       | 日付      | ダンプレベル | テープ名      |
|--------------|---------|--------|-----------|
|              | 第 3 金曜日 | 9      | Q         |
|              | 第 3 土曜日 | 5      | R         |
| ルート (/)      | 第 4 日曜日 | 0      | $n$ 本のテープ |
| /usr         | 第 4 日曜日 | 0      | $n$ 本のテープ |
| /export      | 第 4 日曜日 | 0      | $n$ 本のテープ |
| /export/home | 第 4 日曜日 | 0      | $n$ 本のテープ |
|              | 第 4 月曜日 | 9      | S         |
|              | 第 4 火曜日 | 9      | T         |
|              | 第 4 水曜日 | 5      | U         |
|              | 第 4 木曜日 | 9      | V         |
|              | 第 4 金曜日 | 9      | W         |
|              | 第 4 土曜日 | 5      | X         |

このスケジュールでは、 $4n$  本のテープ (ルート (/)、/usr、/export、および /export/home ファイルシステムの 4 回の完全バックアップに必要な本数) を使用します。また、/export/home ファイルシステムの増分バックアップ用に  $24n$  本のテープを使用します。このスケジュールは、増分バックアップごとに 1 本ずつテープを使用し、それを 1 カ月は保存することを前提としています。

このスケジュールの機能を、次に示します。

1. 日曜日ごとに、ルート (/)、/usr、/export、および /export/home ファイルシステムの完全バックアップ (レベル0) を実行します。レベル 0 のテープを少なくとも 3 カ月は保存します。
2. 月の第 1 月曜日に、テープ A を使用して /export/home ファイルシステムのレベル 9 のバックアップを実行します。ufsdump コマンドは、下のレベルのバックアップ以降に変更があったすべてのファイルをコピーします。この場合、直前の下位レベルのバックアップは、日曜日に実行したレベル 0 のバックアップになります。
3. 月の第 1 火曜日に、テープ B を使用して /export/home ファイルシステムのレベル 9 のバックアップを実行します。この場合も、ufsdump コマンドは、下のレベル (日曜日のレベル 0 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
4. 月の第 1 水曜日に、テープ C を使用して /export/home ファイルシステムのレベル 5 のバックアップを実行します。ufsdump コマンドは、日曜日以降に変更があったすべてのファイルをコピーします。

5. 木曜日と金曜日には、テープ D と E を使用して、`/export/home` ファイルシステムのレベル 9 のバックアップを実行します。`ufsdump` コマンドは、最後の下位レベルのバックアップ (水曜日のレベル 5 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
6. 月の第 1 土曜日に、テープ F を使用して `/export/home` のレベル 5 のバックアップを実行します。`ufsdump` コマンドは、下のレベルのバックアップ (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。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 のバックアップ用に  $4n$  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 のバックアップ用に新しいテープ 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。



## 第 23 章

# ファイルとファイルシステムのバックアップ (手順)

この章では、`ufsdump` コマンドを使用したファイルシステムのバックアップ手順について説明します。

手順の詳細については、401 ページの「ファイルとファイルシステムのバックアップ (作業マップ)」を参照してください。

バックアップの実行方法の概要については、第 22 章を参照してください。

`ufsdump` コマンドの構文、オプション、引数の詳細については、第 26 章を参照してください。

# ファイルとファイルシステムのバックアップ (作業マップ)

| 作業                                | 説明                                          | 参照先                             |
|-----------------------------------|---------------------------------------------|---------------------------------|
| 1. ファイルシステムのバックアップ準備を行う。          | バックアップで使用するファイルシステム、バックアップタイプ、テープデバイスを確認する。 | 402 ページの「ファイルシステムバックアップの実行準備」   |
| 2. ファイルシステムのバックアップに必要なテープの数を決定する。 | ファイルシステムの完全バックアップに必要なテープの数を決定する。            | 403 ページの「完全バックアップに必要なテープ数を判別する」 |

| 作業                    | 説明                                                                                                                       | 参照先                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| 3. ファイルシステムをバックアップする。 | <p>ファイルシステムの完全バックアップを実行して、全ファイルの基準コピーを取得する。</p> <p>日単位で変更されたファイルのコピーを保持することが、サイトで重要かどうかに基づいて、ファイルシステムの増分バックアップを実行する。</p> | 404 ページの「ファイルシステムのバックアップをテープに作成する方法」 |

## ファイルシステムバックアップの実行準備

ファイルシステムバックアップの準備は、計画を作成することから始まります。詳細については、[第 22 章](#)を参照してください。この段階では、次の項目を決定します。

- バックアップを作成するファイルシステム
- 実行するバックアップのタイプ (完全または増分)
- バックアップのスケジュール
- テープドライブ

詳細については、[第 22 章](#)を参照してください。

この節では、ファイルシステムのバックアップを作成する前に実行する必要がある、次の 2 つの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。
- 完全バックアップの作成に必要なテープの本数を決定する。

### ▼ ファイルシステム名を検索する方法

手順 1. `/etc/vfstab` ファイルの内容を表示します。

```
$ more /etc/vfstab
```

2. `mount point` 列に表示されるファイルシステム名を調べます。

3. ファイルシステムのバックアップを作成する際、`mount point` 列に表示されたディレクトリ名を使用します。

#### 例 23-1 ファイルシステム名を検索する

この例では、バックアップ対象のファイルシステムはルート (/)、`/usr`、`/datab`、および `/export/home` です。

```

$ more /etc/vfstab
#device          device          mount          FS   fsck mount  mount
#to mount        to fsck         point          type pass at boot options
#
fd                -                /dev/fd        fd   -    no    -
/proc            -                /proc          proc -    no    -
/dev/dsk/c0t0d0s1 -                -              swap -    no    -
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /              ufs  1    no    -
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /usr           ufs  1    no    -
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /datab        ufs  2    yes   -
/dev/dsk/c0t0d0s7 /dev/rdisk/c0t0d0s7 /export/home  ufs  2    yes   -
swap            -                /tmp           tmpfs -    yes   -

```

## ▼ 完全バックアップに必要なテープ数を判別する

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. バックアップのサイズをバイト単位で予測します。

```
# ufsdump S file-system
```

S オプションを指定すると、バックアップの実行に必要な予想バイト数が表示されます。

3. 予測サイズをテープの容量で割り、必要なテープの数を確認します。  
テープ容量のリストについては、表 22-5を参照してください。

### 例 23-2 テープの本数を決定する

次の例では、150M バイトのテープに 489472 バイトのファイルシステムが入ります。

```
# ufsdump S /export/home
489472
```

## ファイルシステムのバックアップ

バックアップを実行する際の一般的なガイドラインは次のとおりです。

- シングルユーザーモードを使用するか、ファイルシステムをマウント解除する (ファイルシステムのスナップショットを作成している場合を除く)。UFS スナップショットの詳細については、第 24 章を参照。
- ディレクトリレベルの処理 (ファイルの作成、削除、名前変更など) とファイルレベルの処理が同時に行われているときにファイルシステムのバックアップを実行すると、バックアップに組み込まれないデータがあるので注意する。

- 単一のシステムで `ufsdump` コマンドを実行し、リモートシェルまたはリモートログインを通じてネットワーク上のシステムグループのバックアップをリモート実行できる。また、出力をテープデバイスがあるシステムに転送できる。通常、テープデバイスは `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デバイスに作成するには、`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 27 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、テープデバイスを持つシステムの `/.rhosts` ファイル中には、ドライブを使用する各クライアントのエントリが入っていないなければならない。また、バックアップを実行する各システムの `/.rhosts` ファイルには、バックアップを開始するシステムのエントリが入っていないなければならない。
- システム上でリモートテープデバイスを指定するには、リモートテープデバイスを保持するシステムの OS バージョンと一致する命名規則を使用する。たとえば、SunOS 4.1.1 またはその互換バージョンが稼働中のシステム上のリモートデバイスにはデバイス `/dev/rst0` を使用する。また、Solaris 9 またはその互換バージョンを実行しているシステムでは、デバイス `/dev/rmt/0` を使用する。

---

注 - NIS+ マスターサーバーをバックアップする場合は、`nisbackup` コマンドを使用してください。このコマンドの使用方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : FNS、NIS+ 編)』を参照してください。

---

## ▼ ファイルシステムのバックアップをテープに作成する方法

`ufsdump` コマンドを使用してファイルシステムのバックアップを作成する一般的な手順を示します。この例では、オプションと引数の使用方法を示しています。

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. システムをレベル **S** (シングルユーザーモード) に移行します。

たとえば、次のようになります。

```
# shutdown -g30 -y
```

3. (省略可能) ファイルシステムの整合性をチェックします。

たとえば、次のようになります。

```
# fsck -m /dev/rdsk/c0t0d0s7
```

`fsck -m` コマンドを実行すると、ファイルシステムの整合性がチェックされます。たとえば、電源障害が発生すると、ファイルが不整合になることがあります。`fsck` コマンドの詳細については、第 20 章を参照してください。

4. ファイルシステムをリモートテープドライブにバックアップする場合は、次の手順を実行します。

- a. テープドライブが接続されているシステム (テープサーバー) の `./rhosts` ファイルに、次のエントリを追加します。

```
host root
```

`host` エントリには、`ufsdump` コマンドを実行してバックアップを行うシステムの名前を指定します。

- b. テープサーバー上で、上記の `./rhosts` ファイルに追加したホストに、ネームサービス経由でアクセスできることを確認します。
5. テープドライブのデバイス名を確認します。  
デフォルトのテープドライブは、`/dev/rmt/0` です。
6. 書き込み可能なテープをテープドライブに挿入します。
7. システムのバックアップを実行します。

```
# ufsdump options arguments filenames
```

ファイルシステムまたはファイルシステム内のディレクトリやファイルをバックアップできます。ファイルを個別にバックアップする方法については、`tar(1)` または `cpio(1)` のマニュアルページを参照してください。

次の例では、もっとも一般的な `ufsdump` オプションおよび引数の使用方法を示します。

- 例 23-3
- 例 23-4
- 例 23-5
- 例 23-6

その他の `ufsdump` オプションと引数については、[第 26 章](#)を参照してください。

8. プロンプトが表示されたら、テープを取り出して、次のテープを挿入します。
9. 各テープにボリューム番号、ダンプレベル、日付、システム名、ディスクスライス、ファイルシステム名を記入したラベルを貼ります。
10. **Control + D** キーを押してシステムをレベル 3 の動作に戻します。
11. バックアップが正常に実行されたことを確認します。

```
# ufsrestore tf device-name
```

### 例 23-3 ルート (/) の完全バックアップ

次の例では、ルート (/) ファイルシステムの完全バックアップを実行する方法を示します。この例では、バックアップの実行前にシステムをシングルユーザーモードにしています。次の `ufsdump` オプションが含まれます。

- 0 は、レベル 0 のダンプ (完全バックアップ) であることを示す。

- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示す。
- c は、カートリッジテープデバイスを示す。
- f /dev/rmt/0 は、テープデバイスを示す。
- / はバックアップするファイルシステムを示す。

たとえば、次のようになります。

```
# init 0
ok boot -s
# ufsdump 0ucf /dev/rmt/0 /
DUMP: Date of this level 0 dump: Tue Oct 07 16:23:08 2003
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: Writing 63 Kilobyte records
DUMP: Estimated 296644 blocks (144.85MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 296224 blocks (144.64MB) on 1 volume at 424 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue Oct 07 16:23:08 2003
# ufsrestore tf /dev/rmt/0
2      .
3      ./lost+found
3776   ./usr
7552   ./var
11328  ./export
15104  ./export/home
18880  ./etc
22656  ./etc/default
22657  ./etc/default/sys-suspend
22673  ./etc/default/cron
22674  ./etc/default/devfsadm
22675  ./etc/default/dhccpagent
22676  ./etc/default/fs
22677  ./etc/default/inetinit
22678  ./etc/default/kbd
22679  ./etc/default/mpathd
22680  ./etc/default/nfslogd
22681  ./etc/default/passwd
      .
      .
      .
# (Control + D を押して、システムをレベル3に戻す)
```

#### 例 23-4 ルート (/) の増分バックアップ

次の例は、シングルユーザーモードでルート (/) ファイルシステムの増分バックアップを実行する方法を示します。次の ufsdump オプションが含まれます。

- 9 は、レベル 9 のダンプ (増分バックアップ) であることを示す。
- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示す。
- c は、カートリッジテープデバイスを示す。
- f /dev/rmt/0 は、テープデバイスを示す。
- / はバックアップするファイルシステムを示す。

```
# init 0
ok boot -s
.
.
.
Rebooting with command: boot -sSunOS Release 5.9 Generic May 2002
Copyright 1983-2003 Sun Microsystems, Inc. All rights reserved.
.
.
.
# ufsdump 9ucf /dev/rmt/0 /
DUMP: Date of this level 9 dump: Mon Oct 06 12:36:10 2003
DUMP: Date of last level 0 dump: Wed Oct 08 10:12:13 2003
DUMP: Dumping /dev/rdisk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 335844 blocks (163.99MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 335410 blocks (163.77MB) on 1 volume at 893 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Mon Oct 06 12:36:10 2003
# 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 23-5 ホームディレクトリの完全バックアップ

次の例では、`/export/home/kryten` ホームディレクトリの完全バックアップを実行する方法を示します。次の `ufsdump` オプションが含まれます。

- `0` は、レベル `0` のダンプ (完全バックアップ) であることを示す。
- `u` は、このバックアップの実行日に `/etc/dumpdates` ファイルが更新されることを示す。
- `c` は、カートリッジテープデバイスを示す。
- `f /dev/rmt/0` は、テープデバイスを示す。
- `/export/home/kryten` は、バックアップするディレクトリを示す。

```
# umount /export/home
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
DUMP: Date of this level 0 dump: Tue Oct 07 08:41:41 2003
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: Writing 63 Kilobyte records
DUMP: Estimated 470 blocks (235KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 376 blocks (188KB) on 1 volume at 1205 KB/sec
DUMP: DUMP IS DONE
# ufsrestore tf /dev/rmt/0
      2      .
      5      ./export
      6      ./export/home
 80799      ./export/home/kryten
 80800      ./export/home/kryten/filea
 80801      ./export/home/kryten/fileb
 80802      ./export/home/kryten/filec
 80803      ./export/home/kryten/letters
 80804      ./export/home/kryten/letters/letter1
 80805      ./export/home/kryten/letters/letter2
 80806      ./export/home/kryten/letters/letter3
 80807      ./export/home/kryten/reports
 80808      ./export/home/kryten/reports/reportA
 80809      ./export/home/kryten/reports/reportB
 80810      ./export/home/kryten/reports/reportC
#
```

### 例 23-6 リモートシステムへの完全バックアップ (Solaris 9 データを Solaris 9 システムへ)

次の例では、Solaris 9 システム (starbug) のローカルの `/export/home` ファイルシステムを、シングルユーザーモードでリモートの Solaris 9 システム (earth) のテープデバイスに完全バックアップする方法を示します。次の `ufsdump` オプションが含まれます。



- 0 は、レベル 0 のダンプ (完全バックアップ) であることを示す。
- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示す。
- c は、カートリッジテープデバイスを示す。
- f earth:/dev/rmt/0 は、リモートシステム名およびテープデバイスを示す。
- /export/home は、バックアップするファイルシステムを示す。

```
# ufsdump 0ucf earth:/dev/rmt/0 /export/home
DUMP: Date of this level 0 dump: Mon Oct 06 12:46:50 2003
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s7 (starbug:/export/home) to
earth:/dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 410 blocks (205KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 376 blocks (188KB) on 1 volume at 546 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Oct 06 12:46:50 2003
# ufsrestore tf earth:/dev/rmt/0
2      .
3      ./lost+found
4      ./kryten
5      ./kryten/filea
6      ./kryten/fileb
7      ./kryten/filec
8      ./kryten/letters
9      ./kryten/letters/letter1
10     ./kryten/letters/letter2
11     ./kryten/letters/letter3
12     ./kryten/reports
.
.
.
#
```



## 第 24 章

# UFS スナップショットの使用 (手順)

この章では、UFS スナップショットの作成およびバックアップ方法について説明します。

UFS スナップショットの作成に関連した手順については、[411 ページの「UFS スナップショットの使用 \(作業マップ\)」](#)を参照してください。

バックアップの実行方法の概要については、[第 22 章](#)を参照してください。

## UFS スナップショットの使用 (作業マップ)

| 作業                           | 説明                                        | 参照先                              |
|------------------------------|-------------------------------------------|----------------------------------|
| 1. UFS スナップショットを作成する。        | fssnap コマンドを使用して、ファイルシステムの読み取り専用コピーを作成する。 | 414 ページの「UFS スナップショットを作成する方法」    |
| 2. UFS スナップショットの情報を表示する。     | UFS スナップショット情報 (raw スナップショットデバイスなど)を確認する。 | 415 ページの「UFS スナップショットの情報を表示する方法」 |
| 3. (省略可能) UFS スナップショットを削除する。 | バックアップ済みまたは不要になったスナップショットを削除する。           | 416 ページの「UFS スナップショットを削除する方法」    |
| 4. UFS スナップショットをバックアップする。    | 次の方法のいずれかを選択する。                           |                                  |

| 作業                                | 説明                                                      | 参照先                                              |
|-----------------------------------|---------------------------------------------------------|--------------------------------------------------|
|                                   | ufsdump コマンドを使用して、UFS スナップショットの完全バックアップを作成する。           | 417 ページの「UFS スナップショットの完全バックアップを作成する方法 (ufsdump)」 |
|                                   | ufsdump コマンドを使用して、UFS スナップショットの増分バックアップを作成する。           | 418 ページの「UFS スナップショットの増分バックアップの作成方法 (ufsdump)」   |
|                                   | tar コマンドを使用して、UFS スナップショットのバックアップを作成する。                 | 418 ページの「UFS スナップショットのバックアップ方法 (tar)」            |
| 5. (省略可能) UFS スナップショットからデータを復元する。 | ufsrestore コマンドを使用して、データを復元するのと同じ方法で、UFS スナップショットを復元する。 | 429 ページの「ファイルシステム全体を復元する方法」                      |

## UFS スナップショットの概要

Solaris リリースには、ファイルシステムのマウント中にファイルシステムをバックアップするための、`fssnap` コマンドが含まれています。`fssnap` コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。「スナップショット」は、バックアップ操作のためのファイルシステムの一時的イメージです。

`fssnap` コマンドを実行すると、1つの仮想デバイスと1つのバックアップストアファイルが作成されます。ユーザーは、既存の Solaris バックアップコマンドを使用して、実際のデバイスのように動作し実際のデバイスのように見える「仮想デバイス」をバックアップすることができます。「バックアップストア (backing-store)」ファイルは、スナップショット作成後に変更されたデータのコピーを含むビットマップファイルです。

## なぜ UFS スナップショットを使用するか

UFS スナップショット機能を使用することで、バックアップ時に、ファイルシステムをマウントされた状態にし、システムをマルチユーザーモードにしておくことができます。これまでは、バックアップを実行するために `ufsdump` コマンドを使用する時は、ファイルシステムをアクティブでない状態に保つためにシステムをシングルユーザーモードにすることが推奨されていました。より確実なバックアップのために、`tar` や `cpio` などの追加の Solaris のバックアップコマンドを使用して UFS スナップショットのバックアップを行うこともできます。

`fssnap` コマンドにより、企業レベルではないシステムの管理者が、大規模な記憶容量の必要なく、Sun StorEdge™ Instant Image のような企業レベルツールのパワーを得ることができます。

UFS スナップショットの機能は、Instant Image 製品の機能に似ています。UFS スナップショットは大規模なファイルシステムをコピーすることができますが、企業レベルのシステムには Instant Image の方が適しています。UFS スナップショットは、小さめのシステムに適しています。Instant Image は、取り込まれるファイルシステム全体のサイズに等しい容量を割り当てます。ただし、UFS スナップショットが作成するバックアップストアファイルは、必要なディスク容量だけを占有します。

次の表は、UFS スナップショットと Instant Image との特徴的な違いを示します。

| UFS スナップショット                                 | Sun StorEdge Instant Image                  |
|----------------------------------------------|---------------------------------------------|
| バックアップストアファイルのサイズは、スナップショットがとられた後のデータの変更量による | バックアップストアファイルのサイズは、コピーされるファイルシステム全体のサイズに等しい |
| システムのリポート後は保持されない                            | システムのリポート後も保持される                            |
| UFS ファイルシステムで動作する                            | ルート (/) または /usr ファイルシステムでは使用できない           |
| Solaris 8 1/01 以降のリリースで使用できる                 | Sun StorEdge 製品の一部                          |

## UFS スナップショットのパフォーマンス上の問題

UFS スナップショットの初回作成時に、ファイルシステムのユーザーが短い一時停止に気づく場合があります。一時停止の時間は、取り込まれるファイルシステムのサイズに応じて増加します。スナップショットがアクティブな間、ファイルシステムへの書き込み時にユーザーが若干のパフォーマンス低下に気づく場合がありますが、ファイルシステムからの読み込み時には影響はありません。

## UFS スナップショットの作成と削除

`fssnap` コマンドを使用して UFS スナップショットを作成する場合、バックアップストアファイルがどれだけのディスク容量を消費するかを監視してください。バックアップストアファイルは、はじめは容量をまったく使用しませんが、その後特によく使用されているシステムにおいて急速に拡大します。バックアップストアファイルが拡大するのに十分な容量を必ず確保してください。または `-o maxsize=n [k,m,g]` オプション ( $n [k,m,g]$  はバックアップストアファイル最大限のサイズ) でそのサイズを制限してください。



---

注意 - バックアップストアファイルに容量が不足する場合、スナップショットが削除されてしまうことがあり、バックアップが失敗します。スナップショットのエラーの可能性を調べるため、`/var/adm/messages` ファイルをチェックしてください。

---

詳細は、`fssnap_ufs(1M)` のマニュアルページを参照してください。

## ▼ UFS スナップショットを作成する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. ファイルシステムに、バックアップストアファイル用の十分なディスク容量が存在することを確認してください。

```
# df -k
```

3. 同じ場所に同じ名前の既存のバックアップストアファイルが存在していないことを確認します。

```
# ls /backing-store-file
```

4. UFS スナップショットを作成します。

```
# fssnap -F ufs -o bs=/backing-store-file /file-system
```

---

注 - バックアップストアファイルは、UFS スナップショットを使用して取り込むファイルシステムとは異なるファイルシステムに存在する必要があります。

---

5. スナップショットが作成されたことを確認します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

### 例 24-1 UFS スナップショットを作成する

次の例は、`/usr` ファイルシステムのスナップショットを作成する方法を示します。バックアップストアファイルは `/scratch/usr.back.file` です。仮想デバイスは `/dev/fssnap/1` です。

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr  
/dev/fssnap/1
```

次の例は、バックアップストアファイルを 500M バイトに制限する方法を示します。

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /export/home  
/dev/fssnap/1
```

## ▼ UFS スナップショットの情報を表示する方法

`fssnap -i` オプションを使用すると、システムの現在のスナップショットを表示することができます。1つのファイルシステムを指定する場合、そのスナップショットについての詳細な情報が表示されます。特定のファイルシステムを指定しない場合は、現在の UFS スナップショットすべておよび対応する仮想デバイスの情報が表示されます。

---

注 - 次の例に示すように、拡張スナップショット情報を表示する場合は、UFS ファイルシステム固有の `fssnap` コマンドを使用してください。

---

- 手順 1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 現在のすべてのスナップショットをリスト表示します。  
たとえば、次のようになります。

```
# /usr/lib/fs/ufs/fssnap -i
Snapshot number          : 0
Block Device             : /dev/fssnap/0
Raw Device                : /dev/rfssnap/0
Mount point              : /usr
Device state             : idle
Backing store path       : /var/tmp/snapshot3
Backing store size       : 256 KB
Maximum backing store size : Unlimited
Snapshot create time     : Wed Oct 08 10:38:25 2003
Copy-on-write granularity : 32 KB
Snapshot number          : 1
Block Device             : /dev/fssnap/1
Raw Device                : /dev/rfssnap/1
Mount point              : /
Device state             : idle
Backing store path       : /tmp/bs.home
Backing store size       : 448 KB
Maximum backing store size : Unlimited
Snapshot create time     : Wed Oct 08 10:39:29 2003
Copy-on-write granularity : 32 KB
```

3. 特定のスナップショットについての詳細な情報を表示します。  
たとえば、次のようになります。

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number          : 0
Block Device             : /dev/fssnap/0
Raw Device                : /dev/rfssnap/0
Mount point              : /usr
Device state             : idle
Backing store path       : /var/tmp/snapshot3
Backing store size       : 256 KB
Maximum backing store size : Unlimited
```

```
Snapshot create time      : Wed Oct 08 10:38:25 2003
Copy-on-write granularity : 32 KB
```

## UFS スナップショットの削除

UFS スナップショットを作成する際、バックングストアファイルがリンクされないように指定することができます。リンクされていないバックングストアファイルは、スナップショットを削除した後で削除されます。UFS スナップショットを作成する際に `-o unlink` オプションを指定しない場合は、後でバックングストアファイルを手動で削除する必要があります。

バックングストアファイルは、バックングストアファイルの削除に `-o unlink` オプションを使用した場合はスナップショットが削除されるまで、またはバックングストアファイルを手動で削除するまで、ディスク容量を使用します。

### ▼ UFS スナップショットを削除する方法

スナップショットは、システムをリブートするか、あるいは `fssnap -d` コマンドを使用して削除できます。このコマンドを使用するときには、UFS スナップショットを含むファイルシステムのパスを指定する必要があります。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 削除するスナップショットを特定します。

```
# /usr/lib/fs/ufs/fssnap -i
```

3. スナップショットを削除します。

```
# fssnap -d /file-system
Deleted snapshot 1.
```

4. (省略可能) スナップショットの作成時に `-o unlink` オプションを使用しなかった場合は、そのバックングストアファイルを手動で削除してください。

```
# rm /file-system/backing-store-file
```

#### 例 24-2 UFS スナップショットを削除する

次の例は、`-o unlink` オプションを使用しなかった場合に、スナップショットを削除する方法を示します。

```
# fssnap -i
0 / 1 /usr
# fssnap -d /usr
Deleted snapshot 1.
# rm /scratch/usr.back.file
```



---

## UFS スナップショットのバックアップ

UFS スナップショットの完全バックアップまたは増分バックアップを作成できます。UFS スナップショットのバックアップ作成に、標準の Solaris バックアップコマンドを使用できます。

UFS スナップショットを含む仮想デバイスは、標準の読み取り専用デバイスとして動作します。つまり、仮想デバイスを、ファイルシステムのデバイスをバックアップするかのようバックアップできます。

`ufsdump` コマンドを使用して UFS スナップショットをバックアップする場合、バックアップ時にスナップショットの名前を指定することができます。詳細については、次の手順を参照してください。

### ▼ UFS スナップショットの完全バックアップを作成する方法 (`ufsdump`)

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. バックアップする UFS スナップショットを確認します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

たとえば、次のようになります。

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number      : 0
Block Device         : /dev/fssnap/0
Raw Device           : /dev/rfssnap/0
Mount point          : /usr
Device state         : idle
Backing store path   : /var/tmp/snapshot3
Backing store size   : 256 KB
Maximum backing store size : Unlimited
Snapshot create time : Wed Oct 08 10:38:25 2003
Copy-on-write granularity : 32 KB
```

3. UFS スナップショットをバックアップします。

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

たとえば、次のようになります。

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4. スナップショットがバックアップされたことを確認します。  
たとえば、次のようになります。

```
# ufsrestore tf /dev/rmt/0
```

## ▼ UFS スナップショットの増分バックアップの作成方法 (ufsdump)

UFS スナップショットの増分バックアップでは、最後のスナップショット以降に変更のあったファイルだけがバックアップされます。ufsdump コマンドと新規 N オプションを組み合わせて使用します。このオプションは、増分ダンプをトラックするために /etc/dumpdates ファイルに挿入されるファイルシステムのデバイス名を指定します。

次の例では、ufsdump コマンド内で fssnap コマンドを組み込んでファイルシステムの増分バックアップを作成しています。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. UFS スナップショットの増分バックアップを作成します。  
たとえば、次のようになります。

```
# ufsdump lufN /dev/rmt/0 /dev/rdisk/c0t1d0s0 `fssnap -F ufs -o raw,bs=  
/export/scratch,unlink /dev/rdisk/c0t1d0s0`
```

上記の例では、ブロックデバイスではなく raw デバイスの名前を表示するために -o raw オプションが使用されています。このオプションの使用により、fssnap コマンドを raw デバイスを必要とするコマンド (ufsdump コマンドなど) に組み込むことが容易になります。

3. スナップショットがバックアップされたことを確認します。

```
# ufsrestore ta /dev/rmt/0
```

## ▼ UFS スナップショットのバックアップ方法 (tar)

tar コマンドを使用してスナップショットをバックアップする場合、バックアップを行う前にスナップショットをマウントします。

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. スナップショット用のマウントポイントを作成します。  
たとえば、次のようになります。

```
# mkdir /backups/home.bkup
```

3. スナップショットをマウントします。

```
# mount -F ufs -o ro /dev/fssnap/1 /backups/home.bkup
```

4. マウントスナップショットのディレクトリに移動します。

```
# cd /backups/home.bkup
```

5. `tar` コマンドを使用して、スナップショットをバックアップします。

```
# tar cvf /dev/rmt/0 .
```

## UFS スナップショットのバックアップからのデータの復元

仮想デバイスから作成されたバックアップは、基本的には、スナップショットがとられた時点でのオリジナルのファイルシステムの状態を表しています。ファイルシステムをバックアップから復元する場合は、オリジナルのファイルシステムから直接そのバックアップを作成したかのように復元します。そのような復元では、`ufsrestore` コマンドを使用します。`ufsrestore` コマンドを使用してファイルまたはファイルシステムを復元する方法については、[第 25 章](#)を参照してください。



## 第 25 章

# ファイルとファイルシステムの復元 (手順)

この章では、`ufsdump` コマンドを使用してバックアップを作成したファイルおよびファイルシステムを、`ufsrestore` コマンドを使用して復元する方法について説明します。

ファイルおよびファイルシステムの復元に関連した手順については、[421 ページ](#)の「[ファイルおよびファイルシステムのバックアップを復元する \(作業マップ\)](#)」を参照してください。

ファイルやファイルシステムのアーカイブ、復元、コピー、または移動に使用可能な他のコマンドについては、[第 27 章](#)を参照してください。

ファイルシステムのバックアップおよび復元については、[第 22 章](#)を参照してください。

---

## ファイルおよびファイルシステムの バックアップを復元する (作業マップ)

次の作業マップに、ファイルおよびファイルシステムの復元手順を示します。

| 作業                 | 説明                                          | 参照先                                                                  |
|--------------------|---------------------------------------------|----------------------------------------------------------------------|
| ファイルとファイルシステムの復元準備 | 復元するファイルシステムやファイル、テープデバイス、およびそれらの復元方法を決定する。 | <a href="#">422 ページ</a> の「 <a href="#">ファイルとファイルシステムを復元するための準備</a> 」 |

| 作業                           | 説明                                                                                  | 参照先                              |
|------------------------------|-------------------------------------------------------------------------------------|----------------------------------|
| 使用するテープの決定                   | バックアップテープを調べ、復元するファイルまたはファイルシステムを含む最新のバックアップの日付を確認する。                               | 424 ページの「使用するテープを決定する方法」         |
| ファイルの復元                      | 次のいずれかの復元方法を選択する。                                                                   |                                  |
|                              | ファイルの対話的な復元 – 正確なファイル名を把握していない場合、この方法を使用する。この方法では、メディアの内容を参照して、ファイルやディレクトリ単位で選択できる。 | 425 ページの「対話式でファイルを復元する方法」        |
|                              | ファイルの非対話的な復元 – 復元するファイル名を把握している場合には、この方法を使用する。                                      | 427 ページの「特定のファイルを復元する方法」         |
|                              | ファイルシステムの復元 – 復元時の手順の一部として新規ディスクドライブを使用できる場合は、この方法を使用する。                            | 429 ページの「ファイルシステム全体を復元する方法」      |
| ルート (/) または /usr ファイルシステムの復元 | ルート (/) または /usr ファイルシステムを復元する場合には、ローカル CD またはネットワークからシステムを起動する必要がある。               | 432 ページの「ルート (/) と /usr を復元する方法」 |

## ファイルとファイルシステムを復元するための準備

ufsrestore コマンドは、ufsdump コマンドを使用して作成されたバックアップから、現在の作業ディレクトリにファイルをコピーします。ufsrestore コマンドを使用すると、レベル 0 のダンプとそれ以降の増分ダンプからファイルシステム階層全体を読み込み直すことができます。また、このコマンドを使用して、任意のバックアップテープから個々のファイルを復元することもできます。ufsrestore コマンドをスーパーユーザーとして実行した場合には、ファイルの所有者、最新の変更時刻、モード (ファイルのアクセス権) は元のまま、ファイルが復元されます。

ファイルまたはファイルシステムの復元を開始する前に、次の点を確認してください。

- 復元に必要なテープ(またはフロッピーディスク)
- ファイルシステム全体を復元する raw デバイス名
- 使用するテープデバイスのタイプ
- テープデバイスのデバイス名 (ローカルまたはリモート)

## ファイルシステム名の確認

バックアップテープに適切な名前が付いている場合は、テープラベルに入っているファイルシステム名 (`/dev/rdisk/device-name`) が使えるはずですが、詳細については、[402 ページの「ファイルシステム名を検索する方法」](#)を参照してください。

## 必要なテープデバイスのタイプの決定

ファイルを復元するには、バックアップメディアと互換性のあるテープデバイスを使用する必要があります。バックアップメディアの形式によって、ファイルの復元にどんなドライブを使用しなければならないかが決まります。たとえば、使用するバックアップメディアが 8mm テープの場合、ファイルの復元には 8mm テープデバイスを使わなければなりません。

## テープデバイス名の決定

テープデバイス名 (`/dev/rmt/n`) をバックアップテープラベル情報の一部として指定しているかも知れません。同じドライブを使ってバックアップテープを復元しようとする場合には、ラベル内にあるデバイス名を使うことができます。メディアのデバイスとデバイス名の詳細については、[第 28 章](#)を参照してください。

---

## ファイルシステムの復元

バックアップを実行すると、ファイルやディレクトリは、それらが含まれるファイルシステムからの相対的な位置に保存されます。ファイルとディレクトリを復元するときは、`ufsrestore` コマンドが現在の作業ディレクトリにファイル階層を作成し直します。

たとえば、`/export/doc/books` ディレクトリ (`/export` はファイルシステム) からバックアップされたファイルは、`/export` からの相対的な位置に保存されます。つまり、`books` ディレクトリ内の `book1` ファイルは、テープ上で `./doc/books/book1` として保存されます。後で、`./doc/books/book1` ファイルを `/var/tmp` ディレクトリに復元する場合、そのファイルは `/var/tmp/doc/books/book1` に復元されます。

個別のファイルやディレクトリを復元するときには、`/var/tmp` などの一時的な場所に復元する必要があります。ファイルを確認したら、それを適当な位置に移動させてもかまいません。ただし、個別のファイルやディレクトリはそれぞれ元の位置に復元できます。その場合には、新しいファイルをバックアップテープからの古いバージョンで上書きしないかどうか確かめてください。

他のユーザーとの重複を防ぐために、`/var/tmp/restore` などのサブディレクトリを作成し、そのディレクトリに移動して、ファイルを復元することをお勧めします。

階層を復元する場合、ファイルを配置するファイルシステム内の一時ディレクトリにファイルを復元する必要があります。復元後に、`mv` コマンドを使用して階層全体を適切な場所に移動します。

---

注 – 一時的な場合でも、`/tmp` ディレクトリにファイルを復元してはなりません。`/tmp` ディレクトリは通常、TMPFS ファイルシステムとしてマウントされます。TMPFS は ACL などの UFS ファイルシステム属性をサポートしていません。

---

## ▼ 使用するテープを決定する方法

- 手順
1. 復元するファイルを最後に変更したのはいつごろだったかをユーザーにたずねます。
  2. バックアップ計画を参照し、該当するファイルまたはファイルシステムを含む最後のバックアップの日付を調べます。  
ファイルの最新バージョンを検索するには、特にユーザーの要求がない限り、増分バックアップファイルを最高のダンプレベルから最低のダンプレベルへ、最新の日付からもっとも古い日付へ逆方向に調べます。

3. (省略可能) オンラインアーカイブファイルを保持している場合、使用に適したメディアを判別します。

```
# ufsrestore ta archive-name ./path/filename ./path/filename
```

t                    テープ上の各ファイルを表示する。

a                    内容一覧をテープからではなく、オンラインアーカイブファイルから読み取る。

*archive-name*      オンラインアーカイブファイル名を指定する。

*./path/filename*    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、`ufsrestore` は *i* ノード番号とファイル名を出力する。成功しなかった場合は、エラーメッセージを出力する。

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

4. 復元するファイルを含むメディアをドライブに挿入し、適切なメディアであることを確認します。

```
# ufsrestore tf /dev/rmt/n ./path/filename ./path/filename
```

各ファイル名に必ず完全パス名を使用してください。ファイルがバックアップに入っていれば、その名前と *i* ノード番号が表示されます。ファイルがバックアップに入っていない場合、ファイルがそのボリュームに入っていないことを示す



メッセージが表示されます。

5. (省略可能) 同じテープに複数のバックアップファイルが入っている場合は、使用するバックアップファイルの位置までテープを移動します。

```
# ufsrestore tfs /dev/rmt/n tape-number
```

#### 例 25-1 使用するテープを決定する

次の例は、オンラインアーカイブに /etc/passwd ファイルが存在するかどうかを確認する方法を示します。

```
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

次の例は、バックアップテープに /etc/passwd ファイルが存在するかどうかを確認する方法を示します。

```
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

## ▼ 対話式でファイルを復元する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. (省略可能) 安全性のため、書き込み保護を設定します。
  3. ボリューム 1 のテープをテープドライブに挿入します。
  4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5. 対話式で復元を開始します。

```
# ufsrestore if /dev/rmt/n
```

情報を伝えるためのメッセージと ufsrestore> プロンプトが表示されます。

6. 復元するファイルのリストを作成します。

- a. ディレクトリの内容を表示します。

```
ufsrestore> ls [directory-name]
```

- b. ディレクトリを変更します。

```
ufsrestore> cd directory-name
```

- c. 復元するファイルとディレクトリのリストを作成します。

```
ufsrestore> add filenames
```

- d. (省略可能) 必要に応じ、復元するファイルのリストからディレクトリまたはファイルを削除します。

```
ufsrestore> delete filename
```

7. (省略可能) 復元処理中にファイル名を表示します。

```
ufsrestore> verbose
```

8. ファイルを復元します。

```
ufsrestore> extract
```

どのボリューム番号を使用するかを指定するプロンプトが表示されます。

9. ボリューム番号を入力して、**Return** キーを押します。ボリュームが1つしかない場合には、**1**を入力して**Return** キーを押します。

```
Specify next volume #: 1
```

リスト内のファイルとディレクトリが抽出され、現在の作業ディレクトリに復元されます。

10. 現在の作業ディレクトリのモードを変更しない場合は、**set owner/mode** プロンプトが表示されたときに **n** を入力します。

```
set owner/mode for `.'? [yn] n
```

ufsrestore コマンドによる最後の整理が完了すると、ufsrestore のプロンプトが表示されます。

11. **ufsrestore** プログラムを終了します。

```
ufsrestore> quit
```

シェルプロンプトが表示されます。

12. 復元されたファイルを確認します。

- a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

- b. リストをチェックして、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

13. ファイルを適切なディレクトリに移動します。

## 例 25-2 対話式でファイルを復元する

次の例では、`/etc/passwd` ファイルと `/etc/shadow` ファイルがバックアップテープから復元されます。

```

# cd /var/tmp
# ufsrestore if /dev/rmt/0
ufsrestore> ls
.:
.cpr_config  etc/          lost+found/  sbin/
TT_DB/       export/       mnt/         tmp/
b/           home/         net/         usr
bin          kernel/       opt/         var/
dev/         lib           platform/    vol/
devices/     license/      proc/
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
#

```

## ▼ 特定のファイルを復元する方法

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. (省略可能) 安全性のため、書き込み保護を設定します。
  3. ボリューム 1 のテープをテープドライブに挿入します。
  4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5. ファイルを復元します。

```
# ufsrestore xvf /dev/rmt/n filename
```

x *filename* 引数内に指定されたファイルまたはディレクトリをコピーするように ufsrestore コマンドに指定する。

v 復元処理中にファイル名を表示する。

f /dev/rmt/n テープデバイス名を識別する。

*filename* 1つまたは複数のファイル名またはディレクトリ名を指定する。複数の場合は、空白で区切って指定する。たとえば、次のようになります。 ./export/home/user1/mail  
./export/home/user2/mail.

6. ファイルが入っているボリューム番号を入力します。 **Return** キーを押します。

```
Specify next volume #: 1  
ファイルは現在の作業ディレクトリに復元されます。
```

7. 現在のディレクトリのモードを変更しない場合は、 **set owner/mode** プロンプトが表示されたときに **n** と入力して **Return** キーを押します。

```
set owner/mode for './?' [yn] n
```

8. 復元されたファイルを確認します。

- a. 復元されたファイルとディレクトリを表示します。

```
# ls -l  
ファイルとディレクトリのリストが表示されます。
```

- b. リストをチェックして、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

9. ファイルを適切なディレクトリに移動します。

### 例 25-3 特定のファイルを復元する

次の例では、 >passwd ファイルと 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: Mon Oct 06 12:36:10 2003  
Dumped from: the epoch  
Level 9 dump of / on starbug:/dev/dsk/c0t0d0s0  
Label: none  
Extract directories from tape  
Initialize symbol table.  
Make node ./etc  
Extract requested files  
You have not read any volumes yet.  
Unless you know which volume your file(s) are on you should start  
with the last volume and work towards the first.  
Specify next volume #:1  
extract file ./etc/passwd  
extract file ./etc/shadow  
Add links  
Set directory mode, owner, and times.
```

```
set owner/mode for `.'? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

#### 例 25-4 リモートのテープデバイスからファイルを復元する

ufsrestore コマンドを使用するときテープデバイス名の前に *remote-host*: を追加することにより、ファイルをリモートドライブから復元できます。

次の例では、システム *venus* 上のリモートテープドライブ */dev/rmt/0* を使用してファイルを復元します。

```
# ufsrestore xf venus:/dev/rmt/0 ./etc/hosts
```

## ▼ ファイルシステム全体を復元する方法

時には、全面的に復元しなければならないほどファイルシステムが破壊される場合があります。一般的な例として、ディスク障害が発生した場合には、ファイルシステムを全面的に復元する必要があります。この場合、ハードウェアを交換してからソフトウェアを復元しなければならないこともあります。ディスクの交換方法については、[第 12 章](#)または[第 13 章](#)を参照してください。

*/export/home* などのファイルシステムを全面的に復元するには、時間がかかります。ファイルシステムを一貫性のある方法でバックアップしていれば、最後の増分バックアップ時の状態に復元することができます。

---

注 - ルート (/) または */usr* ファイルシステムの復元には、この手順は使用できません。これらのファイルシステムを復元する方法については、[432 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。

---

手順 1. スーパーユーザーになるか、同等の役割を引き受けます。

2. 必要に応じ、ファイルシステムをマウント解除します。

```
# umount /dev/rdisk/device-name
```

または

```
# umount /file-system
```

3. ファイルシステムを新規作成します。

```
# newfs /dev/rdisk/device-name
```

raw デバイス上に新しいファイルシステムを構築するかどうかを確認するプロンプトが表示されます。意図しないファイルシステムを間違っって損失してしまわないように、*device-name* が正しいことを確認します。

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

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. 新しいファイルシステムを一時的なマウントポイントにマウントします。

```
# mount /dev/dsk/device-name /mnt
```

6. マウントポイントのディレクトリに移動します。

```
# cd /mnt
```

7. (省略可能) 安全性のため、書き込み保護を設定します。

8. レベル 0 テープの第 1 ボリュームをテープドライブに挿入します。

9. ファイルを復元します。

```
# ufsrestore rvf /dev/rmt/n
```

ダンプレベル 0 のバックアップが復元されます。バックアップの実行に複数のテープが必要な場合は、番号の順にテープをロードするようにプロンプトが表示されます。

10. テープを外し、次のレベルのテープをドライブにロードします。

テープの復元は必ずダンプレベル 0 から始め、一番高いダンプレベルまで続けてください。

11. ダンプレベルごとに、一番低いレベルから一番高いレベルまで、手順 8 から手順 10 までの操作を繰り返します。

12. ファイルシステムが復元されていることを確認します。

```
# ls
```

13. **restoresymtable** ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために *ufsrestore* コマンドが作成し、使用した *restoresymtable* ファイルを削除します。

14. 別のディレクトリに移動します。

```
# cd /
```

15. 新しく復元されたファイルシステムをマウント解除します。

```
# umount /mnt
```

16. 最後のテープを取り出し、書き込み保護されていない新しいテープをテープドライブに挿入します。

17. 新しく復元されたファイルシステムのレベル 0 のバックアップを作成します。

```
# ufsdump 0ucf /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 コマンド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完全バックアップは、必ずすぐに作成するようにしてください。

18. 復元されたファイルシステムをマウントします。

```
# mount /dev/dsk/device-name mount-point
```

復元されたファイルシステムがマウントされ、使用できるようになります。

19. 復元およびマウントされたファイルシステムが使用できることを確認します。

```
# ls /mount-point
```

#### 例 25-5 ファイルシステム全体を復元する

次の例は、/export/home ファイルシステムの復元方法を示します。

```
# umount /export/home
# newfs /dev/rdisk/c0t0d0s7
newfs: /dev/rdisk/c0t0d0s7 last mounted as /export/home
newfs: construct a new file system /dev/rdisk/c0t0d0s7: (y/n)? y
819314 sectors in 867 cylinders of 15 tracks, 63 sectors
    400.1MB in 55 cyl groups (16 c/g, 7.38MB/g, 3584 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
    32, 15216, 30400, 45584, 60768, 75952, 91136, 106320, 121504, 136688,
    681264, 696448, 711632, 725792, 740976, 756160, 771344, 786528, 801712,
    816896,
Verify volume and initialize maps
Media block size is 126
Dump   date: Tue Oct 07 08:41:41 2003
Dumped from: the epoch
Level 0 dump of a partial file system on starbug:/export/home/kryten
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Extract new leaves.
Check pointing the restore
extract file ./export/home/kryten/filea
extract file ./export/home/kryten/fileb
extract file ./export/home/kryten/filec
extract file ./export/home/kryten/letters/letter1
extract file ./export/home/kryten/letters/letter2
```

```

extract file ./export/home/kryten/letters/letter3
extract file ./export/home/kryten/reports/reportA
extract file ./export/home/kryten/reports/reportB
extract file ./export/home/kryten/reports/reportC
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# mount /dev/dsk/c0t0d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
# 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

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 新しいシステムディスクを、ルート (/) と /usr ファイルシステムが復元されるシステムに追加します。  
システムディスクの追加方法については、[第 12 章](#)または[第 13 章](#)を参照してください。
  3. 新しいファイルシステムを一時的なマウントポイントにマウントします。  
# mount /dev/dsk/device-name /mnt
  4. /mnt ディレクトリに移動します。  
# cd /mnt
  5. (省略可能) 安全性のため、書き込み保護を設定します。
  6. テープデバイスのリンクを作成します。  
# tapes
  7. ルートファイルシステムを復元します。  
# ufsrestore rvf /dev/rmt/n  
ダンプレベル 0 のテープが復元されます。



8. テープを外し、次のレベルのテープをドライブにロードします。  
テープの復元は必ずダンプレベル 0 から始め、もっとも低いダンプレベルから  
もっとも高いダンプレベルまで続けてください。

9. 必要に応じ、復元を続行します。

```
# ufsrestore rvf /dev/rmt/n
```

次のレベルのテープが復元されます。

10. テープを追加するたびに、手順 8 と手順 9 を繰り返します。

11. 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
```

12. `restoresymtable` ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために `ufsrestore` コマンドが作成し、使用した  
`restoresymtable` ファイルを削除します。

13. ルート (/) ディレクトリに変更します。

```
# cd /
```

14. 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
```

15. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdsk/device-name
```

復元されたファイルシステムの整合性がチェックされます。

16. ルートパーティションにブートブロックを作成します。

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk  
/dev/rdsk/device-name
```

詳細は、`installboot` (1M) のマニュアルページを参照してください。

SPARC システム上で `installboot` コマンドを使用する方法については、例 25-6  
を参照してください。x86 システム上で `installboot` コマンドを使用する方法に  
ついては、例 25-7 を参照してください。

17. 新しいテープをテープドライブに挿入します。

18. 新しいファイルシステムのバックアップを作成します。

```
# ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
```

ダンプレベル 0 のバックアップが実行されます。 `ufsrestore` コマンドはファイル  
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  
システムの完全バックアップは、必ずすぐに作成するようにしてください。

19. 必要に応じ、`/usr` ファイルシステムについて手順 5 から 16 を繰り返します。

20. 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。
```

#### 例 25-6 SPARC: ルート (/) ファイルシステムを復元する

次の例は、SPARC システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、システムはローカルの CD またはネットワークから起動するものとします。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

#### 例 25-7 x86: ルート (/) ファイルシステムを復元する

次の例は、x86 システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、システムはローカルの CD またはネットワークから起動するものとします。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i`/lib/fs/
ufs/bootblk /dev/rdisk/c0t3d0s2
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

## 第 26 章

# UFS バックアップおよび復元コマンド (参照情報)

---

この章では、`ufsdump` コマンドと `ufsrestore` コマンドの参照情報を示します。

この章の内容は次のとおりです。

- 435 ページの「`ufsdump` コマンドの機能」
- 440 ページの「`ufsdump` コマンドのオプションと引数」
- 443 ページの「`ufsdump` とセキュリティに関する注意事項」
- 443 ページの「`ufsrestore` コマンドのオプションと引数」

バックアップの実行方法の概要については、第 22 章を参照してください。

バックアップ作業の詳細については、第 23 章を参照してください。

---

## ufsdump コマンドの機能

`ufsdump` コマンドは、ファイルシステムのバックアップ作成時に 2 つのパスを作成します。最初のパスでは、このコマンドは raw デバイスファイル内でファイルシステムを走査し、メモリー内にディレクトリとファイルのテーブルを作成します。次に、そのテーブルをバックアップメディアに書き込みます。2 つ目のパスでは、`ufsdump` は `i` ノードに番号順にアクセスし、ファイルの内容を読み込んでバックアップメディアに書き込みます。

## デバイス特性の判断

`ufsdump` コマンドに必要なことは、適切なテープブロックの大きさを認識することと、どのようにしてメディアの終わりを検出するかということです。

## メディアの終わりの検出

ufsdump コマンドは、一連の固定長レコードを書き込みます。ufsdump コマンドは、レコードの一部にしか書き込まれていないという通知を受け取ると、メディアの物理的な終わりに達したものと判断します。この方法は、ほとんどのデバイスに有効です。部分的なレコードしか書き込まれなかったことをデバイスが ufsdump に通知できない場合、ufsdump が別のレコードの書き込みを試みると、メディアエラーが発生します。

---

注 - DAT デバイスと 8mm テープデバイスでは、メディアの終わりが検出されます。カートリッジテープデバイスと 1/2 インチテープデバイスでは、メディアの終わりは検出されません。

---

ufsdump は、ほとんどのデバイスのメディアの終わりを自動的に検出します。したがって、通常は -c、-d、-s、-t オプションを使用しなくても、複数のボリュームのバックアップを実行できます。

メディアの終わりを検出するオプションを使用する必要があるのは、次の場合に限られます。

- デバイスがメディアの終わりを検出する方法を ufsdump コマンドで判断できない場合
- restore コマンドを使って SunOS 4.1 システム上でファイルを復元しようとしている場合

restore コマンドとの互換性を確保するため、サイズオプションを使用すると、従来どおり、現在のテープやフロッピーディスクの終わりに達する前に、ufsdump を次のテープやフロッピーディスクに強制的に進ませることができます。

## ufsdump コマンドを使用したデータのコピー

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 コマンド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 ファイルの目的

ufsdump コマンドを -u オプション付きで使用すると、/etc/dumpdates というファイルを管理し、更新できます。/etc/dumpdates ファイル内の各行は、次の情報を表しています。

- バックアップが作成されたファイルシステム
- 前回実行されたバックアップのダンプレベル
- バックアップの曜日、日付、および時刻

たとえば、次のようになります。

```
/dev/rdsk/c0t0d0s7          0 Wed Oct  8 10:30:52 2003
/dev/rdsk/c0t0d0s0          0 Tue Oct  6 10:12:13 2003
/dev/rdsk/c0t0d0s0          9 Wed Oct  8 10:26:14 2003
```

増分バックアップの実行時に、`ufsdump` コマンドは `/etc/dumpdates` ファイルをチェックして、下のダンプレベルの最後のバックアップ日付を調べます。次に、下のレベルのバックアップ以降に更新されたすべてのファイルをメディアにコピーします。バックアップが完了すると、完了したばかりのバックアップを記述する新しい情報行によって、そのレベルの最後のバックアップの情報行が置き換えられます。

`/etc/dumpdates` ファイルを使用して、バックアップが実行中であるかどうかを検査してください。機器に問題が発生している場合は、この検査が特に重要です。機器の障害が原因でバックアップを完了できないと、そのバックアップは `/etc/dumpdates` ファイルに記録されません。

ディスク全体を復元する必要がある場合は、`/etc/dumpdates` ファイル内で最後のバックアップの日付とレベルをチェックできるので、ファイルシステム全体を復元するために必要なファイルを判断できます。

---

注 - `/etc/dumpdates` ファイルは、編集可能なテキストファイルです。ただし、編集するかどうかはユーザーの判断によります。ファイルに変更を加えた結果、アーカイブテープと一致なくなると、必要なテープ (またはファイル) がどれであるかわからなくなることがあります。

---

## バックアップデバイス (*dump-file*) 引数

*dump-file* 引数 (`-f` オプションで使用) では、バックアップ先を指定します。バックアップ先は、次のいずれかになります。

- ローカルのテープドライブ
- ローカルの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
- リモートの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* 引数の値を指定しなければなりません。

---

注 - *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ます。誤用するとファイルシステムがいっぱいになる可能性があります。

---

## ローカルのテープドライブまたはフロッピーディスクドライブ

通常、*dump-file* 引数には、テープデバイスかフロッピーディスク用の *raw* デバイスファイルを指定します。*ufsdump* コマンドは、出力デバイスへの書き込み時にバックアップファイルを1つ作成しますが、このファイルは複数のテープやフロッピーディスクにまたがってもかまいません。

デバイスの省略形を使用して、システム上のテープデバイスかフロッピーディスクデバイスを指定します。最初のデバイスは常に 0 です。たとえば、SCSI テープコントローラが1つと、中密度の形式を使用する QIC-24 テープドライブが1つある場合は、次のデバイス名を使用します。

```
/dev/rmt/0m
```

テープデバイス名を指定するときは、名前の末尾に文字 *n* を付けて、バックアップの完了後にテープドライブを巻き戻さないように指定することもできます。たとえば、次のようになります。

```
/dev/rmt/0mn
```

テープに複数のファイルを格納する場合は、*no-rewind* オプションを使用します。バックアップ中に領域を使い果たすと、*ufsdump* コマンドから新しいテープの挿入を促すプロンプトが表示されるまで、テープは巻き戻されません。デバイスの命名規則の詳細については、474 ページの「バックアップデバイス名」を参照してください。

## リモートのテープドライブまたはフロッピーディスクドライブ

*host:device* という形式で、リモートのテープデバイスまたはフロッピーディスクを指定します。ローカルシステム上のスーパーユーザーがリモートシステムへのアクセス権を持っている場合、*ufsdump* コマンドはリモートデバイスに書き込みます。通常、スーパーユーザーとして *ufsdump* コマンドを実行するのであれば、ローカルシステム名をリモートシステムの *.rhosts* ファイルに記述しておく必要があります。デバイスを *user@host:device* と指定した場合、*ufsdump* コマンドは指定されたユーザーでリモートシステム上のデバイスへのアクセスを試みます。この場合、指定されたユーザーの名前が、リモートシステム上の *.rhosts* ファイル中に含まれている必要があります。

デバイスには、`ufsdump` コマンドを実行するシステムではなく、そのデバイスが存在するシステムのオペレーティングシステムに合った命名規則を使用してください。デバイスが SunOS の旧バージョン (4.1.1 など) を実行するシステム上にある場合は、SunOS 4.1 でのデバイス名 (`/dev/rst0` など) を使用します。システムが Solaris ソフトウェアを実行中の場合は、SunOS 5.9 でのデバイス名 (`/dev/rmt/0` など) を使用します。

## ufsdump コマンドで標準出力を使用する

`dump-file` 引数としてダッシュ (-) を指定すると、`ufsdump` コマンドは標準出力に書き込みます。

---

注 - `dump-file` 引数として標準出力を指定すると、`-v` オプション (検査) は機能しません。

---

`ufsdump` コマンドを使用して標準出力に書き込み、`ufsrestore` コマンドを使用して標準入力から読み込むと、パイプライン内でファイルシステムをコピーできます。たとえば、次のようになります。

```
# ufsdump 0f - /dev/rdsk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

## バックアップを作成するファイルを指定する

コマンド行の最後の引数として、バックアップするファイル (`filenames`) を必ず指定してください。この引数は、バックアップのコピー元または内容を指定します。

ファイルシステムの場合、次のように raw デバイスファイルを指定します。

```
/dev/rdsk/c0t0d0s6
```

ファイルシステムは、そのエントリが `/etc/vfstab` ファイルに存在すれば、マウントポイントディレクトリ (`/export/home` など) を使用して指定できます。

デバイスの命名規則の詳細については、[474 ページの「バックアップデバイス名」](#)を参照してください。

個々のファイルやディレクトリごとに、1 つまたは複数の名前を空白で区切って入力します。

---

注 - `ufsdump` コマンドを使用して (ファイルシステム全体ではなく) 1 つまたは複数のディレクトリやファイルのバックアップを作成するときには、レベル 0 のバックアップが実行されます。増分バックアップは適用されません。

---

## テープの性質を指定する

テープの性質を指定しなければ、`ufsdump` コマンドはデフォルト設定を使用します。テープカートリッジ (-c)、密度 (-d)、サイズ (-s)、トラック数 (-t) を指定できます。オプションの順序とその引数の順番が一致していれば、オプションはいくつでも指定できます。

## `ufsdump` の制限

次に、`ufsdump` コマンドでは実行できない操作を示します。

- ファイルシステムのバックアップに必要なテープやフロッピーディスクの数を自動的に計算する。仮実行 (ドライラン) モード (s オプション) を使用すると、実際にファイルシステムをバックアップする前に必要な容量を判断できる。
- アクティブなファイルシステムをバックアップするときの問題を最小限度に抑えるために、組み込みエラーチェック機能を提供する。
- サーバーからリモートにマウントされたファイルをバックアップする。サーバー上のファイルのバックアップは、そのサーバー上で実行しなければならない。ユーザーがサーバー上で所有するファイル上で `ufsdump` コマンドを実行するアクセス権は拒否される。

---

## `ufsdump` コマンドのオプションと引数

この節では、`ufsdump` コマンドのオプションと引数について詳しく説明します。`ufsdump` コマンドの構文を、次に示します。

```
/usr/sbin/ufsdump options arguments filenames
```

- |                  |                                                            |
|------------------|------------------------------------------------------------|
| <i>options</i>   | 1 文字のオプション名からなる 1 つの文字列。                                   |
| <i>arguments</i> | オプションの引数を指定する。複数の文字列も指定可。オプション文字とそれに関連する引数は、同じ順序で並べる必要がある。 |
| <i>filenames</i> | バックアップするファイルを指定する。これらの引数は、空白で区切り、常に最後に指定する。                |

## `ufsdump` のデフォルトオプション

オプションを指定せずに `ufsdump` コマンドを実行する場合は、次の構文を使用します。

```
# ufsdump filenames
```



ufsdump コマンドでは、デフォルトで次のオプションと引数が使用されます。

```
ufsdump 9uf /dev/rmt/0 filenames
```

これらのオプションでは、デフォルトのテープドライブ上にその推奨密度でレベル 9 の増分バックアップが作成されます。

## ufsdump コマンドのオプション

次の表に、ufsdump コマンドのオプションを示します。

表 26-1 ufsdump コマンドのオプション

| オプション                 | 説明                                                                                                                                                                                                                                                  |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-9                   | ダンプレベル。レベル 0 は、 <i>filenames</i> で指定したファイルシステム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。バックアップ用の内容一覧を格納 (アーカイブ) するファイルを指定する。このファイルは、 <i>ufsrestore</i> コマンドでしか認識できない。このコマンドは、指定された内容一覧を使用して、復元すべきファイルがバックアップファイル内にあるかどうかと、もしあればどのメディアボリュームに入っているかを判断する。                                                                           |
| b <i>factor</i>       | ブロック係数。1 処理ごとにテープに書き込まれる 512 バイトのブロック数を指定する。                                                                                                                                                                                                        |
| c                     | カートリッジ。バックアップメディアとしてカートリッジテープを指定する。メディアの終わりの検出を適用するときは、このオプションでブロックサイズを 126 に設定する。                                                                                                                                                                  |
| d <i>bpi</i>          | テープ密度。テープの密度を指定する。このオプションは、 <i>ufsdump</i> コマンドでメディアの終わりを検出できない場合にのみ使用する。                                                                                                                                                                           |
| D                     | フロッピーディスク。バックアップメディアとしてフロッピーディスクを指定する。                                                                                                                                                                                                              |
| f <i>dump-file</i>    | ダンプファイル。デフォルトデバイスではなく <i>dump-file</i> で指定したコピー先にファイルを書き込む。ファイルを <i>user@system:device</i> として指定すると、 <i>ufsdump</i> コマンドは指定されたユーザーとしてリモートシステム上で実行しようとする。ローカルシステム上でこのコマンドを実行してリモートシステムにアクセスするため、指定されたユーザーはリモートシステム上に <i>/.rhosts</i> ファイルを保持する必要がある。 |

表 26-1 ufsdump コマンドのオプション (続き)

| オプション    | 説明                                                                                                                                                                                                                                                        |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| l        | 自動ロード。このオプションは、オートロード (スタックローダ) テープドライブがある場合に使用する。テープの終わりに達すると、このオプションはドライブをオフラインにして、テープドライブの準備ができるまで 2 分間待つ。2 分以内にドライブの準備ができると、自動ロードを続行する。2 分経過してもドライブの準備ができていない場合、オペレータに別のテープをロードするように促すプロンプトが表示される。                                                    |
| n        | 通知。介入が必要になると、sys グループのユーザー全員の端末にメッセージを送る。                                                                                                                                                                                                                 |
| o        | オフライン。テープやフロッピーディスクの処理が終わると、ドライブをオフラインにして巻き戻し (テープの場合)、可能であればメディアをはずす。たとえば、フロッピーディスクを取り出したり、8mm の自動ロードテープをはずしたりする。                                                                                                                                        |
| s size   | サイズ。バックアップメディアのサイズを指定する。テープの場合はフィート数を指定する。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、ufsdump コマンドでメディアの終わりを検出できない場合にのみ使用する。                                                                                                                               |
| S        | サイズ。バックアップのサイズを予想する。バックアップを実際に行わずに必要な容量を判断する。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 tracks | トラック数。1/4 インチカートリッジテープのトラック数を指定する。このオプションは、ufsdump コマンドでメディアの終わりを検出できない場合にのみ使用する。                                                                                                                                                                         |
| u        | アップデート。ダンプレコードをアップデートする。ファイルシステムの完全バックアップを実行する場合には、 <code>/etc/dumpdates</code> ファイルにエントリを追加する。エントリは、ファイルシステムのディスクスライスのデバイス名、ダンプレベル (0 ~ 9)、および日付を示す。u オプションを使用しないときや、個々のファイルかディレクトリのバックアップを作成するときは、レコードは書き込まれない。バックアップのレコードがすでに同じレベルに存在する場合は、それが置き換えられる。 |
| v        | 検査。各テープまたはフロッピーディスクへの書き込み後に、ソースファイルシステムと対照してメディアの内容を検査する。不整合が検出されると、オペレータに新しいメディアのマウントを促すプロンプトを表示してプロセスを繰り返す。ファイルシステム内で操作が実行されると、ufsdump コマンドが不整合を報告するため、このオプションはマウント解除されたファイルシステムにのみ使用する必要がある。                                                           |

表 26-1 ufsdump コマンドのオプション (続き)

| オプション | 説明                                                                                                            |
|-------|---------------------------------------------------------------------------------------------------------------|
| w     | 警告。/etc/dumpdates 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。/etc/dumpdates 内のすべてのファイルシステムを表示し、特定の日にバックアップされていないファイルシステムを強調表示する。このオプションを使用すると、他のすべてのオプションは無視される。 |

注 - /etc/vfstab ファイルには、ファイルシステムのバックアップ頻度に関する情報は含まれません。

## ufsdump とセキュリティに関する注意事項

セキュリティ保護を適用するには、次の操作を実行する必要があります。

- ufsdump コマンドの実行には、スーパーユーザーのアクセス権を必要とする。
- 集中バックアップを実行する場合は、クライアント上とサーバー上の /.rhosts ファイルからスーパーユーザーアクセスのエントリを削除する。

セキュリティに関する一般的な情報は、『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

## ufsrestore コマンドのオプションと引数

ufsrestore コマンドの構文を、次に示します。

```
/usr/sbin/ufsrestore options arguments filenames
```

- options*      1 文字のオプション名からなる 1 つの文字列。i、r、R、t、x から 1 つだけ選択する。表 26-3 に示す追加オプションは、省略可能。
- arguments*    オプションに対応する引数。オプション文字とそれに関連する引数は、同じ順序で並べる必要がある。

*filenames* 復元するファイルを、*x* または *t* オプションの引数として指定する。これらの引数は、空白で区切り、常に最後に指定する。

次の表に示す `ufsrestore` コマンドオプションのうち、1つだけを指定する必要があります。

表 26-2 `ufsrestore` コマンドに必要なオプション

| オプション                         | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>i</i>                      | 対話式。 <code>ufsrestore</code> コマンドを対話モードで実行する。このモードでは、限られたシェルコマンドセットを実行してメディアの内容を表示し、復元するファイルやディレクトリを個別に選択できる。対話式コマンドのリストについては、表 26-4 を参照。                                                                                                                                                                                                                                                                                                              |
| <i>r</i>                      | 再帰的。メディアの内容全体を現在の作業ディレクトリ (ファイルシステムの最上位レベル) に復元する。完全バックアップ ( <code>restoresymtable</code> など) の最上部に増分バックアッ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) バックアップを復元してから、各増分バックアップを復元する。このオプションは新しい ( <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、バックアップメディアにないファイルが保存される。                                                                                                                                                                 |
| <i>R</i>                      | 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 ( <i>r</i> 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。                                                                                                                                                                                                                                                                                                                       |
| <i>x</i> [ <i>filenames</i> ] | 抽出。 <i>filenames</i> 引数で指定したファイルを選択的に復元する。 <i>filenames</i> には、ファイルとディレクトリのリストを空白で区切って指定できる。 <i>h</i> 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>filenames</i> を省略するか、ルートディレクトリを表す「。」を入力すると、メディアのすべてのボリューム上 (または標準入力から) のすべてのファイルが復元される。既存のファイルは上書きされ、警告が表示される。                                                                                                                                                                                       |
| <i>t</i> [ <i>filenames</i> ] | 内容一覧。 <i>filenames</i> 引数で指定したファイルがメディアと対照してチェックされる。ファイルごとに、完全ファイル名と <i>i</i> ノード番号 (ファイルが見つかった場合) が表示されるか、ファイルが「ボリューム」上にないことを示す (複数ボリュームのバックアップ内のボリュームを意味する)。 <i>filenames</i> 引数を入力しなければ、メディアのすべてのボリューム上のファイルが表示される (どのボリュームにファイルが入っているかは区別されない)。<br><br><i>h</i> オプションも指定すると、内容ではなく <i>filenames</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、メディアの最初のボリューム、またはアーカイブファイル ( <i>a</i> オプションを使用した場合) から読み込まれる。このオプションは <i>x</i> 、 <i>r</i> オプションと一緒に使用できない。 |

次の表に、`ufsrestore` の追加オプションを示します。これらのオプションは省略可能です。

表 26-3 `ufsrestore` コマンドの追加オプション

| オプション                                   | 説明                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>a archive-file [filenames]</code> | アーカイブファイル。バックアップの内容一覧は、メディア (最初のボリューム) ではなく、指定した <i>archive-file</i> から取り出される。このオプションを <code>t</code> 、 <code>i</code> 、または <code>x</code> オプションとともに使用すると、メディアをマウントしなくても、ファイルがメディアに存在するかどうかを確認できる。このオプションを <code>x</code> オプションおよび対話型 ( <code>i</code> ) 抽出オプションと組み合わせて使用すると、ファイルを抽出する前に適切なボリュームのマウントを促すプロンプトが表示される。                                                                        |
| <code>b factor</code>                   | ブロック係数。1 回の処理でテープから読み込む 512 バイトのブロック数を指定する。デフォルトでは、 <code>ufsrestore</code> コマンドはテープへの書き込みに使用したブロックサイズの使用を試みる。                                                                                                                                                                                                                                                                              |
| <code>d</code>                          | デバッグ。デバッグメッセージ機能をオンにする。                                                                                                                                                                                                                                                                                                                                                                     |
| <code>f backup-file</code>              | 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 <code>/dev/rmt/0m</code> ではなく <i>backup-file</i> で指定したソースから読み込まれる。 <code>f</code> オプションを使用する場合は、 <i>backup-file</i> の値を指定する必要がある。 <i>backup-file</i> が <code>system:device</code> 形式であれば、 <code>ufsrestore</code> 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 <code>"-"</code> であれば、ファイルは標準入力から読み込まれる。 |
| <code>h</code>                          | 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。                                                                                                                                                                                                                                                                                                                                                |
| <code>m</code>                          | 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る。さらに、指定したファイルの名前が <code>i</code> ノード番号を使用して変更される。たとえば、現在の作業ディレクトリが <code>/files</code> であれば、 <code>i</code> ノード番号が 42 のバックアップ <code>./dready/fcs/test</code> 内のファイルは、 <code>/files/42</code> として復元される。このオプションは、少数のファイルを抽出する場合にのみ有用である。                                                                                                      |
| <code>s n</code>                        | スキップ。最初のボリュームメディア上の <code>n</code> 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。                                                                                                                                                                                                                                                                                            |
| <code>v</code>                          | 詳細表示。各ファイルが復元されるたびに、その名前と <code>i</code> ノード番号が表示される。                                                                                                                                                                                                                                                                                                                                       |

表 26-3 ufsrestore コマンドの追加オプション (続き)

| オプション | 説明                                                                                                                           |
|-------|------------------------------------------------------------------------------------------------------------------------------|
| y     | メディアの読み込みエラーが発生しても処理を続行する。このオプションを指定すると、処理を停止して続行するかどうかを選択するプロンプトを表示せずに、不良ブロックをスキップしようとする。このオプションによって、コマンドは肯定の応答とみなすよう命令される。 |

次の表に、ufsrestore の対話式コマンドを示します。

表 26-4 対話式の復元コマンド

| オプション                        | 説明                                                                                                                                                                                                                                      |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ls [ <i>directory-name</i> ] | 現在のディレクトリまたは指定したディレクトリの内容を表示する。ディレクトリは接尾辞 / 付きで表示される。現在のリスト内で復元 (抽出) されるエントリは接頭辞 * 付きで表示される。詳細オプション (v) を使用すると、i ノード番号が表示される。                                                                                                           |
| cd <i>directory-name</i>     | バックアップ階層内の指定したディレクトリに変更する。                                                                                                                                                                                                              |
| add [ <i>filename</i> ]      | 現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストに追加する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のすべてのファイルがリストに追加される。ディレクトリに復元したいすべてのファイルが 1 つのバックアップテープやフロッピーディスクに入っていないことがある。すべてのファイルの最新バージョンを抽出するには、さまざまなレベルの複数のバックアップから復元しなければならない場合がある。 |
| delete [ <i>filename</i> ]   | 現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストから削除する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のファイルがすべて削除される。ファイルとディレクトリは、構築中の抽出リストからのみ削除される。メディアまたはファイルシステムからは削除されない。                                                                   |
| extract                      | リスト内のファイルを抽出し、ディスク上の現在の作業ディレクトリからの相対パスで指定される位置に復元する。単一ボリュームのバックアップに関して、ボリューム番号を確認するプロンプトが表示されたら、1 を指定する。複数テープや複数フロッピーディスクから少数のファイルを復元する場合は、最後のテープまたはフロッピーディスクから始める。                                                                     |
| help                         | 対話式で使用できるコマンドのリストが表示される。                                                                                                                                                                                                                |
| pwd                          | バックアップ階層内の現在の作業ディレクトリのパス名が表示される。                                                                                                                                                                                                        |
| q                            | それ以上ファイルを復元しないで対話モードを終了する。                                                                                                                                                                                                              |

表 26-4 対話式の復元コマンド (続き)

| オプション    | 説明                                                                                                                                                                                                                                                                                                                                              |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| setmodes | <p>バックアップ元となったファイルシステムのルートディレクトリのモードに合わせて、復元するファイルのモードを設定できる。set owner/mode for '.' [yn]? というプロンプトが表示される。y (yes の意味) を入力すると、バックアップ元となったファイルシステムのルートディレクトリに合わせて、現在のディレクトリのモード (アクセス権、所有者、時刻) を設定できる。このモードは、ファイルシステム全体を復元する場合に使用する。</p> <p>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。</p> |
| verbose  | <p>詳細オプションのオンとオフを切り替える (対話型モードの外側では、コマンド行から v と入力することもできる)。詳細モードがオンの場合、対話型の 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。</p>                                                                                                                                                                                             |
| what     | <p>テープやフロッピーディスク上のバックアップヘッダが表示される。</p>                                                                                                                                                                                                                                                                                                          |





## 第 27 章

# UFS ファイルとファイルシステムのコピー (手順)

---

この章では、各種のバックアップコマンドを使用して、UFS ファイルとファイルシステムをディスク、テープ、フロッピーディスクにコピーする方法について説明します。

この章で説明する手順は次のとおりです。

- 452 ページの「ディスクをコピーする方法 (dd)」
- 455 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)」
- 458 ページの「ファイルをテープにコピーする方法 (tar)」
- 459 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)」
- 459 ページの「テープからファイルを取り出す方法 (tar)」
- 460 ページの「pax コマンドを使用してファイルをテープにコピーする」
- 461 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)」
- 462 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)」
- 463 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)」
- 464 ページの「テープから特定のファイルを取り出す方法 (cpio)」
- 465 ページの「ファイルをリモートテープデバイスにコピーする方法 (tar と dd)」
- 467 ページの「ファイルをリモートテープデバイスから取り出す方法」
- 468 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)」
- 469 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)」
- 470 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)」

---

## ファイルシステムをコピーするためのコマンド

ファイルシステム全体をバックアップして復元する場合は、第 26 章で説明した `ufsdump` コマンドと `ufsrestore` コマンドを使用します。個々のファイル、ファイルシステムの一部、またはファイルシステム全体をコピーまたは移動する場合は、`ufsdump` と `ufsrestore` コマンドの代わりに、この章で説明する手順を使用できます。

次の表に、各種バックアップコマンドの用途を示します。

表 27-1 バックアップコマンドの用途

| 作業                          | コマンド                                                       | 参照先                                                 |
|-----------------------------|------------------------------------------------------------|-----------------------------------------------------|
| ファイルシステムをテープにバックアップする       | <code>ufsdump</code>                                       | 404 ページの「ファイルシステムのバックアップをテープに作成する方法」                |
| ファイルシステムのスナップショットを作成する      | <code>fssnap</code>                                        | 第 24 章                                              |
| ファイルシステムをテープから復元する          | <code>ufsrestore</code> コマンド                               | 429 ページの「ファイルシステム全体を復元する方法」                         |
| ファイルを他のシステムに転送する            | <code>pax</code> 、 <code>tar</code> 、または <code>cpio</code> | 456 ページの「ファイルとファイルシステムをテープにコピーする」                   |
| ファイルまたはファイルシステムをディスク間でコピーする | <code>dd</code>                                            | 452 ページの「ディスクをコピーする方法 (dd)」                         |
| ファイルをフロッピーディスクにコピーする        | <code>tar</code>                                           | 468 ページの「ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)」 |

次の表に、各種のバックアップおよび復元コマンドを示します。

表 27-2 バックアップコマンドの概要

| コマンド名                | ファイルシステム境界の認識 | 複数ボリュームバックアップのサポート | 物理コピー / 論理コピー |
|----------------------|---------------|--------------------|---------------|
| <code>volcopy</code> | あり            | あり                 | 物理            |
| <code>tar</code>     | なし            | なし                 | 論理            |

表 27-2 バックアップコマンドの概要 (続き)

| コマンド名              | ファイルシステム境界の認識 | 複数ボリュームバックアップのサポート | 物理コピー / 論理コピー |
|--------------------|---------------|--------------------|---------------|
| cpio               | なし            | あり                 | 論理            |
| pax                | あり            | あり                 | 論理            |
| dd                 | あり            | なし                 | 物理            |
| ufsdump/ufsrestore | あり            | あり                 | 論理            |

次の節では、各コマンドの長所と短所を説明します。また、コマンドを使用する手順と使用例も挙げていきます。

## ファイルシステムをディスク間でコピーする

ファイルシステムをディスク間でコピーするには、次の2つのコマンドを使用します。

- volcopy
- dd

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

次の節では、dd コマンドを使用してファイルシステムをディスク間でコピーする方法について説明します。

## ファイルシステムのリテラルコピーを作成する

dd コマンドでは、UFS ファイルシステムのリテラル (ブロックレベル) コピーを別のファイルシステムやテープに作成します。デフォルトでは、dd コマンドはその標準入力を標準出力にコピーします。

---

注 - 可変長テープドライブで dd コマンドを使用するときは、必ず適切なブロックサイズを指定してください。

---

標準入力、標準出力、またはその両方の代わりに、デバイス名を指定できます。次の例では、フロッピーディスクの内容が /tmp ディレクトリ内のファイルにコピーされます。

```
$ dd < /floppy/floppy0 > /tmp/output.file
2400+0 records in
2400+0 records out
```

dd コマンドは、読み取りブロック数と書き込みブロック数をレポートします。+ の次の数値は、部分的にコピーされたブロックの数です。デフォルトのブロックサイズは 512 バイトです。

dd コマンドの構文は、他のほとんどのコマンドとは異なっています。オプションは *keyword=value* のペアで指定します。この場合、*keyword* は設定するオプションで、*value* はそのオプションの引数です。たとえば、標準入力と標準出力を次の構文に置き換えることができます。

```
$ dd if=input-file of=output-file
```

上記の例のリダイレクト記号の代わりに *keyword=value* の形式で指定するには、次のように入力します。

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

## ▼ ディスクをコピーする方法 (dd)

手順 1. コピー元とコピー先のディスクが、同じディスクジオメトリを保持していることを確認します。

2. スーパーユーザーになるか、同等の役割を引き受けます。

3. **/reconfigure** ファイルを作成し、リブート時にシステムが追加するコピー先ディスクを認識するようにします。

```
# touch /reconfigure
```

4. システムをシャットダウンします。

```
# init 0
```

5. コピー先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. コピー元ディスクをコピー先ディスクにコピーします。

```
# dd if=/dev/rdisk/device-name of=/dev/rdisk/device-name bs=block-size
```

*if=/dev/rdisk/device-name*    マスターディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。

*of=/dev/rdisk/device-name*    コピー先ディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。

`bs=blocksize`                      ブロックサイズ (128K バイト、256K バイトなど) を指定できる。ブロックサイズの値を大きくすると、ディスクのコピーに要する時間を短縮できる。

詳細については、`dd(1M)` のマニュアルページを参照してください。

8. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

9. コピー先ディスクのルート (/) ファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /mnt
```

10. `/etc/vfstab` ファイルのあるディレクトリに移動します。

```
# cd /mnt/etc
```

11. テキストエディタを使用して、コピー先ディスクの `/etc/vfstab` ファイルを編集して、正しいデバイス名を参照するようにします。

たとえば、`c0t3d0` のインスタンスをすべて `c0t1d0` に変更します。

12. コピー先ディスクのルート (/) ディレクトリに移動します。

```
# cd /
```

13. コピー先ディスクのルート (/) ファイルシステムをマウント解除します。

```
# umount /mnt
```

14. システムをシャットダウンします。

```
# init 0
```

15. コピー先ディスクからシングルユーザーモードでブートします。

```
# boot disk# -s
```

---

注 - `installboot` コマンドをコピー先ディスクで実行する必要はありません。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。

---

16. コピー先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。

17. 再びコピー先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 disk1
```

18. システムがブートしたら、スーパーユーザーとしてログインしてシステム情報を確認します。

```
hostname console login:
```

### 例 27-1 ディスクをコピーする (dd)

次の例では、マスターディスク /dev/rdisk/c0t0d0s2 をコピー先ディスク /dev/rdisk/c0t2d0s2 にコピーする方法を示します。

```
# touch /reconfigure
# init 0
ok boot
# dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2 bs=128k
# fsck /dev/rdisk/c0t2d0s2
# mount /dev/dsk/c0t2d0s2 /mnt
# cd /mnt/etc
# vi vfstab
(新しいディスクのエントリを変更)
# cd /
# umount /mnt
# init 0
# boot disk2 -s
# sys-unconfig
# boot disk2
```

---

## cpio を使用してファイルシステム間でディレクトリをコピーする

cpio (コピーインとコピーアウト) コマンドを使用して、個々のファイル、ファイルグループ、またはファイルシステム全体をコピーできます。この節では、cpio コマンドを使用してファイルシステム全体をコピーする方法について説明します。

cpio コマンドは、ファイルのリストを1つの大型出力ファイルにコピーするアーカイブプログラムです。また、復元しやすいように、個々のファイルの間にヘッダーを挿入します。cpio コマンドを使用すると、ファイルシステム全体を別のスライス、別のシステム、またはテープやフロッピーディスクなどのメディアデバイスにコピーできます。

cpio コマンドは、メディアの終わりを認識し、別のボリュームを挿入するように促すプロンプトを表示するので、複数のテープやフロッピーディスクが必要なアーカイブを作成するにはもっとも効率のよいコマンド(ufsdump 以外では)です。

cpio の使用時には、しばしば `ls` や `find` のコマンドを使用してコピーするファイルを選択し、その出力を `cpio` コマンドにパイプします。

## ▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)

- 手順
1. スーパーユーザーになるか、同等の役割を引き受けます。
  2. 目的のディレクトリに移動します。

```
# cd /filesystem1
```

3. `find` コマンドと `cpio` コマンドを組み合わせ実行し、ディレクトリツリーを `filesystem1` から `filesystem2` へコピーします。

```
# find . -print -depth | cpio -pdm filesystem2
```

`.` 現在の作業ディレクトリで処理を始める。

`-print` ファイル名を出力する。

`-depth` ディレクトリ階層を下降し、すべてのファイル名を出力する。

`-p` ファイルのリストを作成する。

`-d` 必要に応じてディレクトリを作成する。

`-m` ディレクトリ上で正しい変更時刻を設定する。

詳細については、`cpio(1)` のマニュアルページを参照してください。

指定したディレクトリ名からファイルがコピーされます。シンボリックリンクは保持されます。

また、`-u` オプションも指定できます。このオプションは、無条件にコピーを実行します。`-u` オプションを指定しない場合、古いファイルが、新しいファイルで置換されません。このオプションは、ディレクトリごとコピーしたいときに、コピーするファイルの一部がすでにターゲットのディレクトリ中に存在する場合に便利です。

4. コピー先ディレクトリの内容を表示して、コピーに成功したかどうかを確認します。

```
# cd filesystem2
```

```
# ls
```

5. ソースディレクトリが不要な場合は削除します。

```
# rm -rf /filesystem1
```

## 例 27-2 ファイルシステム間でディレクトリをコピーする (cpio)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

# ファイルとファイルシステムをテープにコピーする

tar、pax、および cpio コマンドを使用すると、ファイルとファイルシステムをテープにコピーできます。どのコマンドを選択するかは、コピーする目的に応じて異なります。3つのコマンドはすべて raw デバイスを使用するので、使用する前にテープ上でファイルシステムをフォーマットまたは作成する必要はありません。

表 27-3 tar、pax、cpio コマンドの長所と短所

| コマンド | 機能                                                                                           | 長所                                                                                                                     | 短所                                                                                                                                                                                           |
|------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tar  | ファイルやディレクトリサブツリーを1本のテープにコピーする場合に使用する。                                                        | <ul style="list-style-type: none"><li>■ ほとんどの UNIX オペレーティングシステムで利用できる。</li><li>■ パブリックドメインバージョンもすぐに利用できる。</li></ul>    | <ul style="list-style-type: none"><li>■ ファイルシステムの境界を認識しない。</li><li>■ 絶対パス名の長さが255文字を超えることができない。</li><li>■ 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。</li><li>■ 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。</li></ul> |
| pax  | 複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用する。または、POSIX 準拠システムとの間でファイルをコピーする場合に使用する。 | <ul style="list-style-type: none"><li>■ POSIX 準拠システムに対する互換性は、tar コマンドや cpio コマンドよりもよい。</li><li>■ 複数のベンダーサポート</li></ul> | tar コマンドの短所を参照。ただし、pax は、複数のテープボリュームを作成できる。                                                                                                                                                  |



表 27-3 tar、pax、cpio コマンドの長所と短所 (続き)

| コマンド | 機能                                                                                                               | 長所                                                                                                                                                                                                                                        | 短所                                |
|------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| cpio | 複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用する。または、SunOS 5.9 システムから SunOS 4.0/4.1 システムにファイルをコピーする場合に使用する。 | <ul style="list-style-type: none"> <li>■ tar コマンドよりも効率的に、データをテープに書き込む。</li> <li>■ 復元時、テープ中の不良箇所をスキップする。</li> <li>■ 異なるシステムタイプ間の互換性のために、異なるヘッダー形式 (tar、ustar、crc、odc、bar など) でファイルを書き込むオプションを提供する。</li> <li>■ 複数のテープボリュームを作成する。</li> </ul> | コマンド構文が tar コマンドや pax コマンドよりも難しい。 |

使用するテープドライブとデバイス名は、各システムのハードウェアの構成によって異なります。テープデバイス名の詳細については、473 ページの「使用するメディアの選択」を参照してください。

## tar を使用してファイルをテープにコピーする

tar コマンドを使用してファイルをテープにコピーする前に、次の情報について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。
- ファイルをコピーするとき、ファイル名の一部にファイル名置換ワイルドカード文字 (? と \*) を使用できる。たとえば、接尾辞 `.doc` が付いたすべての文書をコピーするには、ファイル名引数として `*.doc` と入力する。
- tar アーカイブからファイルを抽出するときには、ファイル名置換ワイルドカードは使用できない。

## ▼ ファイルをテープにコピーする方法 (tar)

- 手順
1. コピーするファイルの存在するディレクトリに移動します。
  2. 書き込み可能なテープをテープドライブに挿入します。
  3. ファイルをテープにコピーします。

```
$ tar cvf /dev/rmt/n filenames
```

c                   アーカイブの作成を指定する。

v                   各ファイルがアーカイブされるたびに、その名前を表示する。

f /dev/rmt/n       アーカイブを指定したデバイスまたはファイルに書き込むように指定する。

filenames         コピーするファイルとディレクトリを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

指定した名前のファイルがテープにコピーされ、テープ上の既存のファイルが上書きされます。

4. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。
5. コピーしたファイルがテープ上に存在することを確認します。

```
$ tar tvf /dev/rmt/n
```

tar テープ上のファイルを表示する方法については、459 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

### 例 27-3 ファイルをテープにコピーする (tar)

次の例では、3つのファイルをテープドライブ0のテープにコピーします。

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/0
```

## ▼ テープ上のファイルのリストを表示する方法 (tar)

- 手順 1. テープをテープドライブに挿入します。  
2. テープの内容を表示します。

```
$ tar tvf /dev/rmt/n
```

t テープ上のファイルの内容一覧が表示される。

v t オプションと一緒に使用すると、テープ上のファイルに関する詳細情報が表示される。

f /dev/rmt/n テープデバイスを示す。

### 例 27-4 テープ上のファイルのリストを表示する (tar)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ tar tvf /dev/rmt/0
drwxr-xr-x 1001/10      0 Oct  7 08:18 2003 reports/
-r--r--r-- 1001/10    382 Oct  7 08:18 2003 reports/reportA
-r--r--r-- 1001/10    382 Oct  7 08:18 2003 reports/reportB
-r--r--r-- 1001/10    382 Oct  7 08:18 2003 reports/reportC
```

## ▼ テープからファイルを取り出す方法 (tar)

- 手順 1. ファイルを置きたいディレクトリに移動します。  
2. テープをテープドライブに挿入します。  
3. テープからファイルを取り出します。

```
$ tar xvf /dev/rmt/n [filenames]
```

x 指定したアーカイブファイルからのファイルの抽出を指定する。指定したドライブのテープに含まれるすべてのファイルが現在のディレクトリにコピーされる。

v 各ファイルを取り出すたびに、その名前を表示する。

f /dev/rmt/n アーカイブを含むテープデバイスを示す。

*filenames* 取り出すファイルを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

詳細については、tar (1) のマニュアルページを参照してください。

4. ファイルがコピーされていることを確認します。

```
$ ls -l
```

### 例 27-5 テープ上のファイルを取り出す (tar)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

#### 注意事項

---

注 - テープから抽出されるファイル名は、アーカイブに格納されているファイル名と同一でなければなりません。ファイルの名前やパス名が不明な場合は、まずテープ上のファイルのリストを表示します。テープ上のファイルをリスト表示する方法については、459 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

---

## pax コマンドを使用してファイルをテープにコピーする

### ▼ ファイルをテープにコピーする方法 (pax)

- 手順
1. コピーするファイルの存在するディレクトリに移動します。
  2. 書き込み可能なテープをテープドライブに挿入します。
  3. ファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/n filenames
```

-w 書き込みモードを有効にする。

-f /dev/rmt/n テープドライブを識別する。

filenames コピーするファイルとディレクトリを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。

詳細については、pax(1)のマニュアルページを参照してください。

4. ファイルがテープにコピーされていることを確認します。

```
$ pax -f /dev/rmt/n
```

5. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

#### 例 27-6 ファイルをテープにコピーする (pax)

次の例は、pax コマンドを使用して、現在のディレクトリ内のファイルをすべてコピーする方法を示します。

```
$ pax -w -f /dev/rmt/0 .  
$ pax -f /dev/rmt/0  
filea fileb filec
```

---

## cpio コマンドを使用してファイルをテープにコピーする

### ▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)

- 手順
1. コピーするファイルの存在するディレクトリに移動します。
  2. 書き込み可能なテープをテープドライブに挿入します。
  3. ファイルをテープにコピーします。

```
$ ls | cpio -oc > /dev/rmt/n
```

ls cpio コマンドにファイル名のリストを渡す。

cpio -oc cpio コマンドがコピーアウトモード (-o) で動作し、ASCII 文字形式 (-c) でヘッダー情報を書き込むように指定する。これらのオプションにより他のベンダーのシステムとの互換性を保つ。

> /dev/rmt/n 出力ファイルを指定する。

ディレクトリ内のすべてのファイルは、指定したドライブ内のテープにコピーされ、テープ上の既存のファイルが上書きされます。コピーされた合計ブロック数が表示されます。

4. ファイルがテープにコピーされていることを確認します。

```
$ cpio -civt < /dev/rmt/n
```

-c                    ファイルを ASCII 文字形式で読み込むように指定する。

-i                    cpio がコピーインモードで動作することを指定する。この時点ではファイルをリストするだけ。

-v                    ls -l コマンドの出力と同様の形式で出力を表示する。

-t                    指定したテープドライブ内のテープ上にあるファイルの内容一覧を表示する。

< /dev/rmt/n        既存の cpio アーカイブの入力ファイルを指定する。

5. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

#### 例 27-7 ディレクトリ内のすべてのファイルをテープにコピーする (cpio)

次の例では、/export/home/kryten ディレクトリ内のすべてのファイルをテープドライブ 0 のテープにコピーする方法を示します。

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
16 blocks
$ cpio -civt < /dev/rmt/0
-r--r--r--  1 kryten  staff          76 Oct  7 08:17 2003, filea
-r--r--r--  1 kryten  staff          76 Oct  7 08:17 2003, fileb
-r--r--r--  1 kryten  staff          76 Oct  7 08:17 2003, filec
drwxr-xr-x  2 kryten  staff           0 Oct  7 08:17 2003, letters
drwxr-xr-x  2 kryten  staff           0 Oct  7 08:18 2003, reports
16 blocks
$
```

## ▼ テープ上のファイルのリストを表示する方法 (cpio)

---

注 - テープの内容一覧を表示するには、cpio コマンドがアーカイブ全体を処理する必要があるため、かなりの時間がかかります。

---

- 手順 1. テープをテープドライブに挿入します。

2. テープ上のファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/n
```

### 例 27-8 テープ上のファイルのリストを表示する (cpio)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/0
-r--r--r-- 1 kryten  staff      76 Oct  7 08:17 2003, filea
-r--r--r-- 1 kryten  staff      76 Oct  7 08:17 2003, fileb
-r--r--r-- 1 kryten  staff      76 Oct  7 08:17 2003, filec
drwxr-xr-x 2 kryten  staff       0 Oct  7 08:17 2003, letters
drwxr-xr-x 2 kryten  staff       0 Oct  7 08:18 2003, reports
16 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。

---

- 手順
1. ファイルを置きたいディレクトリに移動します。
  2. テープをテープドライブに挿入します。
  3. テープからすべてのファイルを取り出します。

```
$ cpio -icvd < /dev/rmt/n
```

- i 標準入力からファイルを取り出す。
- c ファイルを ASCII 文字形式で読み込むように指定する。
- v 取り出されたファイルを `ls` コマンドの出力と同様の形式で表示する。
- d 必要に応じてディレクトリを作成する。

< /dev/rmt/n 出力ファイルを指定する。

4. ファイルがコピーされたことを確認します。

```
$ ls -l
```

### 例 27-9 テープからすべてのファイルを取り出す (cpio)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

## ▼ テープから特定のファイルを取り出す方法 (cpio)

- 手順
1. ファイルを置きたいディレクトリに移動します。
  2. テープをテープドライブに挿入します。
  3. テープからファイルのサブセットを取り出します。

```
$ cpio -icv "*file" < /dev/rmt/n
```

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| -i             | 標準入力からファイルを取り出す。                                                                  |
| -c             | ヘッダーを ASCII 文字形式で読み込むように指定する。                                                     |
| -v             | 取り出されたファイルを ls コマンドの出力と同様の形式で表示する。                                                |
| <i>"*file"</i> | パターンに一致するすべてのファイルを現在のディレクトリにコピーするように指定する。複数のパターンを指定できるが、個々のパターンを二重引用符で囲まなければならない。 |

< /dev/rmt/n 入力ファイルを指定する。

詳細については、cpio(1)のマニュアルページを参照してください。

4. ファイルがコピーされたことを確認します。

```
$ ls -l
```



### 例 27-10 指定したファイルをテープから取り出す (cpio)

次の例では、末尾に接尾辞 `chapter` が付いているすべてのファイルをドライブ 0 のテープから取り出します。

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

---

## ファイルをリモートテープデバイスにコピーする

### ▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd)

- 手順 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を実行するユーザーのユーザー名) が、リモートシステムの `/etc/hosts.equiv` ファイルに記述されている必要がある。または、コピーを実行するユーザーが、リモートマシン上の自分のホームディレクトリにアクセス可能にし、かつ `$HOME/.rhosts` 内にローカルマシン名を記述しておく必要がある。  
詳細については、`hosts.equiv(4)` のマニュアルページを参照してください。
  - リモートシステムのエントリがローカルシステムの `/etc/inet/hosts` ファイル内またはネームサービスの `hosts` ファイル内になければならない。
2. リモートコマンドの実行に必要なアクセス権を保持していることを確認するには、次のように入力します。

```
$ rsh remotehost echo test
```

test と表示された場合、リモートコマンドの実行に必要なアクセス権を保持しています。Permission denied と表示された場合は、手順 1 の内容を確認してください。

3. ファイルを置きたいディレクトリに移動します。
4. テープをテープドライブに挿入します。
5. ファイルをリモートテープドライブにコピーします。

```
$ tar cvf - filenames | rsh remote-host dd of=/dev/rmt/n obs=block-size
```

|                      |                                                  |
|----------------------|--------------------------------------------------|
| tar cf               | テープアーカイブを作成し、アーカイブに含まれるファイルをリスト表示し、テープデバイスを指定する。 |
| v                    | tar ファイルのエントリに関する追加情報を表示する。                      |
| - (ハイフン)             | 可変部としてテープデバイスの代わりに指定する。                          |
| filenames            | コピーするファイルを指定する。ファイルが複数の場合は、各ファイルをスペースで区切る。       |
| rsh remote-host      | tar コマンドの出力をパイプを通してリモートシェルに渡す。                   |
| dd of=<br>/dev/rmt/n | 出力デバイスを指定する。                                     |
| obs=block-size       | ブロック係数を指定する。                                     |

6. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

#### 例 27-11 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 obs=126b
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

## ▼ ファイルをリモートテープデバイスから取り出す方法

手順 1. テープをテープドライブに挿入します。

2. 一時ディレクトリに移動します。

```
$ cd /var/tmp
```

3. ファイルをリモートテープドライブから取り出します。

```
$ rsh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

*rsh remote-host* dd コマンドを使用してテープデバイスからファイルを取り出すために起動するリモートシェル。

*dd if=/dev/rmt/n* 入力デバイスを指定する。

*| tar xvBpf -* dd コマンドの出力を tar コマンドにパイプして、ファイルを復元する。

4. ファイルが抽出されたことを確認します。

```
$ ls -l /var/tmp
```

### 例 27-12 ファイルをリモートのテープドライブから抽出する

```
$ cd /var/tmp
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```

---

## ファイルとファイルシステムをフロッピーディスクにコピーする

ファイルやファイルシステムをフロッピーディスクにコピーする前に、フロッピーディスクをフォーマットする必要があります。フロッピーディスクをフォーマットする方法については、第3章を参照してください。

tar コマンドを使用して、UFS ファイルを1枚のフォーマット済みフロッピーディスクにコピーします。

UFS ファイルを複数のフォーマット済みフロッピーディスクにコピーする必要がある場合は、cpio コマンドを使用します。cpio コマンドはメディアの終わりを認識し、次のフロッピーディスクの挿入を促すプロンプトを表示します。

### ファイルをフロッピーディスクにコピーする際の注意事項

- tar に `-c` オプションを指定してフォーマット済みフロッピーディスクにファイルをコピーすると、フロッピーディスク上の既存のファイルは破壊 (上書き) される。
- tar イメージを格納済みのフロッピーディスクはマウントできない。
- 複数のボリュームを扱う場合は、cpio コマンドを使用してください。tar コマンドは1つのボリュームに対して使用できるユーティリティです。

詳細については、tar(1) のマニュアルページを参照してください。

### ▼ ファイルを1枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)

- 手順
1. コピーするファイルの存在するディレクトリに移動します。
  2. 書き込み保護されていないフォーマット済みフロッピーディスクをドライブに挿入します。
  3. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

4. 必要に応じて、再度フォーマットします。

```
$ rmformat -U /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
```

5. ファイルをフロッピーディスクにコピーします。

```
$ tar cvf /vol/dev/aliases/floppy0 filenames
```

指定した名前のファイルがフロッピーディスクにコピーされ、フロッピーディスク上の既存のファイルがすべて上書きされます。

6. ファイルがコピーされたことを確認します。

```
$ tar tvf /vol/dev/aliases/floppy0
```

ファイルのリストを表示する方法については、469 ページの「フロッピーディスク上のファイルのリストを表示する方法 (tar)」を参照してください。

7. フロッピーディスクをドライブから取り出します。

8. ファイル名をフロッピーディスクラベルに記入します。

#### 例 27-13 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、evaluation\* という名前のファイルをフロッピーディスクにコピーする方法を示します。

```
$ cd /home/smith
$ volcheck
$ ls evaluation*
evaluation.doc  evaluation.doc.backup
$ tar cvf /vol/dev/aliases/floppy0 evaluation*
a evaluation.doc 86 blocks
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/aliases/floppy0
```

### ▼ フロッピーディスク上のファイルのリストを表示する方法 (tar)

- 手順 1. フロッピーディスクをドライブに挿入します。

2. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

3. フロッピーディスク上のファイルのリストを表示します。

```
$ tar tvf /vol/dev/aliases/floppy0
```

例 27-14 フロッピーディスク上のファイルのリストを表示する (tar)

次の例では、フロッピーディスク上のファイルのリストを表示します。

```
$ volcheck
$ tar tvf /vol/dev/aliases/floppy0
rw-rw-rw-6693/10  44032 Jun  9 15:45 evaluation.doc
rw-rw-rw-6693/10  43008 Jun  9 15:55 evaluation.doc.backup
$
```

## ▼ ファイルをフロッピーディスクから取り出す方法 (tar)

手順 1. ファイルを置きたいディレクトリに移動します。

2. フロッピーディスクをドライブに挿入します。

3. フロッピーディスクを使用可能な状態にします。

```
$ volcheck
```

4. ファイルをフロッピーディスクから取り出します。

```
$ tar xvf /vol/dev/aliases/floppy0
```

フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

5. ファイルが取り出されたことを確認します。

```
$ ls -l
```

6. フロッピーディスクをドライブから取り出します。

例 27-15 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。

```
$ cd /home/smith/Evaluations
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

次の例では、フロッピーディスクから個々のファイルを取り出します。指定したファイルは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。

```
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0 evaluation.doc
x evaluation.doc, 44032 bytes, 86 tape blocks
$ ls -l
```

## ファイルを複数のフロッピーディスクにアーカイブする方法

大量のファイルをフロッピーディスクにコピーする場合は、いっぱいになったフロッピーディスクを別のフォーマット済みフロッピーディスクと交換するように促すプロンプトを表示させることができます。cpio コマンドにはこの機能があります。使用する cpio コマンドはファイルをテープにコピーする場合と同じですが、テープデバイス名ではなくデバイスとして /vol/dev/aliases/floppy0 を指定します。

cpio コマンドの使用方法については、461 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。





## 第 28 章

---

# テープドライブの管理 (手順)

---

この章では、Solaris™ オペレーティングシステムで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。

- 476 ページの「テープドライブの状態を表示する方法」
- 477 ページの「磁気テープカートリッジのたるみを直す方法」
- 478 ページの「磁気テープカートリッジを巻き戻す方法」

この章で説明する主な内容は次のとおりです。

- 473 ページの「使用するメディアの選択」
- 474 ページの「バックアップデバイス名」
- 476 ページの「テープドライブの状態を表示する」
- 478 ページの「ドライブの管理とメディア処理のガイドライン」

---

## 使用するメディアの選択

通常は、次のテープメディアを使用して Solaris システムのバックアップを作成します。

- 1/2 インチのリールテープ
- 1/4 インチのストリームカートリッジテープ
- 8mm のカートリッジテープ
- 4mm のカートリッジテープ (DAT)

フロッピーディスクを使用してバックアップを実行することもできますが、時間がかかり煩雑です。

どのメディアを選択するかは、メディアをサポートする機器とファイルの格納に使用するメディア (通常はテープ) の可用性によって決まります。バックアップはローカルシステムから実行しなければなりません、ファイルはリモートデバイスに書き込まれます。

次の表に、ファイルシステムのバックアップに使用する標準的なテープデバイスを示します。各デバイスの記憶容量は、ドライブのタイプとテープに書き込むデータのタイプによって異なります。

表 28-1 メディアの記憶容量

| バックアップメディア                        | 記憶容量               |
|-----------------------------------|--------------------|
| 1/2 インチのリールテープ                    | 140M バイト (6250bpi) |
| 2.5G バイト、1/4 インチのカートリッジ (QIC) テープ | 2.5G バイト           |
| DDS3 4mm カートリッジテープ (DAT)          | 12 - 24G バイト       |
| 14G バイト、8mm カートリッジテープ             | 14G バイト            |
| DLT 7000 1/2 インチ カートリッジテープ        | 35 - 70G バイト       |

## バックアップデバイス名

バックアップに使用するテープまたはフロッピーディスクドライブに論理デバイス名を与えて指定します。この名前は、「raw」デバイスファイルの格納されたサブディレクトリを指し、ドライブの論理ユニット番号が含まれます。テープドライブの命名規則に従い、物理デバイス名ではなく論理デバイス名を使用します。次の表に、この命名方式を示します。

表 28-2 バックアップデバイスの基本的なデバイス名

| デバイスの種類   | 名前                                         |
|-----------|--------------------------------------------|
| テープ       | <code>/dev/rmt/n</code>                    |
| フロッピーディスク | <code>/vol/dev/rdiskette0/unlabeled</code> |

通常は、次の図に示す方法で、テープデバイスを指定します。

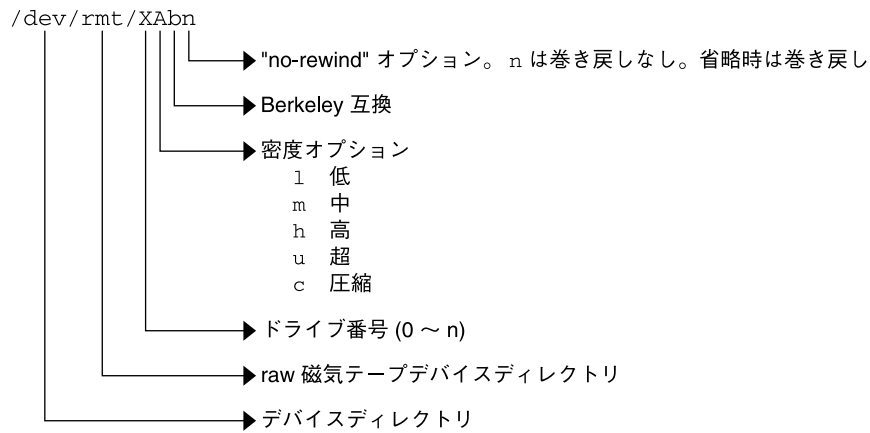


図 28-1 テープドライブデバイス名

密度を指定しないと、テープドライブは通常その「推奨」密度で書き込みます。推奨密度は、一般にそのテープドライブでサポートされる最大密度です。ほとんどの SCSI ドライブはテープ上の密度やフォーマットを自動的に検出し、それに従って読み取りを実行します。ドライブでサポートされる密度を調べるには、`/dev/rmt` サブディレクトリを確認してください。このサブディレクトリには、各テープで異なる出力密度をサポートするためのテープデバイスファイルのセットが含まれます。

SCSI コントローラは、最大 7 台の SCSI テープドライブを持つことができます。

## テープドライブの巻き戻しオプションを指定する

通常は、テープドライブを 0 から  $n$  までの論理デバイス番号で指定します。次の表に、「巻き戻し」または「巻き戻しなし」のオプションを付けてテープデバイス名を指定する方法を示します。

表 28-3 テープドライブの「巻き戻し」または「巻き戻しなし」オプション

| ドライブおよび巻き戻し      | 使用するオプション                |
|------------------|--------------------------|
| 第 1 のドライブ、巻き戻し   | <code>/dev/rmt/0</code>  |
| 第 1 のドライブ、巻き戻しなし | <code>/dev/rmt/0n</code> |
| 第 2 のドライブ、巻き戻し   | <code>/dev/rmt/1m</code> |
| 第 2 のドライブ、巻き戻しなし | <code>/dev/rmt/1n</code> |

## テープドライブに別の密度を指定する

デフォルトでは、テープドライブはその「推奨」密度で書き込みますが、これは一般にそのテープドライブでサポートされる最大密度です。テープデバイスを指定しなければ、コマンドはデバイスでサポートされるデフォルト密度でドライブ番号 0 に書き込みます。

テープを特定の密度しかサポートされないテープドライブが付いたシステムに転送するには、目的の密度で書き込むデバイス名を指定します。次の表に、テープドライブに別の密度を指定する方法を示します。

表 28-4 テープドライブに別の密度を指定する

| ドライブ、密度、巻き戻し         | 使用するオプション                 |
|----------------------|---------------------------|
| 第 1 のドライブ、低密度、巻き戻し   | <code>/dev/rmt/0l</code>  |
| 第 1 のドライブ、低密度、巻き戻しなし | <code>/dev/rmt/0ln</code> |
| 第 2 のドライブ、中密度、巻き戻し   | <code>/dev/rmt/1m</code>  |
| 第 2 のドライブ、中密度、巻き戻しなし | <code>/dev/rmt/1mn</code> |

密度のオプションについては、474 ページの「バックアップデバイス名」を参照してください。

## テープドライブの状態を表示する

`mt` コマンドの `status` オプションを使用すると、テープドライブに関する状態情報を表示できます。`mt` コマンドは、`/kernel/drv/st.conf` ファイルに記述されているすべてのテープドライブの情報を表示します。

### ▼ テープドライブの状態を表示する方法

- 手順 1. 情報を表示するドライブにテープをロードします。
2. テープドライブの状態を表示します。

```
# mt -f /dev/rmt/n status
```

3. テープドライブ番号を 0、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 28-1 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) の状態を示します。

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
```

次の例は、Exabyte テープドライブ (/dev/rmt/1) の状態を示します。

```
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0   retries= 0
file no= 0   block no= 0
```

次の方法を使用すると、システムを手早くポーリングしてすべてのテープドライブを検査できます。

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

---

## 磁気テープカートリッジの取り扱い

テープの読み込み中にエラーが発生した場合は、テープのたるみを直し、テープドライブを掃除してからやり直してください。

### 磁気テープカートリッジのたるみを直す方法

mt コマンドを使用して磁気テープカートリッジのたるみを直します。

たとえば、次のようになります。

```
$ mt -f /dev/rmt/1 retension
$
```

---

注 - QIC 以外のテープドライブのたるみは直さないでください。

---

## 磁気テープカートリッジを巻き戻す方法

磁気テープカートリッジを巻き戻すには、`mt` コマンドを使用します。

たとえば、次のようになります。

```
$ mt -f /dev/rmt/1 rewind
$
```

---

## ドライブの管理とメディア処理のガイドライン

バックアップテープは読み込めなければ役に立ちません。このため、定期的に掃除およびチェックを行い、テープドライブが正常に動作するようにしてください。テープドライブのクリーニング手順については、ハードウェアのマニュアルを参照してください。次のいずれかの方法で、テープハードウェアをチェックできます。

- テープにファイルをコピーし、ファイルを読み込んで、オリジナルのファイルとコピーしたファイルを比較する。
- `ufsdump` コマンドの `-v` オプションを使用して、メディアの内容をソースファイルシステムと比較して検査する。`-v` オプションを有効にするには、ファイルシステムをマウント解除するか、完全にアイドル状態にする必要がある。

ハードウェアは、システムからレポートされないような障害を起こす可能性があるので注意してください。

バックアップ後は、必ずテープにラベルを付けてください。第 22 章で説明したバックアップ方法を採用する場合は、ラベルを「テープ A」、「テープ B」という具合に指定する必要があります。このラベルは変更しないでください。バックアップを実行するたびに、次の情報を記入した別のテープラベルを作成します。

- バックアップ日付
- マシン名および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 (バックアップが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

テープは、磁気を発生させる機器から離れた埃のない安全な場所に保管してください。たとえば、アーカイブしたテープを遠隔地の防火キャビネットに保管します。

各ジョブ (バックアップ) がどのメディア (テープボリューム) に格納されているかということと、各バックアップファイルがどこに保管されているかを記録したログを作成し、管理する必要があります。





# 索引

---

## 数字・記号

- 4.3 Tahoe ファイルシステム, 257
- 9660 CD フォーマット, 32

## A

- AutoFS, 272-273

## B

- BSD Fat Fast File システム, 257

## C

- cachefspack コマンド
  - 概要, 316
  - 使用方法, 316
- CacheFS ファイルシステム
  - cachefspack エラーの障害追跡, 321
  - cachefspack コマンドによるバック (概要), 316
  - cachefspack コマンドによるバック (手順), 316
  - CacheFS 統計情報の表示, 329
  - CacheFS の統計情報の収集 (概要), 325
  - CacheFS ロギングの設定 (手順), 327
  - CacheFS ロギングの停止, 328
  - CacheFS ログファイルの場所の調査, 327
  - fsck コマンドによるチェック (例), 315
  - 概要, 303

## CacheFS ファイルシステム (続き)

- キャッシュへのファイルのバック (手順), 319
- 削除 (手順), 313
- 作成 (手順), 305
- 情報の表示 (手順), 311
- パッキングリストの作成 (手順), 319
- バックされたファイルの表示 (手順), 317
- バックされたファイルの表示 (例), 317
- パラメータ, 303
- ファイルのパッキング解除 (手順), 320
- マウント (手順), 306

## CD

- ISO 9660 フォーマット, 32
- UFS CD
  - SPARC vs. x86 フォーマット, 32
  - 名前, 30
- cdrw コマンド
  - CD メディアの確認 (手順), 61
  - アクセス制限 (手順), 60
  - 説明, 57
  - データ CD とオーディオ CD の書き込み (概要), 59
- cfgadm
  - PCI ホットプラグ (概要), 81
  - SCSI ホットプラグ (概要), 81
- cfsadmin コマンド, 305, 313
- clri コマンド, 262
- cpio コマンド
  - 概要, 454-456
  - テープからすべてのファイルを取り出す (手順), 463

cpio コマンド (続き)  
テープ上のファイルのリストを表示する (手順), 463  
ファイルシステム間でディレクトリをコピーする (手順), 455

## D

dd コマンド  
概要, 451  
ディスクのコピー (手順), 452  
/dev/dsk ディレクトリ, 157  
/dev/rdisk ディレクトリ, 157  
devfsadm コマンド, 156  
dfstab ファイル, 共有ローカルリムーバブルメディアの設定 (手順), 39  
df コマンド, 158, 262  
dmesg コマンド, 76  
SPARC の例, 76  
x86 の例, 76  
DOS, ファイルシステム, 257  
driver not attached メッセージ, 73  
DVD-ROM, 259

## E

eject コマンド, リムーバブルメディア (手順), 37  
/etc/dfs/dfstab ファイル, 共有ローカルリムーバブルメディアの設定 (手順), 39  
/etc/dumpdates ファイル, 436-437  
/etc/rmmount.conf ファイル, リムーバブルメディアドライブの共有 (手順), 39  
/export/home ディレクトリ, 264

## F

FDFS ファイルシステム, 261  
ff コマンド, 262  
FIFOFS ファイルシステム, 261  
FIFO の i ノード, 348  
format.dat ファイル  
エントリの作成 (概要), 198  
エントリの作成 (手順), 198  
キーワード, 239, 242

format.dat ファイル (続き)  
構文規則, 239  
format コーティリティ  
analyze メニュー, 236-237  
defect メニュー, 237-238  
fdisk メニュー, 235-236  
partition メニュー, 234-235, 235  
SCSI ディスクドライブの自動構成 (概要), 199  
SCSI ディスクドライブの自動構成 (手順), 201  
Solaris fdisk パーティションの作成 (手順), 221  
概要, 175  
機能と利点, 175  
コマンド名の入力方法, 244  
システム上のディスクの確認 (手順), 184  
システム上のディスクの確認 (例), 186  
使用上のガイドライン, 176-178  
使用する場合, 176  
情報を維持するための推奨事項, 231  
ディスクがフォーマット済みかを調べる (手順), 187  
ディスクスライス情報の表示 (例), 190  
ディスクスライスとディスクラベルの作成 (手順)  
SPARC, 211  
x86, 227  
ディスクのフォーマット (例), 188  
ディスクラベルの作成  
例, 193  
入力, 243, 245  
破損したディスクラベルの復元 (手順), 195  
ブロック番号の指定方法, 243  
ヘルプ機能の使用, 245  
メインメニュー, 232  
free hog スライス, 「提供側のスライス」を参照  
fsck コマンド, 158, 262  
FSACTIVE 状態フラグ, 344  
FSBAD 状態フラグ, 344  
FSCLEAN 状態フラグ, 344  
FSSTABLE 状態フラグ, 344  
構文とオプション, 361  
修復, 357  
修復する条件, 346  
状態フラグ, 344  
対話式で使用, 352

fsck コマンド (続き)  
  チェック  
    i ノードリストのサイズ, 347  
    空き i ノード数, 348  
    空きブロック数, 347  
    スーパーブロック, 347  
fsdb コマンド, 262  
fssnap コマンド, UFS スナップショットの作成 (手順), 414  
fstypes ファイル, 273  
fstyp コマンド, 262  
fuser コマンド  
  リムーバブルメディアが使用中かどうかの調査 (手順), 37  
  リムーバブルメディアにアクセスしているプロセスの強制終了 (手順), 37

## G

grep コマンド, 273

## H

High Sierra ファイルシステム, 257  
/home (自動マウント), 273  
HSFS, 「High Sierra ファイルシステム」を参照

## I

installboot コマンド, 216, 229  
ISO 9660 ファイルシステム, 257  
ISO 標準, 9660 CD フォーマット, 32  
i ノード, 373-374  
  FIFO, 348  
  キャラクタ型特殊ファイル, 348  
  形式とタイプチェック, 348  
  ごとのバイト数, 378  
  サイズ, 350  
  シンボリックリンク, 348  
  通常ファイル, 348  
  ディレクトリ, 348  
  不正な i ノード番号, 351  
  ブロック型特殊ファイル, 348  
  リンク数, 349  
i ノードの形式, 348

i ノードの状態, 348  
i ノードのタイプ, 348  
i ノードのリンク数, 349  
i ノードリストのサイズ, 347

## K

/kernel/drv ディレクトリ, 72

## L

labelit コマンド, 262  
lost+found ディレクトリ, 344

## M

media was found メッセージ, 49  
mkfile コマンド, 338, 339  
mkfs コマンド, 262, 276  
mkisofs コマンド, データ CD ファイルシステム (手順), 62  
MNTPFS ファイルシステム, 265  
mnttab ファイル, 270  
mountall コマンド, 262  
mount コマンド, 158  
mt コマンド, 477

## N

NAMEFS ファイルシステム, 261  
ncheck コマンド, 262  
newfs コマンド, 158, 276, 379-382  
NFS  
  vfstab エントリ, 290  
  サーバーの説明, 272  
  説明, 272  
nfsd デーモン  
  起動, 39  
  実行中かどうかの確認, 38  
no media was found メッセージ, 49

## O

/opt ディレクトリ, 264

## P

passwd ファイル, テープからの復元 (例), 428

PCFS ファイルシステム, 257

PCI デバイス

PCI アダプタカードの取り付け (手順), 98

PCI アダプタカードの取り外し (手順), 97

PCI 構成に関する問題の障害追跡, 99

PCMCIA メモリーカード

他のシステム上の PCMCIA メモリーカード  
へのアクセス (例), 43

リモートでのマウント (例), 43

PROCFS ファイルシステム, 概要, 260

/proc ディレクトリ, 260, 264

prtconf コマンド, 74

prtvtoe コマンド, 158

使用例, 194

## R

raw ディスクデバイスインタフェース, 157, 158

rmmount.conf ファイル, リムーバブルメ

ディアドライブの共有 (手順), 39

Rock Ridge 拡張 (HSFS ファイルシステム), 257

## S

SCSI ディスクドライブ, 199

SCSI テープドライブ, 475

SCSI デバイス

cfgadm コマンドによる切り離し (手順), 89

SCSI 構成に関する問題の障害追跡, 93

失敗した SCSI 構成解除操作の解決 (手順)  
, 95

shareall コマンド, 272

share コマンド, 272

リムーバブルメディアを他のシステムで使用  
可能にする (手順), 39

Solaris fdisk パーティション, ガイドライ  
ン, 220-221

SPARC システム, UFS フォーマット, 32

SPECFS ファイルシステム, 261

SunOS のデフォルトファイルシステ  
ム, 264-265

swapadd コマンド, 335

SWAPFS ファイルシステム, 261

swap コマンド, 338

sysdef コマンド, 74

## T

tar コマンド

概要, 457

テープからファイルを取り出す (手順), 459

テープ上のファイルのリストを表示する (手  
順), 459

ファイルをフロッピーディスクから取り出す  
(手順), 470

ファイルをリモートテープから取り出す (dd  
コマンド)(手順), 467

ファイルをリモートテープにコピーする (dd  
コマンド)(手順), 466

フロッピーディスク上のファイルのリストを  
表示する (手順), 469

TMPFS ファイルシステム, 概要, 259-260

/tmp ディレクトリ, 259, 264

## U

UDF ファイルシステム, 258-259

UFS CD, SPARC vs. x86 フォーマット, 32

ufsdump コマンド

オプションと引数, 440

完全バックアップの例, 405, 408

機能, 435-440

制限, 440

増分バックアップの例, 406

データのコピー方法, 436

メディアの終わりの検出, 436

ufsdump コマンド (概要), 404

ufsrestore コマンド, 443

使用するための準備 (概要), 422

使用するテープの決定 (手順), 424

UFS スナップショット

完全バックアップの作成 (手順), 417

作成 (手順), 414

説明, 413

UFS ファイルシステム, 257, 265-268

UFS ファイルシステム (続き)  
  /etc/vfstab を使用したマウント, 291  
  拡張基礎タイプ, 266  
  状態フラグ, 266  
  大規模ファイルシステム, 266  
  大規模ファイルを持たない UFS ファイルシステムのマウント (手順), 293  
  マウント, 290  
UFS ロギング、概要, 267  
umountall コマンド, 263  
umount コマンド, 263  
UNIX ファイルシステム, 257  
USB デバイス  
  Solaris USB アーキテクチャ (USB-A), 122  
  USB オーディオデバイス問題の障害追跡, 147  
  USB デバイスの構成 (手順), 151  
  USB デバイスのサブツリーの接続解除 (手順), 152  
  USB デバイスの接続 (手順), 152  
  USB デバイスのリセット (手順), 152  
  volld を使用して大容量ストレージデバイスをマウント解除する (手順), 135  
  volld を使用して大容量ストレージデバイスをマウントする (手順), 135  
  オーディオ  
    概要, 144  
    主デバイスの変更 (手順), 147  
    デバイスの所有権, 147  
  概要, 119  
  キーボードとマウスデバイス, 123  
  ケーブル, 126  
  合成デバイス, 121  
  サポート, 114  
  ストレージデバイス, 129  
  デバイスクラス, 122  
  デバイス情報の表示 (手順), 132  
  デバイスノード, 122  
  電源管理, 126  
  ドライバ, 122  
  名前, 121  
  バスの説明, 120  
  複合デバイス, 121  
  物理デバイス階層, 121  
  ホストコントローラとルートハブ, 124  
  ホットプラグ (概要), 139  
  略語, 120  
/usr ファイルシステム, 264

**V**  
  /var ディレクトリ, 264  
  vfstab ファイル, 273, 335  
    LOFS のエントリ, 281  
    エントリの追加 (手順), 289  
    すべてのファイルをマウント, 290  
    スワップの追加, 335  
    デフォルト, 271  
  volcopy コマンド, 263  
  volmgt start コマンド, 34

**X**  
  x86 システム, UFS フォーマット, 32

**あ**  
  アーカイブ, ファイルを複数のフロッピーディスクにアーカイブする (cpio)(手順), 471  
  空き i ノード数, 348  
  空きブロック, 374-375  
  空きブロック数, 347  
  空き容量 (最小), 377  
  アクセス  
    ディスクデバイス, 157, 160  
    テープデバイス, 160  
    リムーバブルメディア (手順), 34  
    リムーバブルメディアのパス名, 27

**い**  
  一時ファイルシステム (TMPFS), 概要, 259-260

**お**  
  オプション, ufsdump コマンド, 440

**か**  
  解決, 失敗した SCSI 構成解除操作 (手順), 95  
  書き込み, データ CD とオーディオ CD (概要), 59

拡張基礎タイプ (UFS ファイルシステム), 266  
確認

- CD メディア (手順), 61
- nfsd デーモンが実行中かどうか, 38
- システム上のディスク (手順), 185
- ファイルシステム名, 423

仮想ファイルシステムテーブル, 271

仮想メモリー記憶域、定義, 332

間接ブロック, 350

完全バックアップ

- 定義, 388
- 例, 405, 408

## き

記憶域 (仮想メモリー)、定義, 332

記憶容量 (メディア), 388, 474

疑似ファイルシステム, 概要, 258-261

起動

- nfsd デーモン, 39
- ボリューム管理 (手順), 34

キャラクタ型特殊ファイルの i ノード, 348

強制終了

- ファイルシステムを使用中のすべてのプロセス (手順), 297
- リムーバブルメディアにアクセスしているプロセス (手順), 37

共有, ファイル, 272

切り離し, SCSI コントローラ (手順), 89

記録

- 増分バックアップ, 437
- ダンプ, 436-437

## け

決定

- テープデバイスのタイプ, 423
- テープデバイス名, 423
- ファイルシステムのタイプ, 273

検出、メディアの終わり, `ufsdump` コマンド, 436

## こ

構成, USB デバイス (手順), 151

構造、シリンダグループの, 372-375

構文

- `fsck` コマンド, 361
- `newfs`, 379-382

コピー

- `cpio` コマンドを使用して個々のファイルをコピーする (概要), 454
- `cpio` コマンドを使用してファイルグループをコピーする (概要), 454
- `cpio` コマンド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 (概要), 454
- ファイルシステム全体 (`dd`), 451
- ファイルをフロッピーディスクにコピーする (手順), 468

## さ

再構成ブート, 200

- SPARC の例, 210
- x86 の例, 220

最小空き容量, 377

サイズ

- i ノード, 350
- ファイルシステムのチェック, 347
- フラグメント, 376-377

最適化のタイプ, 377-378

削除

- CacheFS ファイルシステム (手順), 313
- UFS スナップショット情報
  - 例, 416
- 不要なスワップファイル, 340

作成

- UFS スナップショット
    - 例, 414
  - UFS スナップショット情報の完全バックアップ (手順), 417
  - UFS スナップショット (手順), 414
  - スワップファイル, 338
  - データ CD ファイルシステム (手順), 62
  - パッキングリスト (手順), 319
  - ファイルシステム (概要), 276
  - ループバックファイルシステム (概要), 280
- サポートされていないデバイス, 72

## し

時間 (最適化のタイプ), 377-378  
識別, デバイス, 73  
システムディスク  
  接続 (手順)  
    x86, 219  
    説明, 174  
自動構成プロセス, 71  
自動マウント, /home, 273  
収集, CacheFS の統計情報 (概要), 325  
重複ブロック, 349  
終了

  ファイルシステムに対するすべてのプロセス (手順), 297  
  ボリューム管理 (手順), 33  
  リムーバブルメディアにアクセスしているプロセスの強制終了 (手順), 37

## 準備

  バックアップ, 402-403  
  ファイルの復元 (概要), 422-423

## 障害追跡

  cachefspack エラー, 321  
  PCI 構成に関する問題, 99  
  SCSI 構成に関する問題, 93  
  USB オーディオデバイス問題, 147

## 状態フラグ

  fsck, 344  
  UFS ファイルシステム, 266

シリンダグループ, 372-375

シンボリックリンク, 348

## す

スーパーブロック, 347, 359, 372-373  
スライス (定義), 169  
スワップパーティション、定義, 332  
スワップファイル  
  作成, 338  
  追加vfstab, 335  
  表示, 336-338  
  不要なスワップファイルの削除, 340

## せ

制限, リムーバブルメディアへのアクセス (手順), 60

接続, USB デバイス (手順), 152  
接続解除, USB デバイスのサブツリー (手順), 152  
設定, CacheFS ログイン, 327

## そ

増分バックアップ, 388, 437  
  例, 406

## た

大規模ファイルオプション, 286  
ダンプレベル  
  定義, 393  
  日単位増分バックアップ, 393

## ち

### チェック

  CacheFS ファイルシステム (例), 315  
  i ノードの形式とタイプ, 348  
  i ノードリストの整合性, 347  
  空き i ノード数, 348  
  空きブロック数, 347  
  ファイルシステムのサイズ, 347  
  ファイルシステムの修復, 352

### 調査

  ファイルシステムのタイプ, 273  
  マウント済みのファイルシステム, 289

直接入出力, 268

## つ

### 追加

  エントリ/etc/vfstab ファイル (手順), 289  
  スワップvfstab, 335  
  ディスク (概要)  
    SPARC, 208  
    x86, 218-229

通常ファイル i ノード, 348

## て

停止, CacheFS ロギング, 328

### ディスク

SCSI ドライブの自動構成, 199

欠陥セクターの修復, 201, 203

追加 (概要)

SPARC, 208

x86, 218-229

ディスクスライスとディスクラベルの作成  
(手順)

SPARC, 211

ディスクスライスとディスクラベルの作成  
(例)

SPARC, 212

二次ディスクの接続 (例)

SPARC, 213

破損したディスクラベルの復元 (概要), 195

破損したディスクラベルの復元 (手順), 195

フォーマット (概要), 178

フォーマット済みかを調べる (手順), 187

フォーマットする場合 (概要), 186

ディスクコントローラ, 158

### ディスクスライス

システム構成の要件, 174

使用するスライスの決定, 174

情報の表示 (概要), 189-191

定義, 169

ディスクスライスの指定, 158, 160

ディスクのフォーマット, 概要, 178

ディスクベースのファイルシステム, 256-257

### ディスクラベル

作成 (概要), 191

説明, 178

### ディレクトリ

cpio を使用してファイルシステム間でコ  
ピーする (概要), 454

i ノード, 348

/proc, 260

/tmp, 259

未割り当てブロック, 351

ディレクトリデータブロック, 350

データブロック, 352, 374

### テープ

記憶容量, 388, 474

サイズ, 388, 474

性質, 440

ファイルを取り出す (tar コマンド)(手順)  
, 459

### テープ (続き)

容量, 440

テープデバイス, 復元するタイプの決定, 423

テープデバイス (命名), 161

### テープドライブ

管理, 478-479

最大の SCSI テープドライブ台数, 475

巻き戻し, 475-476

テープドライブの管理, 478-479

デバイス, アクセス, 155

### デバイスドライバ

追加, 79

定義, 71

デバイスのインスタンス名, 156

### デバイス名

テープデバイス名の決定, 423

バックアップ, 474-476

ファイルシステム名の確認, 423

### デフォルト

SunOS ファイルシステム, 264-265

/tmp 用のファイルシステム (TMPFS), 259

## と

動的再構成, 81

### 取り出し

テープからファイルを取り出す (tar コマン  
ド)(手順), 459

リムーバブルメディア (手順), 37

取り付け, PCI アダプタカード (手順), 98

取り外し, PCI アダプタカード (手順), 97

## に

### 二次ディスク

システムへの接続 (手順)

SPARC, 210

x86, 220

説明, 174

入出力, 直接, 268

## ね

ネットワークベースのファイルシステム, 257



## は

- パーティション (スワップ)、定義, 332
- バイト (i ノードごとの数), 378
- 場所の調査, CacheFS ログファイル, 327
- バス指向のディスクコントローラ, 159, 160
- 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, 388
  - 準備, 402-403
  - 増分バックアップの記録, 437
  - タイプ, 388
  - デバイス名, 474-476
  - ファイルシステムの選択, 385
  - ファイルシステムのバックアップと復元
    - コマンド, 384
    - 定義, 384
    - 理由, 385
- バックアップスケジュール, 390
  - ガイドライン, 390
  - サーバー用, 397-400
  - 推奨事項, 391
  - ダンプレベルの使用, 393
  - 日単位増分、週単位累積バックアップ, 396-397
  - 日単位累積、週単位増分バックアップ, 395
  - 日単位累積、週単位累積バックアップ, 394例, 394, 400
- パラメータ (ファイルシステム), 375-378

## ひ

- 日単位個別バックアップ, 393
- 表示
  - CacheFS 統計情報, 329
  - USB デバイス情報 (手順), 132
  - システム構成情報, 73, 75
  - スワップ空間, 336-338
  - ディスクスライス情報 (概要), 189
  - デバイス情報, 76
  - バックされたファイル (手順), 317
  - バックされたファイル (例), 317
  - リムーバブルメディアのユーザー (手順), 37

## ふ

- ファイル
  - /etc/default/fs, 273
  - /etc/dfs/fstypes, 273
  - /proc ディレクトリ内, 260
  - 共有, 272
  - テープから取り出す (tar コマンド)(手順), 459
  - 複数のフロッピーディスクにアーカイブする (cpio)(手順), 471
  - メディアにコピーするためのコマンド (概要), 450
- ファイルシステム
  - /, 264
  - 4.3 Tahoe, 257
  - BSD Fat Fast, 257
  - DOS, 257
  - /export/home, 264
  - FDFS, 261
  - FIFOFS, 261
  - High Sierra, 257
  - ISO 9660, 257
  - MNTFS, 265
  - NAMEFS, 261
  - /opt, 264
  - PCFS, 257
  - /proc, 264
  - PROCFS、概要, 260
  - SPECFS, 261
  - SunOS のデフォルト, 264-265
  - SWAPFS, 261
  - TMPFS, 259-260
  - UFS, 257
  - UNIX, 257
  - /usr, 264
  - /var, 264
  - カスタムパラメータ, 375-378
  - 管理コマンドの説明, 262
  - 疑似、概要, 258-261
  - キャッシュ (概要), 303
  - 共有, 272
  - サイズのチェック, 347
  - 作成 (概要)
    - ループバック (LOFS), 280
  - 修正, 358
  - 修復, 357
  - 使用中のすべてのプロセスを終了(手順), 297
  - シリンダグループの構造, 372-375

- ファイルシステム (続き)
  - 全体のコピー (dd), 451
  - 大規模, 286
  - タイプ, 256-262
  - タイプの調査, 273
  - チェックと修復, 352
  - ディスクベースの, 256-257
  - ネットワークベース, 257
  - 破損, 344
  - バックアップするファイルシステム, 385
  - バックアップする理由, 385
  - 不整合の原因, 346
  - プロセス、概要, 260
  - マウントテーブル, 270
  - マニュアルページ, 263
  - 利用可能にする (概要), 283-288
- ファイルシステムテーブル, 仮想, 271
- ファイルシステムの修復, 357
- ファイルシステムのタイプ, 256-262
- ファイルシステムの破損, 344
- ファイルシステムの破損の原因, 344
- ファイルシステムの復元
  - 準備 (概要), 422-423
  - 使用するテープの決定 (手順), 424
  - 全体 (例), 429, 431
  - テープドライブのタイプ, 423
  - ルートと /usr (SPARC) (例), 434
  - ルートと /usr (x86) (例), 434
- ファイルシステムの不整合, 346
- ファイルシステム名, 423
- ファイルシステム用のカスタムパラメータ, 375-378
- ファイルの復元
  - 対話式による復元の例, 426
  - 非対話式による復元の例, 428
- ブートブロック, 372
- ブートブロックのインストール (手順), SPARC, 216
- 不整合なファイルシステムの修正, 358
- 不正な「.」と「..」エントリ, 351
- 不正な i ノード番号, 351
- 不正なスーパーブロック, 359
- 不正なスーパーブロックの復元, 359
- 不正なブロック番号, 350
- 物理デバイス名, 定義, 156
- フラグメントサイズ, 376-377
- プロセスファイルシステム (PROCFS), 260
- ブロック
  - 空き, 374-375
  - 間接, 350
  - 重複, 349
  - 通常データ, 352
  - ディレクトリデータ, 350
  - データ, 374
  - ブート, 372
  - 不正, 350
  - 論理サイズ, 375-376
- ブロック型, 特殊ファイル i ノード, 348
- ブロックディスクデバイスインタフェース
  - いつ使用するか, 158
  - 定義, 157
- フロッピーディスク
  - ファイルを複数のフロッピーディスクにアーカイブする (cpio)(手順), 471
  - ボリューム管理による読み込み (手順), 49
  - 読み込み
    - ボリューム管理による読み込み, 50
- へ
- 変更, 主 USB オーディオデバイス (手順), 147
- ほ
- ホットプラグ
  - cfgadm コマンドによる SCSI コントローラの切り離し (手順), 89
  - PCI アダプタカードの取り付け (手順), 98
  - PCI アダプタカードの取り外し (手順), 97
  - PCI デバイス (概要), 96
  - 概要, 81
- ボリューム管理
  - 再起動 (手順), 34
  - 終了 (手順), 33
  - 手動マウントと自動マウントの比較, 26
  - フロッピーディスク
    - 読み込み, 50
  - フロッピーディスクの読み込み (手順), 49
  - 利点, 26
  - リムーバブルメディア
    - アクセス, 27

## ま

### マウント

- /etc/vfstab を使用したファイルシステムのマウント, 291
- NFS ファイルシステム, 290
- UFS ファイルシステム, 290
- UFS ファイルシステム (手順)
  - 大規模ファイルを持たない, 293
- vfstab ファイル内のすべてのファイル, 290
- vold を使用して USB 大容量ストレージデバイスをマウントする (手順), 136
- 他のシステム上の PCMCIA メモリーカード (例), 43
- ファイルシステムの自動マウント, 272-273
- リムーバブルメディア
  - 自動マウントとの比較, 26
- リモートのリムーバブルメディアの手動によるマウント (例), 42
- リモートのリムーバブルメディアを手動で (例), 43
- ループバックファイルシステム (LOFS), 290
- マウント解除, vold を使用して USB 大容量ストレージデバイスをマウント解除する (手順), 136
- マウントテーブル, 270
- マウントポイント定義, 269
- マニュアルページ、ファイルシステム用の, 263

## み

- 未割り当てディレクトリブロック, 351
- 未割り当ての i ノード, 348

## め

- メディアの終わりの検出
  - cpio コマンド, 454
  - ufsdump コマンド, 436
- メモリー記憶域 (仮想)、定義, 332

## よ

### 読み込み

- フロッピーディスク
  - ボリューム管理による読み込み, 50
- ボリューム管理によるフロッピーディスクの読み込み (手順), 49

## り

- リセット, USB デバイス (手順), 152
- リムーバブルメディア
  - アクセスしているプロセスの強制終了 (手順), 37
  - アクセス (手順), 34
  - アクセス (例), 34
  - 他のシステム上のメディアへのアクセス (例), 42, 43
  - 取り出し (手順), 37
  - 名前, 30
  - マウント
    - 手動と自動の比較, 26
    - メディアが使用中かどうかの調査 (手順), 37
    - リムーバブルメディアのマウント (例), 42
    - リモートのメディアのマウント (例), 43
- 領域最適化タイプ, 377-378

## る

- ルート (/) ファイルシステム, 264
- ループバックファイルシステム (LOFS)
  - 作成 (概要), 280
  - マウント, 290

## れ

- レベル 0 バックアップ, 393

## ろ

- ログ (ダンプの記録), 436-437
- 論理デバイス名
  - 定義, 156
  - ディスク, 157
  - テープ, 160

論理デバイス名 (続き)

リムーバブルメディア, 161

論理ブロックサイズ, 375-376

わ

割り当て済みの i ノード, 348