



Solaris のシステム管理 (第 1 卷)

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

Part No: 805-5854-10
1998年 11 月

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。日本サン・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。

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

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリコービイ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イスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, SunSoft, SunDocs, SunExpress, OpenWindows, Solstice, Solstice AdminSuite, Solstice AutoClient, Solstice DiskSuite, Solstice Enterprise Agent, Admintool, CacheFS, Solaris Solve, Java, JavaStation, JumpStart, DeskSet は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

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

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

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

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

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1998 All Rights Reserved.)

ATOK は、株式会社ジャストシステムの登録商標です。

ATOK7 は株式会社ジャストシステムの著作物であり、ATOK7 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

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

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。(Copyright (c) 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ないものとします。

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、日本サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *System Administration Guide, Volume 1*

Part No: 805-3727-10

Revision A

© 1998 by Sun Microsystems, Inc.



目次

| | |
|-------------------------|------|
| はじめに | xxix |
| パートI ユーザーアカウントとグループの管理 | |
| 1. ユーザーアカウントとグループの管理の概要 | 3 |
| ユーザーアカウントとグループとは | 3 |
| ユーザーアカウント管理のガイドライン | 4 |
| ネームサービス | 4 |
| ユーザー (ログイン) 名 | 5 |
| ユーザー ID 番号 | 5 |
| パスワード | 9 |
| パスワード有効期限 | 10 |
| ホームディレクトリ | 10 |
| ユーザーの作業環境 | 11 |
| グループを管理するガイドライン | 11 |
| ユーザーアカウントとグループを管理するツール | 12 |
| Admintool の機能 | 16 |
| ユーザーアカウントの変更 | 16 |
| ユーザーアカウントの削除 | 17 |
| カスタマイズしたユーザー初期設定ファイルの追加 | 17 |
| パスワードの管理 | 17 |

| | |
|-----------------------------------|-----------|
| ユーザーアカウントを無効にする | 17 |
| ユーザーアカウントとグループ情報の格納場所 | 18 |
| パスワードファイルのフィールド | 18 |
| shadow ファイルのフィールド | 19 |
| グループファイルのフィールド | 20 |
| ユーザーの作業環境のカスタマイズ | 22 |
| サイト初期設定ファイルの使用方法 | 23 |
| ローカルシステムへの参照を避ける | 24 |
| シェル機能 | 24 |
| シェル環境 | 25 |
| PATH 変数 | 29 |
| ロケール変数 | 30 |
| デフォルトのファイルアクセス権 (umask) | 31 |
| ユーザー初期設定ファイルとサイト初期設定ファイルの例 | 32 |
| サイト初期設定ファイルの例 | 34 |
| 2. ユーザーアカウントとグループの設定と管理の手順 | 35 |
| スーパーユーザー (root) になる | 36 |
| ▼ スーパーユーザー (root) になる方法 | 36 |
| ユーザーアカウントの設定 | 37 |
| ユーザー情報データシート | 37 |
| ▼ ユーザー初期設定ファイルをカスタマイズする方法 | 39 |
| ▼ Admintool を起動する方法 | 40 |
| ▼ グループを追加する方法 | 41 |
| ▼ 新しいユーザーアカウントを追加する方法 | 42 |
| ▼ ユーザーのホームディレクトリを共有する方法 | 44 |
| ▼ ユーザーのホームディレクトリをマウントする方法 | 46 |
| ユーザーアカウントの管理 | 48 |
| ▼ グループを変更する方法 | 49 |

- ▼ グループを削除する方法 50
- ▼ ユーザーアカウントを変更する方法 50
- ▼ ユーザーアカウントを無効にする方法 52
- ▼ ユーザーのパスワードを変更する方法 54
- ▼ ユーザーアカウントのパスワード有効期限を変更する方法 55
- ▼ ユーザーアカウントを削除する方法 56

Solaris ユーザー登録 57

SolarisSM SolveSM へのアクセス 58

Solaris ユーザー登録の問題の解決 59

- ▼ Solaris ユーザー登録をやり直す方法 59
- ▼ ユーザー登録を無効にする方法 60

パートII サーバーとクライアントサポートの管理

3. サーバーとクライアントサポートの管理の概要 65
 - サーバーとクライアントの作業についての参照先 66
 - サーバーとクライアントとは 66
 - サポートとは 66
 - システムタイプの概要 67
 - サーバー 68
 - スタンドアロンシステム 69
 - ディスクレスクライアント 69
 - JavaStation クライアント 69
 - AutoClient システム 70
 - システムタイプ選択のガイドライン 70
 - サーバーとクライアントサポートの管理ツール 71
 - ホストマネージャでできること 72
 - サポートの追加と管理 72
 - システムタイプの更新 73
 - システムタイプの変換 74

- OS サービスの追加 74
 - リモートインストールサービスの設定 76
 - タスクを待ち行列に入れる 76
 - root パスワードの設定 76
 - スクリプトの有効化 77
 - マルチホームホストの追加 77
- ホストマネージャの制限事項 77
- ホストマネージャをスーパーユーザーとして実行する 78
- 4. サーバーとクライアントサポートの管理の手順 79
 - サーバーとクライアントサポートの追加 80
 - ▼ Solstice ホストマネージャを起動する方法 81
 - ▼ システムタイプを更新する方法 83
 - ▼ サポート追加時のデフォルトを設定する方法 84
 - ▼ スタンドアロンシステムまたは OS サーバーのサポートを追加する方法 84
 - ▼ システムを OS サーバーに変換する方法 87
 - ▼ SunOS 4.0 およびその互換バージョンの OS サービスを OS サーバーに追加する方法 89
 - ▼ Solaris 7 OS サービスを OS サーバーに追加する方法 94
 - ▼ AutoClient システムのサポートを追加する方法 96
 - ▼ ディスクレスクライアントサポートを追加する方法 99
 - サーバーとクライアントサポートの管理 102
 - ▼ システムサポートを変更する方法 103
 - ▼ システムサポートを削除する方法 104
 - ホストマネージャのコマンド行インタフェースを使用して設定作業を自動化する 104
 - パートIII システムのシャットダウンとブート
- 5. システムのシャットダウンとブートの概要 109
 - システムのシャットダウンとブートに関する新機能 109
 - レベル S (シングルユーザーモード) でのシステムの起動 110

| | | |
|-----------|--|------------|
| | 32 ビットまたは 64 ビット Solaris 7 オペレーティング環境のシステムのブート | 110 |
| | シャットダウンとブートについての参照先 | 110 |
| | 用語 | 110 |
| | システムのシャットダウンに関するガイドライン | 111 |
| | システムのブートに関するガイドライン | 112 |
| | 再構成用ブートの実行 | 112 |
| | システムをシャットダウンする場合 | 113 |
| | システムをブートする場合 | 114 |
| 6. | 実行レベルとブートファイルの手順 | 117 |
| | 実行レベル | 117 |
| | ▼ システムの実行レベルを確認する方法 | 119 |
| | /etc/inittab ファイル | 119 |
| | 例 — デフォルトの inittab ファイル | 120 |
| | システムが実行レベル 3 になると実行される処理 | 121 |
| | 実行制御スクリプト | 123 |
| | 実行制御スクリプトを使用してサービスを起動または停止する | 124 |
| | ▼ 実行制御スクリプトを使用してサービスを起動または停止する方法 | 124 |
| | 実行制御スクリプトの追加 | 125 |
| | ▼ 実行制御スクリプトを追加する方法 | 125 |
| | 実行制御スクリプトを無効にする | 126 |
| | ▼ 実行制御スクリプトを無効にする方法 | 126 |
| | 実行制御スクリプトのまとめ | 127 |
| 7. | システムのシャットダウンの手順 | 131 |
| | システムをシャットダウンする場合 | 131 |
| | システムをシャットダウンするには | 132 |
| | デバイスの電源を落とす場合 | 133 |
| | システムのダウンについてユーザーに通知する | 133 |

- ▼ システムにログインしているユーザーを知る方法 134
- ▼ サーバーをシャットダウンする方法 134
- ▼ スタンドアロンシステムをシャットダウンする方法 138
- ▼ すべてのデバイスの電源を落とす方法 140
- 8. **SPARC: システムのブートの手順 143**
 - SPARC: ブート PROM の使用方法 144
 - ▼ SPARC: ok プロンプトに切り換える方法 144
 - ▼ SPARC: システムの PROM リリースを確認する方法 144
 - ▼ SPARC: デフォルトのブートデバイスを変更する方法 145
 - ▼ SPARC: システムをリセットする方法 146
 - SPARC システムのブート 146
 - ▼ SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法 148
 - ▼ SPARC: システムを実行レベル S (シングルユーザー状態) にする方法 149
 - ▼ SPARC: システムを対話式でブートする方法 150
 - ▼ SPARC: システムを復元するためにブートする方法 152
 - ▼ SPARC: 復元のためにシステムを停止する方法 154
 - SPARC: クラッシュダンプを強制してシステムをリポートする 155
 - ▼ SPARC: クラッシュダンプを強制して、システムをリポートする方法 155
 - ▼ SPARC: カーネルデバッグ (kadb) を使ってシステムをブートする方法 156
- 9. **x86: システムのブートの手順 159**
 - x86 システムのブート 159
 - ▼ x86: システムを実行レベル 3 (マルチユーザー状態) にする方法 161
 - ▼ x86: システムを実行レベル S (シングルユーザー状態) にする方法 162
 - ▼ x86: システムを対話式でブートする方法 163
 - ▼ x86: システムを復元するためにブートする方法 165
 - ▼ x86: 復元を目的としてシステムを停止する方法 169
 - x86: クラッシュダンプを強制してシステムをリポートする 169
 - ▼ x86: クラッシュダンプを強制して、システムをリポートする方法 170

- 10. ブートプロセスの参照情報 171
 - SPARC: ブート PROM 171
 - SPARC: ブートプロセス 172
 - SPARC: ブートプロセスの詳細 172
 - x86: PC BIOS 173
 - x86: ブートサブシステム 174
 - x86: Solaris のブート 175
 - x86: デバイス識別段階で表示されるメニュー 176
 - x86: ブート段階で表示されるメニュー 178
 - x86: ブートプロセス 179
 - x86: ブートプロセスの詳細 180
- パートIV 取り外し可能な媒体の管理
- 11. CD とフロッピーディスクの使用法の概要 185
 - 取り外し可能な媒体の管理についての参照先 185
 - 機能と利点 186
 - 自動マウントと手作業によるマウントの比較 186
 - フロッピーディスクと CD で行える操作 187
- 12. コマンド行での CD の使用方法 189
 - CD の使用方法 189
 - CD 名の使用方法 190
 - ▼ CD をロードする方法 191
 - ▼ CD の内容を調べる方法 191
 - ▼ CD から情報をコピーする方法 191
 - ▼ CD が使用中かどうかを調べる方法 193
 - ▼ CD を取り出す方法 193
 - ▼ 他のシステム上の CD にアクセスする方法 194
 - ▼ ローカル CD を他のシステムで使用可能にする方法 196
 - ▼ 音楽 CD を演奏するようにシステムを設定する方法 199

- ▼ 新しい CD-ROM ドライブ用にシステムを準備する方法 200
- ボリューム管理の設定 201
- ▼ ボリューム管理を終了させる方法 201
- ▼ ボリューム管理を再起動する方法 201
- 13. コマンド行でのフロッピーディスクのフォーマットと使用方法 203
 - フロッピーディスクのフォーマット 203
 - フロッピーディスクの名前について 204
 - ハードウェアの考慮点 205
 - ▼ UFS フロッピーディスクをフォーマットする方法 208
 - ▼ UFS ファイルシステムをフロッピーディスクに作成する方法 211
 - ▼ DOS フロッピーディスクをフォーマットする方法 213
 - フロッピーディスクの使用方法 216
 - ▼ フロッピーディスクをロードする方法 216
 - ▼ フロッピーディスクの内容を調べる方法 218
 - ▼ フロッピーディスクから情報をコピーまたは移動する方法 219
 - ▼ フロッピーディスクへ情報をコピーまたは移動する方法 220
 - ▼ フロッピーディスクが使用中かどうかを調べる方法 221
 - ▼ フロッピーディスクを取り出す方法 222
 - ▼ 他のシステム上のフロッピーディスクにアクセスする方法 223
 - ▼ ローカルフロッピーディスクを他のシステムで使用可能にする方法 225
- 14. コマンド行での PCMCIA メモリーカードの使用方法 229
 - PCMCIA メモリーカードのフォーマット 230
 - PCMCIA メモリーカードの名前について 230
 - ハードウェアの考慮点 231
 - ▼ UFS PCMCIA メモリーカードをフォーマットする方法 232
 - ▼ UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法 234
 - ▼ DOS PCMCIA メモリーカードをフォーマットする方法 237
 - PCMCIA メモリーカードの使用方法 239

- ▼ PCMCIA メモリーカードをロードする方法 240
 - ▼ PCMCIA メモリーカードの内容を調べる方法 242
 - ▼ PCMCIA メモリーカードから情報をコピーまたは移動する方法 243
 - ▼ PCMCIA メモリーカードへ情報をコピーまたは移動する方法 244
 - ▼ PCMCIA メモリーカードが使用中かどうかを調べる方法 245
 - ▼ PCMCIA メモリーカードを取り出す方法 246
 - ▼ 他のシステム上の PCMCIA メモリーカードにアクセスする方法 247
 - ▼ ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする方法 249
- 15. ボリューム管理の動作の概要 253**
- ボリューム管理が取り外し可能な媒体すべてをマウントする 253
 - フロッピーディスクへのアクセス 254
 - CD へのアクセス 255
 - アクセスが容易なマウントポイント 256
 - 最終的に作成される 2 セットのシンボリックリンク 258
 - ファイルシステムアクセス用シンボリックリンク 258
 - raw デバイスアクセス用シンボリックリンク 259
 - UFS フォーマットによる制限 260
 - 混合 (ハイブリッド) フォーマットについて 260
- パート V ソフトウェアの管理
- 16. ソフトウェア管理の概要 265**
- ソフトウェア管理についての参照先 265
 - ソフトウェアパッケージ 266
 - ソフトウェア管理ツール 266
 - パッケージの追加または削除時の動作 267
 - パッケージを追加または削除する前に知っておくこと 268
 - クライアントソフトウェア管理に関するガイドライン 269
 - Sun パッケージをサーバーおよびクライアントへインストールする 269

| | | |
|------------|---|------------|
| | サードパーティパッケージをサーバーおよびクライアントへインストールする | 270 |
| | 異機種環境でのパッケージのインストール | 271 |
| | パッケージの削除に関するガイドライン | 272 |
| | パッケージ追加時のユーザーの対話操作を省略する | 272 |
| | 管理ファイルの使用 | 272 |
| | 応答ファイルの使用 | 274 |
| 17. | ソフトウェア管理の手順 | 275 |
| | ソフトウェアパッケージを処理するコマンド | 276 |
| | パッケージの追加と削除における既知の問題 | 276 |
| | パッケージの追加 | 277 |
| | ▼ スタンドアロンシステムにパッケージを追加する方法 | 277 |
| | スプールディレクトリの使用方法 | 280 |
| | ▼ スプールディレクトリにパッケージを追加する方法 | 280 |
| | 同機種クライアント/サーバー環境でのパッケージの追加 | 282 |
| | クライアントへの Sun パッケージの追加 | 283 |
| | クライアントへのサードパーティパッケージの追加 | 284 |
| | 異機種クライアント/サーバー環境でのパッケージの追加 | 284 |
| | ▼ パッケージファイルのインストール場所を決定する方法 | 285 |
| | ▼ ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法 | 286 |
| | ▼ パッケージをサーバーに追加する方法 | 289 |
| | パッケージのインストールの検査 | 293 |
| | ▼ インストールされたすべてのパッケージに関する情報を表示する方法 | 293 |
| | ▼ インストールされたパッケージが完全であるかどうかをチェックする方法 | 294 |
| | ▼ パッケージに関する詳細を表示する方法 | 295 |
| | サーバーとスタンドアロンシステムからのパッケージの削除 | 296 |
| | ▼ パッケージを削除する方法 | 296 |
| | ▼ スプールされたパッケージを削除する方法 | 297 |

- ▼ ディスクレスクライアントまたは AutoClient システムのパッケージを削除する方法 297
- Admintool を使用してソフトウェアパッケージを追加または削除する 299
- ▼ Admintool を起動する方法 299
- ▼ Admintool でパッケージを削除する方法 301
- 18. パッチ管理の概要 303
 - パッチについて 303
 - パッチ管理用ツール 304
 - パッチの配布 305
 - Sun パッチにアクセスするために必要な条件 305
 - World-Wide Web を介したパッチアクセス 306
 - ftp を介したパッチアクセス 306
 - パッチの番号付け 307
 - パッチの追加に伴う動作 307
 - パッチの削除に伴う動作 308
 - パートVI デバイスの管理
- 19. デバイス管理の概要と手順 311
 - デバイス管理における新機能 312
 - 動的再構成 312
 - x86: SCSI ディスクドライバ (sd) 312
 - デバイス管理作業の参照先 313
 - デバイスドライバについて 313
 - デバイスの自動設定 314
 - 機能と利点 315
 - 標準サポートされていないデバイスを使用する場合 315
 - システムへ周辺デバイスを追加する 315
 - ▼ 周辺デバイスを追加する方法 316
 - ▼ デバイスドライバを追加する方法 317

| | | |
|------------|-------------------------------|------------|
| | デバイス設定情報の表示 | 319 |
| | driver not attached メッセージ | 319 |
| | システムデバイスの識別 | 319 |
| | ▼ システム設定情報を表示する方法 | 320 |
| | ▼ デバイス情報を表示する方法 | 322 |
| 20. | デバイスへのアクセス | 325 |
| | デバイスへのアクセス | 325 |
| | デバイス情報が作成される方法 | 326 |
| | デバイス名の命名規則 | 326 |
| | 論理ディスクデバイス名 | 327 |
| | ディスクサブディレクトリの指定 | 327 |
| | スライスの指定 | 328 |
| | SPARC: 直接コントローラでアクセスされるディスク | 329 |
| | x86: 直接コントローラでアクセスされるディスク | 329 |
| | SPARC: バス指向コントローラでアクセスされるディスク | 330 |
| | x86: SCSI コントローラでアクセスされるディスク | 330 |
| | 論理テープデバイス名 | 331 |
| | 論理 CD-ROM デバイス名 | 332 |
| | パートVII ディスクの管理 | |
| 21. | ディスク管理の概要 | 335 |
| | ディスク管理作業の参照先 | 335 |
| | 概要 | 336 |
| | ディスク関連の用語 | 336 |
| | ディスクスライスについて | 336 |
| | SPARC: ディスクスライス | 337 |
| | x86: ディスクスライス | 339 |
| | raw データスライスの使用 | 340 |
| | 複数のディスク上のスライス配置 | 341 |

| | | |
|------------|---------------------------|------------|
| | 使用するスライスの決定 | 341 |
| | format ユーティリティ | 342 |
| | 定義 | 343 |
| | 機能と利点 | 343 |
| | format ユーティリティを使用する場合 | 344 |
| | format ユーティリティ使用上のガイドライン | 344 |
| | ディスクのフォーマット | 346 |
| | ディスクラベルについて | 347 |
| | パーティションテーブル | 347 |
| | ディスクをスライスに分割する | 351 |
| | free hog スライスの使用方法 | 351 |
| 22. | ディスク管理の手順 | 353 |
| | ディスクの管理 | 354 |
| | システム上のディスクの確認 | 355 |
| | ▼ システム上のディスクを確認する方法 | 355 |
| | ディスクのフォーマット | 357 |
| | ▼ ディスクがフォーマット済みかどうかを調べる方法 | 358 |
| | ▼ ディスクをフォーマットする方法 | 358 |
| | ディスクスライスの表示 | 360 |
| | ▼ ディスクスライス情報を表示する方法 | 360 |
| | ディスクラベルの作成と検査 | 363 |
| | ▼ ディスクラベルを作成する方法 | 363 |
| | ▼ ディスクラベルを検査する方法 | 365 |
| | 破損したディスクラベルの復元 | 366 |
| | ▼ 破損したディスクラベルを復元する方法 | 366 |
| | サードパーティのディスクの追加 | 369 |
| | format.dat のエントリの作成 | 370 |
| | ▼ format.dat のエントリを作成する方法 | 371 |

- SCSI ディスクドライブの自動構成 371
 - ▼ SCSI ドライブを自動構成する方法 372
 - 欠陥セクターの修復 374
 - ▼ 表面解析を使用して欠陥セクターを調べる方法 375
 - ▼ 欠陥セクターを修復する方法 376
 - ヒント 377
 - format セッションのデバッグ 378
 - 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 378
- 23. **SPARC: ディスク追加の手順 381**
 - SPARC: システムディスクと二次ディスクについて 381
 - SPARC: システムディスクまたは二次ディスクの追加 382
 - ▼ SPARC: システムディスクを接続してブートする方法 383
 - ▼ SPARC: 二次ディスクを接続してブートする方法 384
 - ▼ SPARC: ディスクスライスとディスクラベルを作成する方法 385
 - ▼ SPARC: ファイルシステムを作成する方法 390
 - ▼ 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 391
- 24. **x86: ディスク追加の手順 393**
 - x86: システムディスクと二次ディスクについて 393
 - x86: システムディスクまたは二次ディスクの追加 394
 - x86: fdisk パーティションの作成上のガイドライン 394
 - ▼ x86: システムディスクを接続してブートする方法 395
 - ▼ x86: 二次ディスクを接続してブートする方法 396
 - ▼ x86: Solaris fdisk パーティションを作成する方法 398
 - ▼ x86: ディスクスライスとディスクラベルを作成する方法 405
 - ▼ x86: ファイルシステムを作成する方法 408
 - ▼ 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 409
- 25. **format ユーティリティの参照情報 411**

| | |
|--------------------------------|-----|
| format ユーティリティを使用するための要件または制限 | 411 |
| format を使用する場合に情報を保存するための推奨事項 | 412 |
| format のメニューとコマンド記述について | 412 |
| partition メニュー | 414 |
| x86: fdisk メニュー | 415 |
| analyze メニュー | 416 |
| defect メニュー | 418 |
| format に使用されるファイル — format.dat | 419 |
| 構造 | 420 |
| 構文 | 420 |
| キーワード | 421 |
| スライステーブル | 423 |
| 位置 | 424 |
| format コマンドへの入力規則 | 425 |
| 数値 | 425 |
| ブロック番号 | 425 |
| コマンド名 | 426 |
| その他の名前 | 426 |
| ヘルプ | 427 |
| 関連するマニュアルページ | 427 |
| パートVIII ファイルシステムの管理 | |
| 26. ファイルシステムの概要 | 431 |
| ファイルシステムにおける新機能 | 431 |
| 概要 | 432 |
| ファイルシステムのタイプ | 433 |
| ディスクベースのファイルシステム | 433 |
| ネットワークベースのファイルシステム | 434 |
| 仮想ファイルシステム | 435 |

- ファイルシステム管理コマンド 437
 - 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 438
 - 汎用コマンドと専用コマンドのマニュアルページ 438
- デフォルトの Solaris ファイルシステム 439
- スワップ空間 440
- UFS ファイルシステム 441
 - UFS ファイルシステムの構成 441
 - UFS ログイン 442
 - UFS ファイルシステムの計画 443
- ファイルシステムのマウントとマウント解除 443
 - マウントされたファイルシステムテーブル 445
 - 仮想ファイルシステム 446
 - ネットワークファイルシステム (NFS) 447
 - 自動マウント (AutoFS) 448
 - キャッシュファイルシステム (CacheFS) 449
 - マウント方法の決定 449
- ファイルシステムのタイプを調べる 450
 - ▼ ファイルシステムのタイプを調べる方法 451
- 27. ファイルシステムの作成手順 453
 - UFS ファイルシステムの作成 453
 - ファイルシステムパラメータ 454
 - ▼ UFS ファイルシステムを作成する方法 455
 - 一時ファイルシステム (TMPFS) の作成 457
 - ▼ 一時ファイルシステムを作成する方法 457
 - ループバックファイルシステム (LOFS) の作成 459
 - ▼ LOFS ファイルシステムを作成する方法 459
- 28. ファイルシステムのマウントとマウント解除の手順 461
 - ファイルシステムのマウント 462

| | |
|---|------------|
| 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 | 463 |
| 汎用マウントオプション | 463 |
| ▼ 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 | 465 |
| /etc/vfstab ファイルでファイルシステムをマウントする | 466 |
| /etc/vfstab ファイルのフィールドの説明 | 466 |
| ▼ /etc/vfstab ファイルにエントリを追加する方法 | 467 |
| ▼ /etc/vfstab ファイル内に列挙されている複数のファイルシステムをマウントする方法 | 469 |
| ▼ /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する方法 | 471 |
| mount コマンドでファイルシステムをマウントする | 472 |
| ▼ UFS ファイルシステムをマウントする方法 | 472 |
| ▼ 大規模ファイルを持たない UFS ファイルシステムを再マウントする方法 | 473 |
| ▼ NFS ファイルシステムをマウントする方法 | 474 |
| ▼ x86: System V (S5FS) ファイルシステムをマウントする方法 | 475 |
| ▼ ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法 | 476 |
| ファイルシステムのマウント解除 | 477 |
| 前提条件 | 478 |
| マウント解除したファイルシステムを検査する | 478 |
| ▼ 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 | 479 |
| ▼ ファイルシステムをマウント解除する方法 | 480 |
| ▼ /etc/vfstab ファイルに列挙されているファイルシステムをマウント解除する方法 | 481 |
| 29. キャッシュファイルシステムの手順 | 483 |
| CacheFS の機能 | 484 |
| キャッシュファイルシステムの設定 | 485 |
| ▼ キャッシュを作成する方法 | 486 |
| キャッシュにマウントするファイルシステムの指定 | 487 |

- ▼ mount を使用してキャッシュにマウントするファイルシステムを指定する方法 487
- ▼ /etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法 490
- ▼ AutoFS を使用してファイルシステムをキャッシュにマウントする方法 491
- キャッシュされたファイルシステムの管理 492
- キャッシュの管理 493
- ▼ キャッシュ内でファイルシステムを変更する方法 493
- ▼ キャッシュされたファイルシステムに関する情報を表示する方法 494
- ▼ 必要に応じて整合性チェックを指定する方法 495
- ▼ キャッシュされたファイルシステムを削除する方法 496
- ▼ キャッシュされたファイルシステムの完全性をチェックする方法 497
- cachefspack によるキャッシュファイルシステムの管理 498
- ▼ キャッシュにファイルをバックする方法 499
- パッキングリスト 500
- ▼ パッキングリストを作成する方法 500
- ▼ パッキングリストに指定してファイルをキャッシュにバックする方法 500
- ▼ 正規表現として扱うファイルをパッキングリストに指定する方法 501
- ▼ 共有ディレクトリのファイルをバックする方法 502
- ファイルのパッキング解除 503
- ▼ 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 503
- バックされたファイルの情報の表示 504
- ▼ バックされたファイルの情報を表示する方法 505
- cachefspack コマンドのヘルプの表示 506
- cachefspack エラー 507
- CacheFS の統計情報 512
- CacheFS 統計情報を設定して表示するための前提条件 513
- CacheFS 統計情報の設定 513

| | |
|---------------------------------------|------------|
| CacheFS ログイン | 514 |
| ▼ ログインプロセスを設定する方法 | 514 |
| ログファイルの場所を調べる方法 | 515 |
| ログインプロセスを停止する方法 | 515 |
| キャッシュサイズの表示 | 516 |
| ▼ 作業セット (キャッシュ) のサイズを表示する方法 | 516 |
| 統計情報の表示 | 517 |
| ▼ キャッシュ統計情報を表示する方法 | 518 |
| キャッシュの構造と動作 | 519 |
| キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック | 520 |
| 必要に応じて行う整合性チェック | 520 |
| 30. 追加スワップ空間の構成の手順 | 521 |
| スワップ空間について | 521 |
| スワップ空間と仮想メモリー | 522 |
| スワップ空間と TMPFS ファイルシステム | 523 |
| スワップ空間の追加が必要かどうかを調べる方法 | 524 |
| スワップ関連のエラーメッセージ | 524 |
| TMPFS 関連のエラーメッセージ | 524 |
| スワップ空間の割り当て方法 | 525 |
| /etc/vfstab ファイル | 525 |
| スワップ空間の計画 | 526 |
| スワップ資源の監視 | 527 |
| スワップ空間の追加 | 528 |
| スワップファイルの作成 | 528 |
| ▼ スワップファイルを作成して使用可能にする方法 | 529 |
| スワップファイルを削除する | 530 |
| ▼ 余分なスワップ空間を削除する方法 | 531 |

- 31. ファイルシステムの整合性チェック 533
 - ファイルシステムの状態はどのように記録されるか 534
 - fsck でチェックして修復される内容 536
 - 非整合状態が発生する原因 536
 - 整合性がチェックされる UFS 構成要素 537
 - 自動ブートチェック機能の変更 543
 - /etc/vfstab ファイル 544
 - ▼ ブート中に実行される自動チェック機能を変更する方法 545
 - UFS ファイルシステムを対話式でチェックして修復する 546
 - ▼ ファイルシステムのチェックが必要かどうかを調べる方法 546
 - ▼ ファイルシステムを対話式でチェックする方法 547
 - UFS ファイルシステムの修復 548
 - ▼ ファイルシステムを修復する方法 548
 - 不正なスーパーブロックの復元 549
 - ▼ 不正なスーパーブロックを復元する方法 549
 - fsck で修復できない UFS ファイルシステムを修正する方法 551
 - fsck コマンドの構文とオプション 552
 - 汎用 fsck コマンドの構文、オプション、引数 552
- 32. ファイルシステムの参照情報 557
 - 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 557
 - プラットフォームに依存するディレクトリ 564
 - UFS ファイルシステムのシリンダグループの構造 565
 - ブートブロック 566
 - スーパーブロック 566
 - i ノード 567
 - データブロック 568
 - 空きブロック 568
 - カスタムファイルシステムパラメータの決定 569

| | | |
|-----|-------------------------------|-----|
| | 論理ブロックサイズ | 570 |
| | フラグメントサイズ | 570 |
| | 最小空き容量 | 571 |
| | 回転の遅れ (ギャップ) | 571 |
| | 最適化のタイプ | 572 |
| | ファイルの数 | 573 |
| | 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 | 573 |
| | newfs コマンドの構文、オプション、引数 | 573 |
| | 汎用 mkfs コマンド | 576 |
| | パートIX データのバックアップと復元 | |
| 33. | ファイルシステムのバックアップと復元の概要 | 579 |
| | バックアップと復元についての参照先 | 579 |
| | ファイルシステムのバックアップと復元とは | 580 |
| | ファイルシステムをバックアップしなければならない理由 | 581 |
| | テープデバイスの選択 | 581 |
| | バックアップを作成するファイルシステムの計画 | 582 |
| | バックアップコマンドと復元コマンドの概要 | 585 |
| | バックアップタイプの選択 | 586 |
| | 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 | 587 |
| | バックアップスケジュールに影響する要素 | 587 |
| | バックアップ頻度 | 588 |
| | ダンプレベルを使用して増分バックアップを作成する | 588 |
| | バックアップスケジュールの例 | 590 |
| | 例 — 日単位累積、週単位累積バックアップ | 590 |
| | 例 — 日単位累積、週単位増分バックアップ | 591 |
| | 例 — 日単位増分、週単位累積バックアップ | 592 |
| | 例 — サーバーのバックアップスケジュール | 593 |
| | 他のバックアップスケジュールに関する推奨事項 | 597 |

- 34. ファイルとファイルシステムのバックアップの手順 599
 - バックアップを実行する準備 599
 - ▼ ファイルシステム名を検索する方法 600
 - ▼ 完全バックアップをとるために必要なテープの本数を決定する方法 600
 - バックアップの実行 601
 - ▼ テープにバックアップをとる方法 602
- 35. ファイルとファイルシステムの復元の手順 611
 - ファイルとファイルシステムを復元する準備 611
 - ディスクデバイス名の決定 612
 - 使用するテープドライブのタイプの決定 612
 - テープドライブ名の決定 612
 - ファイルシステム全体の復元 613
 - 個別のファイルとディレクトリの復元 613
 - ファイルシステムの復元 614
 - ▼ 使用するテープを決定する方法 614
 - ▼ 対話式でファイルを復元する方法 615
 - ▼ 対話式でない方法で特定のファイルを復元する方法 618
 - ▼ リモートドライブを使ってファイルを復元する方法 620
 - ▼ ファイルシステム全体を復元する方法 621
 - ▼ ルート (/) と /usr を復元する方法 624
- 36. ufsdump コマンドと ufsrestore コマンドの参照情報 629
 - ufsdump の機能 629
 - デバイス特性の判断 630
 - 媒体の終りの検出 630
 - データのコピー 630
 - /etc/dumpdates ファイルの役割 630
 - バックアップデバイス (*dump-file*) 引数 631
 - バックアップを作成するファイルを指定する 633

- 媒体の終りの検出 634
 - テープの性質を指定する 634
 - ufsdump の制限 634
 - ufsdump コマンドのオプションと引数 635
 - デフォルトのコマンドオプション 635
 - ufsdump コマンドのオプション 636
 - ufsdump とセキュリティに関する注意事項 638
 - ufsrestore コマンドのオプションと引数 639
 - コマンド構文 639
 - オプションと引数 639
 - 対話式の復元コマンド 642
- 37. **UFS** ファイルとファイルシステムのコピー手順 645
 - ファイルシステムのコピーコマンド 646
 - ファイルシステムをディスクにコピーする 648
 - ファイルシステムのリテラルコピーを作成する 648
 - ▼ ディスクをクローン化する方法 (dd) 649
 - 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 652
 - ▼ 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) 652
 - ファイルとファイルシステムをテープにコピーする 654
 - tar を使用してファイルをテープにコピーする 656
 - ▼ ファイルをテープにコピーする方法 (tar) 656
 - ▼ テープ上のファイルのリストを表示する方法 (tar) 657
 - ▼ テープからファイルを取り出す方法 (tar) 658
 - pax を使用してファイルをテープにコピーする 660
 - ▼ ファイルをテープにコピーする方法 (pax) 660
 - ▼ 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) 661
 - ▼ テープ上のファイルのリストを表示する方法 (cpio) 662
 - ▼ テープからすべてのファイルを取り出す方法 (cpio) 663

- ▼ 指定したファイルをテープから取り出す方法 (cpio) 664
- ▼ ファイルをリモートテープドライブにコピーする方法 (tar と dd) 665
- ▼ ファイルをリモートテープドライブから取り出す方法 667
- 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 668
 - 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 669
- ▼ 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) 669
- ▼ 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) 670
- ▼ ファイルをフロッピーディスクから取り出す方法 (tar) 671
- ▼ 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 672
- ファイルを別のヘッダー形式でコピーする 672
- ▼ SunOS の旧リリース用のアーカイブを作成する方法 673
 - bar コマンドで作成したファイルを取り出す 674
- ▼ bar ファイルをフロッピーディスクから取り出す方法 674
- 38. テープドライブの管理手順 675
 - 使用する媒体の選択 675
 - バックアップデバイス名 676
 - テープドライブのデフォルト密度を指定する 677
 - テープドライブに別の密度を指定する 678
 - テープドライブの状態を表示する 678
 - ▼ テープドライブの状態を表示する方法 678
 - 磁気テープカートリッジの取り扱い 680
 - ▼ 磁気テープカートリッジのたるみを直す方法 680
 - ▼ 磁気テープカートリッジを巻き戻す方法 680
 - ドライブの管理と媒体処理のガイドライン 680
- A. 64 ビット Solaris オペレーティング環境 683
 - 64 ビット Solaris オペレーティング環境の概要 683
 - 64 ビット Solaris アプリケーション環境を使用する必要があるか 684

| | |
|--|-----|
| 64 ビット Solaris オペレーティング環境を使用する必要があるか | 684 |
| どの SPARC システムが 64 ビットアプリケーション開発をサポートしているか | 684 |
| どのようにして 64 ビット Solaris オペレーティング環境をインストールするか | 685 |
| システムが 64 ビット Solaris オペレーティング環境を実行できるかどうかを調べる方法 | 686 |
| ▼ システムが 64 ビット Solaris の機能を有効にしているかどうかを調べる方法 | 686 |
| どのようにして 64 ビット Solaris オペレーティング環境でアプリケーションを実行するか | 688 |
| 32 ビットと 64 ビットの Solaris 7 カーネルを実行するシステムを同じ方法でブートするのか | 689 |
| 64 ビット Solaris ブート時の問題の解決 | 690 |
| 64 ビット Solaris パッケージの変更 | 691 |
| 32 ビットまたは 64 ビットのアプリケーションを自動的にマウントする | 691 |
| 索引 | 693 |

はじめに

本書『Solaris のシステム管理 (第 1 巻)』は、Solaris システム管理に関する情報を提供する、2 巻構成のマニュアルの第 1 巻です。本書は、SPARC™ および x86 の両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。また、Solstice™ AdminSuite™ ツールを使用してシステム管理作業を行う方法についても説明しています。

このマニュアルでは、システム管理者である読者が SunOS 5.7 オペレーティングシステムと Solstice AdminSuite をすでにインストールして、ネットワークソフトウェアの設定を終了していることを想定しています。SunOS 5.7 オペレーティングシステムは、多くのユーティリティと OpenWindows™ バージョン 3 を含む Solaris 7 製品の一部です。また、SunOS 5.7 は、AT&T System V リリース 4 オペレーティングシステムに準拠しています。

システム管理に関する Solaris 7 の新機能については、各章の新機能に関する節を参照してください。

対象読者

このマニュアルは、Solaris 7 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。また、コンピュータサイエンスの学位またはそれに相当する知識があれば役立ちます。

このマニュアルの構成

このマニュアルは、システム管理作業の内容ごとに分類されたパートに分かれています。各パートでは概要と作業手順を説明しています。

概要は各パートのはじめの章で紹介され、その後の章で各作業手順を説明しています。また、各手順の説明の後には、その作業が正しく実行されたか確認する方法と、その作業を行う具体的な例が含まれています。

AnswerBook2 を使用してこのマニュアルを参照する

このマニュアルを AnswerBook2™ ビューアで表示している場合、本文中の下線部分(クロスリファレンス)をクリックすると、AnswerBook2 コレクションの参照情報をすぐに表示できます。元の表示に戻るには、ブラウザの「Back」ボタンをクリックします。

マニュアルの注文方法

SunDocs™ プログラムでは、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) の 250 冊以上のマニュアルを扱っています。このプログラムを利用して、マニュアルのセットまたは個々のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、米国 SunExpress™, Inc. のインターネットホームページ <http://www.sun.com/sunexpress> にあるカタログセクションを参照してください。

SPARC 情報と x86 情報

このマニュアルでは、SPARC システムと x86 システムの両方についてシステム管理情報を記載しています。特に断りがなければ、このマニュアルの説明はすべて両シ

システムに適用されます。表 P-1 に SPARC と x86 のシステム管理作業の違いを要約します。

表 P-1 SPARC と x86 のシステム管理作業の違い

| 分類 | SPARC | x86 |
|----------------------|--|---|
| カーネルが読み込まれる前のシステムの動作 | <ul style="list-style-type: none"> ■ モニタープログラムを持つ PROM チップが診断を実行し、デバイス情報を表示します。 ■ デフォルトのブートパラメータを設定したり、システムに接続されているデバイスをテストしたりする場合にも使用します。 | <ul style="list-style-type: none"> ■ BIOS が診断を実行し、デバイス情報を表示します。 <p>Multiple Device Boot (MDB) と呼ばれるプログラムを持つ Solaris Device Configuration Assistant (構成用補助) フロッピーディスクを使用して、デフォルト以外のブートパーティション、ネットワーク、または CD-ROM からブートさせます。</p> |
| システムのブート | <ul style="list-style-type: none"> ■ PROM レベルのコマンドとオプションを使用して、システムをブートします。 | <ul style="list-style-type: none"> ■ MDB、一次、および二次ブートサブシステムレベルのコマンドとオプションを使用して、システムをブートします。 |
| ブートプログラム | <ul style="list-style-type: none"> ■ bootblk — ufsboot を読み込む一次ブートプログラム ■ ufsboot — カーネルを読み込む二次ブートプログラム | <ul style="list-style-type: none"> ■ mboot — マスターブートレコード。pboot を読み込みます。 <p>pboot — Solaris パーティションブートプログラム。bootblk を読み込みます。</p> <ul style="list-style-type: none"> ■ bootblk — 一次ブートプログラム。ufsboot を読み込みます。 <p>ufsboot — 二次ブートプログラム。カーネルを読み込みます。</p> |
| システムのシャットダウン | <ul style="list-style-type: none"> ■ shutdown コマンドと init コマンドを、他の操作なしで使用できます。 | <ul style="list-style-type: none"> ■ shutdown コマンドと init コマンドを使用できます。ただし、「type any key to continue」プロンプトで入力する必要があります。 |
| ディスクコントローラ | <ul style="list-style-type: none"> ■ SCSI | <ul style="list-style-type: none"> ■ SCSI と IDE |

表 P-1 SPARC と x86 のシステム管理作業の違い 続く

| 分類 | SPARC | x86 |
|------------------|--|---|
| ディスクスライスとパーティション | <ul style="list-style-type: none"> ■ 1台のディスクは最大 8 個のスライスを持つことができます。番号は 0 ~ 7 です。 ■ Solaris の fdisk パーティションは最大 10 個のスライスを持つことができます。番号は 0 ~ 9 ですが、0 ~ 7 だけにユーザーデータを格納できます。 | <ul style="list-style-type: none"> ■ 1台のディスクは最大 4 個の fdisk パーティションを持つことができます。 ■ Solaris の fdisk パーティションは最大 10 個のスライスを持つことができます。番号は 0 ~ 9 ですが、0 ~ 7 だけにユーザーデータを格納できます。 |
| フロッピーディスクドライブ | <ul style="list-style-type: none"> ■ デスクトップシステムには通常 1 台の 3.5 インチドライブがあります。 | <ul style="list-style-type: none"> ■ システムに 3.5 インチと 5.25 インチの 2 つのドライブがある場合があります。 |

表記上の規則

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

表 P-2 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、またはコード例を示します。 | .login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system% |
| AaBbCc123 | ユーザーが入力する文字を、画面上のコンピュータ出力とは区別して示します。 | system% su password: |
| <i>AaBbCc123</i> | 変数を示します。実際に使用する特定の名前または値で置き換えます。 | ファイルを削除するには、rm <i>filename</i> と入力します。 |
| 『 』 | 参照する書名を示します。 | 『コードマネージャ・ユーザーズガイド』を参照してください。 |

表 P-2 表記上の規則 続く

| 字体または記号 | 意味 | 例 |
|---------|--|--|
| 「」 | 参照する章、節、ボタンやメ ニュー名、または強調する単語を 示します。 | 第 5 章「衝突の回避」を参照して ください。 この操作ができるのは、「スー パーユーザー」だけです。 |
| \ | 枠で囲まれたコード例で、テキス トがページ行幅を越える場合、 バックスラッシュは継続を示しま す。 | <pre>sun% grep `^#define \ XV_VERSION_STRING`</pre> |

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

コード例は次のように表示されます。

■ C シェルプロンプト

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

■ Bourne シェルおよび Korn シェルのプロンプト

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

■ スーパーユーザーのプロンプト

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

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

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

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

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

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が適宜、併記されています。
- 「x86」という用語は、一般に Intel 8086 ファミリに属するマイクロプロセッサを意味します。これには、Pentium、Pentium Pro の各プロセッサ、および AMD と Cyrix が提供する互換マイクロプロセッサチップが含まれます。このマニュアルでは、このプラットフォームのアーキテクチャ全体を指すときに「x86」という用語を使用し、製品名では「Intel 版」という表記で統一しています。
- このマニュアル中の手順を実行したり、例(コマンド入力、コードなど)を使用する場合には、二重引用符(")、左一重引用符(‘)、右一重引用符(’)をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。
- /sbin、/usr/sbin、/usr/bin、/etc ディレクトリにあるコマンドについては、このマニュアルでは絶対パス名で表記していない場合があります。
- このマニュアル中の例は、SunOS 5.7 ソフトウェアが標準的にインストールされていることを前提としています。つまり、バイナリ互換パッケージがインストールされていることや /usr/ucb が検索パスに設定されていることは、前提としていません。



注意 - /usr/ucb を検索パスに設定する場合は、パスの一番最後に追加してください。ps コマンドや df コマンドなどは、SunOS 5.7 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。

パート I ユーザーアカウントとグループの管理

このパートでは、ユーザーとグループの管理方法について説明します。次の章で構成されています。

| | |
|-------|---|
| 第 1 章 | ネットワーク環境でユーザーアカウントとグループを設定する方法について概要を説明します。 |
|-------|---|

| | |
|-------|--|
| 第 2 章 | ユーザーマネージャとグループマネージャを使用して、ユーザーアカウントとグループを設定する手順について説明します。 |
|-------|--|



ユーザーアカウントとグループの管理の概要

この章では、ユーザーアカウントとグループを管理するためのガイドラインと計画を説明し、さらに、ネットワーク環境でユーザーアカウントとグループを設定する方法について概要を説明します。また、ユーザーアカウントとグループ情報を格納するファイル、およびユーザーの作業環境のカスタマイズについても説明します。

この章の内容は次のとおりです。

- 3ページの「ユーザーアカウントとグループとは」
- 4ページの「ユーザーアカウント管理のガイドライン」
- 11ページの「グループを管理するガイドライン」
- 12ページの「ユーザーアカウントとグループを管理するツール」
- 18ページの「ユーザーアカウントとグループ情報の格納場所」
- 22ページの「ユーザーの作業環境のカスタマイズ」

ユーザーアカウントとグループを管理する手順については、第 2 章を参照してください。

ユーザーアカウントとグループとは

基本的なシステム管理作業の 1 つに、サイトにおいて各ユーザーにユーザーアカウントを設定することがあります。通常のユーザーアカウントには、ユーザーがシステムにログインして、システムを (root のパスワードを知らなくても) 使用するのに

必要な情報が含まれます。ユーザーアカウント情報は、主に次の4つで構成されています。

- ユーザー名 – ユーザーがシステムにログインするのに使用する名前 (ログイン名とも呼ばれる)。
- パスワード – ユーザーがシステムにアクセスするために、ユーザー名とともに入力しなければならない文字の組み合わせ。
- ユーザーのホームディレクトリ – 通常はログイン時のユーザーの現在のディレクトリであるディレクトリ。通常ホームディレクトリにはほとんどのユーザーファイルが含まれます。
- ユーザー初期設定ファイル – ユーザーがシステムにログインするとき、ユーザーの作業環境を設定する方法を制御するシェルスクリプト。

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

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

ユーザーアカウント管理のガイドライン

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

ネームサービス

大規模なサイトでユーザーアカウントを管理する場合、NIS または NIS+ などのネームサービスを使用できます。ネームサービスにより、ユーザーアカウント情報を各システムの `/etc` 内のファイルに格納するのではなく、1ヶ所で一貫した管理を行えます。ユーザーアカウントにネームサービスを使用すれば、サイト全体

のユーザーアカウント情報をシステムごとに /etc 内のファイルにコピーしなくても、同じユーザーアカウントのままシステム間を移動できます。

ユーザー (ログイン) 名

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

- 複数のドメインにまたがることもあるユーザーの組織内で、固有であること。
- 2文字から8文字の英数字を使用する (最初の文字は英字でなければならず、少なくとも1文字は小文字でなければならない)。
- 下線や空白文字は入れない。

ユーザー名の標準的な作り方を決めておくと便利です。ユーザー名はユーザーが覚えやすいものにしてください。単純な規則の例としては、ユーザーのファーストネームの頭文字とラストネームの最初の7文字を使用します。たとえば、Ziggy Ignatz は zignatz になります。他のユーザー名と重複する場合は、ユーザーのファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の6文字を使用します。たとえば、Ziggy Top Ignatz は ztignatz になります。さらに重複する場合、固有の名前になるまで、ファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の5文字、および1、2、3などの数字を使用できます。

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

ユーザー ID 番号

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

UID 番号は、2147483647 以下の整数でなければなりません。また、通常のユーザーアカウントと特殊なシステムアカウントに必要です。表 1-1 にユーザーアカウントとシステムアカウントに予約されている UID 番号を示します。

表 1-1 予約済みの UID 番号

| ユーザー ID 番号 | ログインアカウント | 説明 |
|------------------|------------------------|--|
| 0 - 99 | root、daemon、bin、sys、など | システムアカウント |
| 100 - 2147483647 | 通常のユーザー | 汎用アカウント |
| 60001 | nobody | アクセス権のないユーザー |
| 60002 | noaccess | Solaris 2.0 およびその互換バージョンと SVR4 リリースとの互換性のために提供 |

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

ユーザー (ログイン) 名と同様に、固有の UID を割り当てる方法を決めてください。企業によっては、固有の従業員番号に、管理者が 1000 を加えて固有の UID 番号を作成している場合もあります。

セキュリティ上のリスクを最小限に抑えるために、削除したアカウントの UID を再利用することは避けてください。どうしても UID を再利用する必要がある場合、はじめから作りなおして、新しいユーザーが前のユーザーの属性に影響されないようにしてください。たとえば、前のユーザーがプリンタの拒否リストに含まれていたためプリンタにアクセスできなかった場合、その属性を新しいユーザーにも適用するとは限りません。必要な場合には、固有の UID が使い果たされたとき、1 つの NIS+ ドメイン内で重複する UID を使用することができます。

大きなユーザー ID とグループ ID の使用

以前の Solaris ソフトウェアリリースは、32 ビットのデータ型を使用してユーザー ID (UID) とグループ (GID) を格納していました。ただし、UID と GID に使用できる最大値は 60000 に制限されていました。Solaris 2.5.1 リリースおよびその互換バージョンからは、UID と GID の値の制限は符号付き整数の最大値 (つまり、2147483647) に引き上げられました。

60000 を超える UID と GID は機能的に完全でなく、多くの Solaris の機能と互換性がありません。したがって、60000 を超える UID と GID を使用することは避けてください。Solaris 製品と Solaris コマンドとの相互運用性についての詳細は、表 1-2 を参照してください。

表 1-2 60000 を超える UID と GID の相互運用性に関する問題

| 分類 | 製品またはコマンド | 問題または注意 |
|---------------|----------------------------------|---|
| NFS™ 互換性 | SunOS 4.0 NFS ソフトウェアおよびその互換バージョン | NFS サーバーとクライアントのコードは、大きな UID と GID を 16 ビットに切り捨てる。これによって、大きな UID と GID を使用している環境において SunOS 4.0 およびその互換バージョンのシステムを使用すると、セキュリティの問題が発生する可能性がある。SunOS 4.0 およびその互換バージョンのシステムにはパッチが必要である。 |
| ネームサービスの相互運用性 | NIS ネームサービスおよびファイルベースのネームサービス | 60000 を超える UID を持つユーザーは、Solaris 2.5 およびその互換バージョンが動作しているシステムでは、ログインしたり、su コマンドを使用できるが、そのユーザーの UID と GID は 60001 (nobody) に設定される。 |
| | NIS+ ネームサービス | 60000 を超える UID を持つユーザーは、Solaris 2.5 およびその互換バージョンと NIS+ ネームサービスが動作しているシステムではアクセスが拒否される。 |
| UID と GID の表示 | OpenWindows のファイルマネージャ | OpenWindows™ のファイルマネージャを拡張ファイルリスト表示オプションで使用する場合、大きな UID と GID は正しく表示されない。 |

表 1-3 大きな UID と GID の制限の要約

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

パスワード

ユーザー名は公表されますが、パスワードを知っているのは各ユーザーだけでなければなりません。各ユーザーアカウントには、6文字から8文字の英数字と特殊文字を組み合わせたパスワードを割り当ててください。ユーザーアカウントを作成するときにユーザーのパスワードを設定します。ユーザーは、システムに初めてログインするときにそのパスワードを変更できます。

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

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

次のようなパスワードを選択してください。

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

次のものは、パスワードに不適當です。

- 自分の名前そのもの、逆読み、飛ばし読みのもの
- 家族やペットの名前
- 免許証番号
- 電話番号
- 社会保健番号
- 従業員番号
- 趣味や興味に関連した名前
- 12月の "Santa" のように季節に関連した言葉
- 辞書にある単語

パスワード有効期限

NIS+ または /etc 内のファイルを使用してユーザーアカウント情報を格納する場合は、ユーザーのパスワードにパスワード有効期限を設定できます。パスワード有効期限の設定により、ユーザーに定期的なパスワード変更を強制したり、あるパスワードを保持するのに必要な最低日数以前にパスワードを変更するのを防止したりできます。不正ユーザーが、古くて使用されていないアカウントを使用して、発覚せずにシステムのアクセス権を得るような場合を防止するために、アカウントが無効になる日付けを設定することができます。

ホームディレクトリ

ホームディレクトリは、ユーザーが専有ファイルを格納するのに割り当てられるファイルシステムの一部です。ホームディレクトリに割り当てられる大きさは、ユーザーが作成するファイルの種類と作業によって異なります。一般的には、各ユーザーのホームディレクトリに少なくとも 15M バイトのディスク空間を割り当ててください。

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

ホームディレクトリが配置される位置に関係なく、ユーザーは通常 /home/username という名前のマウントポイントを通してホームディレクトリにアクセスします。Autofs を使用してホームディレクトリをマウントすると、どのシステムでも /home マウントポイントの下にディレクトリを作成することは許可されません。Autofs が使用されていると、システムはマウントされている /home を特別なものと認識します。ホームディレクトリの自動マウントについての詳細は、『NFS の管理』を参照してください。

ネットワーク上の任意の位置からホームディレクトリを使用するには、/export/home/username ではなく、常に \$HOME という環境変数の値により参照するようにしてください。前者はマシンに固有の指定です。さらに、ユーザーのホームディレクトリで作成されるシンボリックリンクはすべて相対パス (たとえば

.././././x/y/x)を使用する必要があります。こうすることによって、そのリンクはどのシステムにホームディレクトリがマウントされても有効になります。

ユーザーの作業環境

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

ユーザーの作業環境をうまく管理するには、カスタマイズしたユーザー初期設定ファイル (.login、.cshrc、.profile) をユーザーのホームディレクトリに提供します。ユーザー初期設定ファイルをユーザー用にカスタマイズする詳細については、22ページの「ユーザーの作業環境のカスタマイズ」を参照してください。ユーザー初期設定ファイルをカスタマイズしたあと、新しいユーザーアカウントを作成するときにそれらをユーザーのホームディレクトリに追加できます。

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

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

グループを管理するガイドライン

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

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

- 一次グループ - オペレーティングシステムが、ユーザーにより作成されたファイルに割り当てるグループです。各ユーザーは、1 つの一次グループに所属していなければなりません。
- 二次グループ - ユーザーが所属する 1 つまたは複数のグループです。ユーザーは、最高 16 の二次グループに所属できます。

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

`groups` コマンドを使って、ユーザーが所属しているグループを表示できます。ユーザーは一度に 1 つの一次グループにしか所属できません。ただし、自分がそのメンバーである他のグループに (`newgrp` コマンドで) 一時的に変更することはできます。

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

グループは、システムにとってローカルになるか、またはネームサービスを通して管理することができます。グループ管理を単純化するために、グループメンバーを集中管理できる NIS+ のようなネームサービスを使用してください。

ユーザーアカウントとグループを管理するツール

表 1-4 に、ユーザーとグループを管理する推奨ツールを示します。

表 1-4 ユーザーとグループを管理するための推奨ツール

| 管理するユーザーとグループの場所 | 推奨ツール | 必要なもの | 参照先 |
|---|--|--|----------------------------------|
| ネットワークに接続されたネームサービス (NIS、NIS+) 環境におけるリモートまたはローカル、またはその両方のシステム | Solstice ™ AdminSuite™ のユーザーマネージャとグループマネージャ (グラフィカルユーザーインターフェース) | CDE や OpenWindows などの X window が動作しているグラフィックモニター | 『Solstice AdminSuite 2.3 管理者ガイド』 |
| ローカルシステム | Admintool (グラフィカルユーザーインターフェース) | CDE や OpenWindows などの X window が動作しているグラフィックモニター | 第 2 章 |

Solaris の `useradd` コマンドと `groupadd` コマンドを使用しても、ローカルシステムのユーザーとグループを設定できます。ただし、これらのコマンドではネームサービスのマップやテーブルを変更できません。表 1-5 で、Solstice AdminSuite または Admintool を使用しない場合に、ユーザーアカウントとグループを管理する Solaris のコマンドについて説明します。

表 1-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法

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

表 1-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法 続く

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

表 1-5 Solaris のコマンドを使用してユーザーアカウントとグループを管理する方法 続く

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

Admintool の機能

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

ユーザーアカウントの変更

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

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

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

ユーザーアカウントの中で変更できる情報に、ユーザーのグループメンバシップがあります。Admintool の「変更 (Modify)」オプションで、ユーザーの二次グループを追加したり、削除したりできます。あるいは、「グループ」ウィンドウを使ってグループのメンバーリストを直接修正したりすることもできます。

ユーザーアカウントの次の部分も変更できます。

- コメント
- ログインシェル
- パスワード
- ホームディレクトリ
- AutoHome 設定
- 資格テーブルの設定
- メールサーバー

ユーザーアカウントの削除

Admintool でユーザーアカウントを削除すると、passwd、group、ファイル内のエントリが自動的に削除されます。さらに、ユーザーのホームディレクトリにあるファイルを削除し、ユーザーのメールボックスの内容を削除できます。

メールをユーザーのメールボックスに送るよう設定されている 1 つのメール別名だけが削除されます。そのユーザー名は、他のメール別名からは削除されません。メールボックスにメールを送るよう設定されている以外のメール別名からエントリを削除したいときは、手作業で削除しなければなりません。

カスタマイズしたユーザー初期設定ファイルの追加

ユーザーマネージャでユーザー初期設定ファイルをカスタマイズして作成することはできませんが、指定された「スケルトン」ディレクトリ内のユーザー初期設定ファイルでユーザーのホームディレクトリを生成することができます。

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

パスワードの管理

Admintool を使って、次のようにパスワードを管理できます。(1) ユーザーアカウントに通常のパスワードを指定する(2) ユーザーが最初のログイン時にパスワードを作成できるようにする(3) ユーザーアカウントを無効にするかロックする(4) 有効期限とパスワード有効期限情報を指定する

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

ユーザーアカウントを無効にする

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

最も簡単にユーザーアカウントを無効にする方法は、Admintool を使用してアカウントのパスワードをロックすることです。また、「有効期限 (Expiration Date)」

フィールドに有効期限を入力して、ユーザーアカウントの無効期間を設定することができます。

また、パスワード有効期限を設定するか、パスワードを変更しても、ユーザーアカウントを無効にできます。

ユーザーアカウントとグループ情報の格納場所

ユーザーアカウントとグループ情報は、サイトの方針に応じて、ネームサービスかローカルシステムの /etc 内のファイルのどちらかに格納できます。NIS+ ネームサービスでは、情報はテーブルに格納され、NIS ネームサービスではマップに格納されます。

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

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

グループ情報は group ファイルに格納されます。

パスワードファイルのフィールド

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

```
username:パスワード:uid:gid:comment:home-directory:login-shell
```

具体的な例は次のようになります。

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

表 1-6 に、passwd ファイルの各フィールドについて説明します。

表 1-6 passwd ファイルのフィールド

| フィールド名 | 説明 |
|-----------------------|--|
| <i>username</i> | ユーザー (またはログイン) 名。ユーザー名は固有で、1 から 8 文字の英字 (A-Z、a-z) と数字 (0-9) を使用する。最初の文字は英字で、少なくとも 1 文字は小文字を使用する。ユーザー名には下線や空白文字は使用できない。 |
| <i>password</i> | 暗号化されたパスワードの代わりに x (パスワードフィールドはもう使用されない)。暗号化されたパスワードは shadow ファイルに格納される。 |
| <i>uid</i> | ユーザーをシステムに識別させるユーザー識別番号 (UID)。一般ユーザーの UID は 100 から 60000 までの範囲とする。UID 番号はすべて固有でなければならない。 |
| <i>gid</i> | ユーザーの一次グループのグループ識別番号 (GID)。各 GID は 0 から 60002 までの範囲の整数でなければならない (60001 と 60002 はそれぞれ <i>nobody</i> と <i>noaccess</i> に割り当てられる)。 |
| <i>comment</i> | 通常はユーザーのフルネーム。(このフィールドはコメントとしての情報専用。) このフィールドは、もともとは、Bell 研究所の UNIX システムから GECOS (General Electric Computer Operating System) を実行するメインフレームにバッチジョブを依頼する場合、必要なログイン情報を保持するために使われていたので、GCOS フィールドと呼ばれることもある。 |
| <i>home-directory</i> | ユーザーのホームディレクトリのパス名。 |
| <i>login-shell</i> | ユーザーのデフォルトログインシェル。これは <code>/bin/sh</code> 、 <code>/bin/csh</code> 、 <code>/bin/ksh</code> のどれかになる。表 1-11 のシェル機能の説明を参照のこと。 |

shadow ファイルのフィールド

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

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

具体的な例は次のようになります。

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

表 1-7 に shadow ファイルの各フィールドについて説明します。

表 1-7 shadow ファイルのフィールド

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

グループファイルのフィールド

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

```
group-name:group-パスワード:gid:user-list
```

具体的な例は次のようになります。

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

表 1-8 に group ファイルの各フィールドについて説明します。

表 1-8 group ファイルの各フィールド

| フィールド名 | 説明 |
|--------------------|---|
| <i>group-name</i> | グループに付けられた名前。たとえば、大学の化学部のメンバーであれば chem などと指定する。グループ名に許される最大文字数は 9 文字。 |
| <i>group-パスワード</i> | 通常は空のままか、アスタリスクを指定する。グループパスワードフィールドは初期バージョンの UNIX のなごりで、グループにパスワードがある場合、newgrp コマンドはユーザーにグループパスワードを入力するよう求める。ただし、パスワードを設定するユーティリティはない。 |
| <i>gid</i> | グループ GID 番号。ローカルシステムで固有でなければならず、組織全体を通じても固有であることが望ましい。GID 番号は 0 から 60002 までの範囲の整数で指定する。ただし、100 未満の番号はシステムのデフォルトグループアカウント用に予約されている。したがって、ユーザー定義グループの範囲は 100 から 60000 (60001 と 60002 はそれぞれ nobody と noaccess に割り当てられている)。 |
| <i>user-list</i> | カンマで区切られたユーザー名のリスト。ユーザーの二次グループメンバシップを表す。各ユーザーは最高 16 までの二次グループに所属できる。 |

UNIX ユーザーグループ

すべての Solaris 7 システムに、デフォルトで次のグループがあります。

```

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

```

ユーザーの作業環境のカスタマイズ

ユーザーのホームディレクトリの設定には、ユーザーのログインシェルにユーザー初期設定ファイルを提供することも含まれます。ユーザー初期設定ファイルは、ユーザーがシステムにログインしたあとでユーザーのために作業環境を設定するシェルスクリプトです。基本的にシェルスクリプトでできる作業はどれもユーザー初期設定ファイルで実行できますが、その主な仕事は、ユーザーの検索パス、環境変数、ウィンドウ機能の環境などユーザーの作業環境を定義することです。表 1-9 に示すように、各ログインシェルには、1 つまたは複数の、固有のユーザー初期設定ファイルがあります。

表 1-9 Bourne、C、Korn シェルのユーザー初期設定ファイル

| シェル | 初期設定ファイル | 目的 |
|--------|------------------------------|--|
| Bourne | <code>\$HOME/.profile</code> | ログイン時のユーザー環境の定義 |
| C | <code>\$HOME/.cshrc</code> | ログインシェルのあとで起動されるすべての C シェルに対するユーザー環境の定義 |
| | <code>\$HOME/.login</code> | ログイン時のユーザー環境の定義 |
| Korn | <code>\$HOME/.profile</code> | ログイン時のユーザー環境の定義 |
| | <code>\$HOME/\$ENV</code> | Korn シェルの ENV 環境変数によって指定されるファイルでのログイン時のユーザー環境の定義 |

Solaris システムソフトウェアには、表 1-10 に示すように、各システムの `/etc/skel` ディレクトリに、各シェル用のデフォルトのユーザー初期設定ファイルが提供されています。

表 1-10 デフォルトのユーザー初期設定ファイル

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

これらのファイルを変更して、すべてのユーザーに共通な作業環境を提供する標準のファイルセットを作成したり、あるいは異なるタイプのユーザーに作業環境を提供したりすることができます。異なるタイプのユーザーにユーザー初期設定ファイルを作成する手順については、39ページの「ユーザー初期設定ファイルをカスタマイズする方法」を参照してください。

サイト初期設定ファイルの使用方法

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

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

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

C シェルのユーザー初期設定ファイルでサイト初期設定ファイルを参照するには、ユーザー初期設定ファイルの初めに次のような行を入れてください。

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

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

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

ローカルシステムへの参照を避ける

ユーザー初期設定ファイルでは、ローカルシステムへの固有の参照は追加しないでください。初期設定ファイルの設定は、ユーザーがどのシステムにログインしても有効になる必要があります。次に例を示します。

- ユーザーのホームディレクトリをネットワーク上の任意の位置で利用できるようにするには、常に環境変数の値 `$HOME` を使用してホームディレクトリを参照してください。たとえば、`/export/home/username/bin` ではなく、`$HOME/bin` を使用してください。`$HOME` は、ユーザーが別のシステムにログインする場合でも有効で、その場合ホームディレクトリは自動マウントされます。
- ローカルディスクのファイルにアクセスするには、`/net/machine-name/directory-name` などのグローバルパス名を使用してください。システムが `AutoFS` を実行していれば、`/net/machine-name` で参照されるディレクトリはすべてユーザーがログインする任意のシステムに自動的にマウントできます。

シェル機能

表 1-11 に、各シェルの基本的機能を示します。ユーザー初期設定ファイルを作成するのにどのシェルがどんな機能を提供するか参考にしてください。

表 1-11 Bourne、C、Korn シェルの基本機能

| 機能 | Bourne | C | Korn |
|---------------------|--------|---|------|
| UNIX で標準シェルとして知られる | ○ | × | × |
| Bourne シェルと互換性がある構文 | - | × | ○ |
| ジョブ制御 | ○ | ○ | ○ |
| 履歴リスト | × | ○ | ○ |

表 1-11 Bourne、C、Korn シェルの基本機能 続く

| 機能 | Bourne | C | Korn |
|------------------------|--------|---|------|
| コマンド行編集 | × | ○ | ○ |
| 別名 (エイリアス) | × | ○ | ○ |
| ログインディレクトリの 1 文字省略形 | × | ○ | ○ |
| ファイルの上書き保護 (noclobber) | × | ○ | ○ |
| CTRL-D 無視 (ignoreeof) | × | ○ | ○ |
| 拡張 cd | × | ○ | ○ |
| .profile とは別の初期設定ファイル | × | ○ | ○ |
| ログアウトファイル | ○ | ○ | × |

シェル環境

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

- 環境変数 – シェルによって生成されるすべてのプロセスにエクスポートされる変数。環境変数の設定値は env コマンドで表示できます。PATH などを含む環境変数の一部が、シェルそのものの動作に影響を与えます。
- シェル (ローカル) 変数 – 現在使用中のシェルだけに関する変数。C シェルの場合は、シェル変数は環境変数と特別に対応しています。これらのシェル変数は user、term、home、path です。シェル変数は、対応する環境変数の値によって初期設定されます。

C シェルでは、小文字を使って set コマンドでシェル変数を設定し、大文字を使って setenv コマンドで環境変数を設定します。シェル変数を設定すると、対応する

環境変数が設定され、その逆もあります。たとえば、path シェル変数を新しいパスで更新すると、シェルは PATH 環境変数も新しいパスで更新します。

Bourne、Korn 両シェルでは、一般的に、大文字を使って `setenv` コマンドでシェル変数と環境変数を設定します。また、`export` コマンドで環境変数の設定を終了しなければなりません。すべてのシェルで、シェル変数と環境変数は一般的に大文字の名前で参照します。

ユーザー初期設定ファイルで、ユーザーのシェル環境を、あらかじめ定義された変数の値を変更するか、変数を追加することによってカスタマイズできます。表 1-12 はユーザー初期設定ファイルで環境変数を設定する方法を示します。

表 1-12 ユーザー初期設定ファイルでの環境変数の設定方法

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

表 1-13 に、ユーザー初期設定ファイルでカスタマイズできる環境変数とシェル変数を説明します。各シェルで使用される変数の詳細については、`sh(1)`、`ksh(1)`、`csh(1)` の各マニュアルページを参照してください。

表 1-13 シェル変数と環境変数の説明

| 変数 | 説明 |
|-------------------------|--|
| ARCH | ユーザーのシステムアーキテクチャ (たとえば sun4、i386) を設定する。この変数は ARCH = 'uname -p' (Bourne または Korn シェル) または setenv ARCH 'uname -p' (C シェル) で設定する。この変数に影響されるシェルの動作はない。この変数は単にシェルスクリプト内での分岐に利用する。 |
| CALENDAR | Calendar 実行ファイルにパスを設定する。 |
| CDPATH (C シェルでは cdpath) | cd コマンドで使用する変数を設定する。cd コマンドの対象ディレクトリを相対パス名で指定すると、cd コマンドは対象ディレクトリをまず現在のディレクトリ (.) 内で検索する。見つからなかった場合は、CDPATH 変数のリストの順で検索され、見つかると、ディレクトリの変更が行われる。CDPATH で対象のディレクトリが見つからなかった場合は、現在の作業ディレクトリは変更されない。たとえば、CDPATH 変数を /home/jean に設定し、その下に bin と rje に 2 つのディレクトリがある場合に、/home/jean/bin ディレクトリの中で cd rje と入力すると、絶対パスを指定しなくても、ディレクトリを /home/jean/rje に変更することになる。 |
| DESKSET | DeskSet™ 実行ファイルへのパスを設定する。 |
| history | C シェルの履歴を設定する。 |
| HOME (C シェルでは home) | ユーザーのホームディレクトリへのパスを設定する。 |
| LANG | ロケールを設定する。 |
| LOGNAME | 現在ログインしているユーザーの名前を設定する。LOGNAME のデフォルト値は、passwd ファイルに指定されているユーザー名にログインプログラムによって自動的に設定される。したがって、この変数を参照すればユーザー名を入手できる (設定を変更してはならない)。 |
| LPDEST | ユーザーのデフォルトプリンタを設定する。 |
| MAIL | ユーザーのメールボックスへのパスを設定する。 |
| MANPATH | アクセスできるマニュアルページの階層を設定する。 |
| MANSECTS | アクセスできるマニュアルページの階層を設定する。 |
| OPENWINHOME | OpenWindows サブシステムへのパスを設定する。 |

表 1-13 シェル変数と環境変数の説明 続く

| 変数 | 説明 |
|-----------------------|---|
| PATH (C シェルでは path) | <p>ユーザーがコマンドを入力したときに実行するプログラムについて、シェルが検索するディレクトリを順番に指定する。ディレクトリが検索パス上にない場合は、ユーザーはコマンドの絶対パス名を入力しなければならない。</p> <p>デフォルトの PATH は、ログインプロセスで .profile (Bourne または Korn シェル) または .cshrc (C シェル) の指定どおりに自動的に定義され、設定される。</p> <p>検索パスの順序が重要となる。同じコマンドが異なる場所にそれぞれ存在するときは、最初に見つかったコマンドが使用される。たとえば、PATH が (Bourne、Korn シェル構文で) PATH=/bin:/usr/bin:/usr/sbin:\$HOME/binのように定義されていて、sample というファイルが /usr/bin と /home/jean/bin の両方にあるものとする。ユーザーが sample コマンドを、その絶対パスを指定しないで入力した場合は、/usr/bin で見つかったバージョンが使用される。</p> |
| prompt | C シェルのシェルプロンプトを設定する。 |
| PS1 | Bourne または Korn シェルのシェルプロンプトを設定する。 |
| SHELL (C シェルでは shell) | make、vi、その他のツールが使うデフォルトシェルを設定する。 |
| TERMINFO | <p>terminfo ファイルに追加した、サポートされていない端末のパス名を指定する。/etc/profile または /etc/.login で TERMINFO 変数を使う。</p> <p>TERMINFO 環境変数を設定すると、システムはまずユーザーが定義した TERMINFO パスを調べる。ユーザーが定義した TERMINFO ディレクトリ内に端末の定義が見つからなかった場合は、システムはデフォルトディレクトリ /usr/share/lib/terminfo で定義を探す。どちらにも見つからなかった場合は、端末は dumb として定義される。</p> |

表 1-13 シェル変数と環境変数の説明 続く

| 変数 | 説明 |
|---------------------|--|
| TERM (C シェルでは term) | 端末を設定する。この変数は /etc/profile または /etc/.login で再度設定する必要がある。ユーザーがエディタを起動すると、システムはこの環境変数の定義と同じ名前のファイルを探す。システムは、TERMINFO が参照するディレクトリ内を探して端末の特性を知る。 |
| TZ | 時間帯を設定する。これは、たとえば <code>ls -l</code> コマンドで日付を表示する場合に使われる。TZ をユーザーの環境に設定しないと、システムの設定が使用される。設定する場合、グリニッジ標準時が使用される。 |

PATH 変数

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

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

ガイドライン

次に、効率的な PATH 変数を設定するガイドラインをいくつか示します。

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

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

例 — ユーザーのデフォルトパスの設定

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

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

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

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

ロケール変数

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

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

表 1-14 は、LANG と LC 環境変数の値を示します。

表 1-14 LANG と LC 変数の値

| ロケール | 値 |
|------|--------|
| de | German |
| fr | French |

表 1-14 LANG と LC 変数の値 続く

| ロケール | 値 |
|------------|----------------------|
| iso_8859_1 | English および European |
| it | Italian |
| japanese | Japanese |
| korean | Korean |
| sv | Swedish |
| tchinese | Taiwanese |

例 — LANG 変数によるロケールの設定

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

```
setenv LANG DE
```

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

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

デフォルトのファイルアクセス権 (umask)

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

ユーザーマスクは 3 桁の 8 進値で設定します。最初の桁でそのユーザーのアクセス権を設定し、第 2 桁でグループのアクセス権を設定し、第 3 桁で「その他」(ワールドとも呼ばれます)のアクセス権を設定します。最初の桁がゼロの場合、この桁は表示されません。たとえば、umask を 022 に設定すると、22 が表示されます。

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

また、表 1-15 から `umask` 値を決めることもできます。この表は、`umask` の各 8 進値から作成される、ファイルとディレクトリのアクセス権を示します。

表 1-15 `umask` 値のアクセス権

| <code>umask</code> 8 進値 | ファイルアクセス権 | ディレクトリアクセス権 |
|-------------------------|-----------------------|-----------------------|
| 0 | <code>rw-</code> | <code>rwX</code> |
| 1 | <code>rw-</code> | <code>rw-</code> |
| 2 | <code>r--</code> | <code>r-X</code> |
| 3 | <code>r--</code> | <code>r--</code> |
| 4 | <code>-w-</code> | <code>-wX</code> |
| 5 | <code>-w-</code> | <code>-w-</code> |
| 6 | <code>--x</code> | <code>--x</code> |
| 7 | <code>---</code> (なし) | <code>---</code> (なし) |

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

```
umask 000
```

ユーザー初期設定ファイルとサイト初期設定ファイルの例

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

コード例 — .profile ファイルの例

コード例 1-1 .profile ファイルの例

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

コード例 — .cshrc ファイルの例

コード例 1-2 .cshrc ファイルの例

```
8set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)  
9setenv MAIL /var/mail/$LOGNAME  
10setenv NNTPSERVER server1  
11setenv PRINTER printer1  
12alias h history  
13umask 022
```

1. ユーザーのシェルの検索パスを設定する。
2. ユーザーのメールファイルへのパスを設定する。
3. ユーザーの Usenet ニュースサーバー用の環境変数を設定する。
4. マニュアルページへのユーザーの検索パスを設定する。
5. ユーザーのデフォルトプリンタを設定する。
6. ユーザーのデフォルトのファイル作成アクセス権を設定する。
7. 指定された環境変数をエクスポートする。
8. ユーザーのシェルの検索パスを設定する。
9. ユーザーのメールファイルへのパスを設定する。
10. ユーザーの Usenet ニュースサーバーを設定する。
11. ユーザーのデフォルトプリンタを設定する。
12. History コマンドの別名を作成する (h と入力するだけで history コマンドを実行できる)。
13. ユーザーのデフォルトのファイル作成アクセス権を設定する。

(続く)

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

サイト初期設定ファイルの例

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

表 1-16 サイト初期設定ファイルの例

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

set choice = $<

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

switch ($choice)

case "1":
setenv APPHOME /opt/app-v.1
breaksw

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

このサイト初期設定ファイルはユーザーの `.cshrc` ファイル (C シェルユーザーのみ使用可能) で、次のように参照できます。

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

この行では、サイト初期設定ファイルは `site.login` と指定され、`server2` という名前のサーバー上にあります。また、この行では自動マウントがユーザーのシステムで実行されていると仮定しています。

14. 34ページの「サイト初期設定ファイルの例」に示すサイト初期設定ファイルを実行する。

ユーザーアカウントとグループの設定と管理の手順

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

この章の内容は次のとおりです。

- 39ページの「ユーザー初期設定ファイルをカスタマイズする方法」
- 40ページの「Admintool を起動する方法」
- 41ページの「グループを追加する方法」
- 42ページの「新しいユーザーアカウントを追加する方法」
- 44ページの「ユーザーのホームディレクトリを共有する方法」
- 46ページの「ユーザーのホームディレクトリをマウントする方法」
- 49ページの「グループを変更する方法」
- 50ページの「グループを削除する方法」
- 50ページの「ユーザーアカウントを変更する方法」
- 52ページの「ユーザーアカウントを無効にする方法」
- 54ページの「ユーザーのパスワードを変更する方法」
- 55ページの「ユーザーアカウントのパスワード有効期限を変更する方法」
- 56ページの「ユーザーアカウントを削除する方法」
- 59ページの「Solaris ユーザー登録をやり直す方法」
- 60ページの「ユーザー登録を無効にする方法」

ユーザーアカウントとグループ管理の概要については、第 1 章を参照してください。

スーパーユーザー (root) になる

ユーザーを追加するなどのほとんどの管理作業では、まず、root (UID=0) でログインする必要があります。root アカウントは、システムを変更したり、緊急時にユーザーファイルの保護を無効にしたりできるので、スーパーユーザーアカウントとしても知られています。

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

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

▼ スーパーユーザー (root) になる方法

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

■ su コマンドを使用して、スーパーユーザーに変更する方法

```
% su
Password: root_password
#
```

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

■ システムコンソールでスーパーユーザーとしてログインする方法

```
hostname console: root
Password: root_password #
```

この方法は、デフォルトでは使用できません。システムコンソールでスーパーユーザーとしてログインするには、/etc/default/login ファイルを変更しなければなりません。このファイルの変更方法については、『Solaris のシステム管理 (第 2 巻)』の「システムの保護」を参照してください。

ユーザーアカウントの設定

表 2-1 作業マップ: ユーザーアカウントの設定

| 作業 | 説明 | 手順の説明 |
|------------------------|---|---------------------------------|
| 1. ユーザー初期設定ファイルのカスタマイズ | (省略可能) ユーザー初期設定ファイル (.cshrc、.profile、.login) を設定する。これによって新規ユーザーに一貫した環境を提供できる。 | 39ページの「ユーザー初期設定ファイルのカスタマイズする方法」 |
| 2. グループの追加 | (省略可能) ユーザーの管理を容易にするため、グループマネージャの「編集(Edit)」メニューから「追加(Add)」を選択してグループを追加する。通常は 1 回だけ行う作業。 | 41ページの「グループを追加する方法」 |
| 3. ユーザーアカウントの追加 | 新しいユーザーアカウントを追加する Admintool の「ユーザー」メインウィンドウを使用して、ユーザーアカウントを追加する。 | 42ページの「新しいユーザーアカウントを追加する方法」 |
| 4. ユーザーのホームディレクトリの共有 | ユーザーのホームディレクトリを共有することによって、ユーザーのシステムからそのディレクトリをリモートでマウントできる。 | 44ページの「ユーザーのホームディレクトリを共有する方法」 |
| 5. ユーザーのホームディレクトリのマウント | ユーザーのホームディレクトリをユーザーのシステムに mount コマンドにより手作業でマウントする。 | 46ページの「ユーザーのホームディレクトリをマウントする方法」 |

ユーザー情報データシート

ユーザーアカウントを追加する前に、ユーザーに関する情報を以下のような形式で記録しておくると便利です。

| 項目 | 記入欄 |
|---------|-------|
| ユーザー名 | _____ |
| ユーザー ID | _____ |

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

▼ ユーザー初期設定ファイルをカスタマイズする方法

1. ユーザーのホームディレクトリを作成し、共有するシステムでスーパーユーザーになります。
2. 各タイプのユーザー用にスケルトンディレクトリを作成します。

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

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

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

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

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

注 `ls -a` コマンドを使うと `.` (ドット) ファイルを一覧表示できます。

4. 各ユーザータイプ用にユーザー初期設定ファイルを編集し、必要に応じてカスタマイズします。
ユーザー初期設定ファイルをカスタマイズする方法についての詳細は、22ページの「ユーザーの作業環境のカスタマイズ」を参照してください。
5. ユーザー初期設定ファイルのアクセス権を設定します。

```
# chmod 744 /shared-directory/skel/user-type/*
```

例 — ユーザー初期設定ファイルのカスタマイズ

次の例では、特定のタイプのユーザー向けの、`/export/skel/enduser` ディレクトリにある C シェルユーザー初期設定ファイルをカスタマイズします。

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

(.cshrc の編集 -- 33ページの「コード例 — .cshrc ファイルの例」を参照)
# chmod 744 /export/skel/enduser/.*
```

▼ Admintool を起動する方法

1. 次の条件がそろっていることを確認します。**Admintool** を使用するには、次のものがが必要です。

- ビットマップディスプレイモニター。Admintool ソフトウェアは、Sun ワークステーションの標準ディスプレイモニターなど、ビットマップ画面のコンソールを使用するシステムでだけ使用できます。
- OpenWindows 環境などの X Window システムが動作していること。
- `sysadmin` グループのメンバーであること (グループ 14)。

コンソールとして ASCII 端末を使用するシステムで管理作業を行うには、Solaris のコマンドを使用してください。詳細は、`useradd(1M)` のマニュアルページを参照してください。

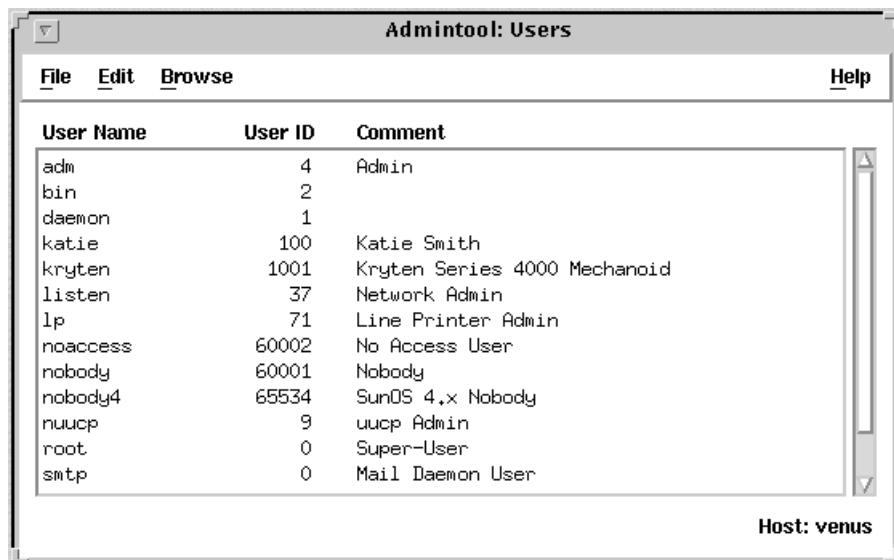
2. **Admintool** を起動します。

```
$ admintool &
```

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

例 — Admintool を起動する

これは「Admintool:ユーザー (Users)」メインウィンドウです。このウィンドウを使用すれば、ユーザーアカウント情報を管理できます。



▼ グループを追加する方法

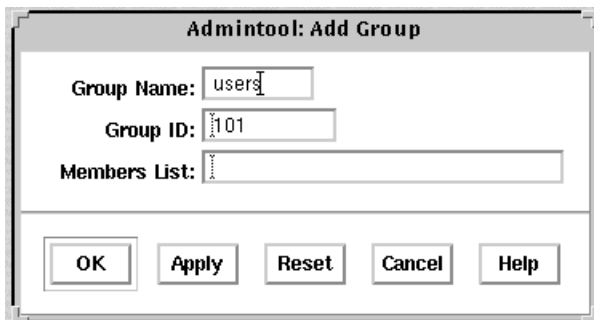
1. **Admintool** を起動します。
詳細については 40 ページの「Admintool を起動する方法」を参照してください。
2. 「ブラウズ (**Browse**)」メニューから「グループ (**Group**)」を選択します。
「Admintool:グループ (Groups)」ウィンドウが表示されます。
3. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。
「追加 (Add)」ウィンドウが表示されます。各フィールドについての説明は、「ヘルプ (**Help**)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
4. 新しいグループの名前を「グループ名 (**Group Name**)」テキストボックスに入力します。
5. 新しいグループのグループ ID を「グループ ID (**Group ID**)」テキストボックスに入力します。
グループ ID は固有でなければなりません。
6. (省略可能) 「メンバーリスト (**Members List**)」にユーザー名を入力します。
ユーザーのリストをグループに追加します。ユーザー名はカンマで区切ります。

7. 「了解 (OK)」 をクリックします。

「Admintool:グループ (Groups)」 ウィンドウに表示されるグループのリストに、新しいグループが追加されます。

例 — グループを追加する

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



▼ 新しいユーザーアカウントを追加する方法

1. (省略可能) 37ページの「ユーザー情報データシート」のユーザー情報データシートに記入します。
2. **Admintool** を起動します。
詳細については 40ページの「Admintool を起動する方法」を参照してください。
3. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。
「追加 (Add)」ウィンドウが表示されます。
4. 「追加 (**Add**)」ウィンドウの各項目を入力します。
各フィールドについての説明は、「ヘルプ (Help)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
5. 「了解 (**OK**)」 をクリックします。
「Admintool: ユーザー (Users)」のメインウィンドウに表示されるユーザーアカウントのリストに、新しいユーザーアカウントが追加されます。

次に進む手順

ユーザーのホームディレクトリを作成した場合、ユーザーのシステムがリモートでマウントできるように、そのディレクトリを共有しなければなりません。詳細については、44ページの「ユーザーのホームディレクトリを共有する方法」を参照してください。

ディスク容量に制限がある場合、ユーザーのホームディレクトリを含むファイルシステムで、そのユーザーのためのディスク割り当てを設定できます。ディスク割り当てを設定する方法については、『Solaris のシステム管理 (第 2 巻)』の「割り当て量の管理」を参照してください。

例 — 新しいユーザーアカウントを追加する

次の例は、ユーザー `kryten` をシステムに追加します。

Admintool: Add User

USER IDENTITY

User Name:

User ID:

Primary Group:

Secondary Groups:

Comment:

Login Shell:

ACCOUNT SECURITY

Password:

Min Change: days

Max Change: days

Max Inactive: days

Expiration Date:

(dd/mm/yy)

Warning: days

HOME DIRECTORY

Create Home Dir:

Path:

▼ ユーザーのホームディレクトリを共有する方法

1. ホームディレクトリを含むシステムでスーパーユーザーになります。
2. 次のように入力して、mountd デーモンが動作していることを確認します。

```
# ps -ef | grep mountd
root 176 1 0 May 02 ? 0:19 /usr/lib/nfs/mountd
```

mountd デーモンが動作している場合、次の行が表示されます。

```
/usr/lib/nfs/mountd
```

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

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

4. システム上で共有されているファイルシステムを一覧表示します。

```
# share
```

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

| | |
|----------------------------------|------|
| ユーザーのホームディレクトリを含む ファイルシステムの状態 | 次の手順 |
|----------------------------------|------|

| | |
|------------|-----------|
| すでに共有されている | 次の確認手順へ進む |
|------------|-----------|

| | |
|----------|------------|
| 共有されていない | 45ページの手順 6 |
|----------|------------|

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

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

| | |
|--------------------|---|
| <i>file-system</i> | 共有するユーザーのホームディレクトリを含むファイルシステム。習慣上、このファイルシステムは /export/home になる。 |
|--------------------|---|

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

```
# shareall -F nfs
```

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

8. ユーザーのホームディレクトリが共有されていることを確認します。

```
# share
```

次に進む手順

ユーザーのホームディレクトリがユーザーのシステム上にない場合、それが配置されているシステムから、ユーザーのホームディレクトリをマウントしなければなりません。詳細については、46ページの「ユーザーのホームディレクトリをマウントする方法」を参照してください。

例 — ユーザーのホームディレクトリの共有

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

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

# shareall -F nfs
# share
-           /usr/dist           ro   ""
-           /export/home/user-name   rw   ""
```

▼ ユーザーのホームディレクトリをマウントする方法

1. ユーザーのホームディレクトリが共有されていることを確認します。詳細については、44ページの「ユーザーのホームディレクトリを共有する方法」を参照してください。
2. ユーザーのシステムにスーパーユーザーとしてログインします。
3. /etc/vfstab ファイルを編集して、次のようなユーザーのホームディレクトリ用のエントリを作成します。

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


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

/etc/vfstab ファイルにエントリを追加する方法については、第 28 章を参照してください。

4. ユーザーのホームディレクトリのマウント先を作成します。

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

5. ユーザーのホームディレクトリをマウントします。

```
# mountall
```

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

6. *mount* コマンドを使って、ホームディレクトリがマウントされているかどうかを確認してください。

例 — ユーザーのホームディレクトリのマウント

```
# vi /etc/vfstab
(venus:/export/home/ripley - /export/home/ripley
nfs - yes rw の 1 行を追加する)
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid/
largefiles on Tue Jun 2 12:37:36 1998
```

(続く)

続き

```

/usr on /dev/dsk/c0t3d0s6 read/write/setuid/
largefiles on Tue Jun 2 12:37:36 1998
/proc on /proc read/write/setuid on Tue Jun 2 12:37:36 1998
/dev/fd on fd read/write/setuid on Tue Jun 2 12:37:38 1998
/opt on /dev/dsk/c0t3d0s5 setuid/read/write/
largefiles on Tue Jun 2 12:37:38 1998
/tmp on swap read/write on Jun 2 12:37:39 1998
/export/home/ripley on venus:/export/home/ripley /read/write/remote
on Jun 2 12:37:40 ...

```

ユーザーアカウントの管理

表 2-2 作業マップ：ユーザーアカウントの管理

| 作業 | 説明 | 手順の説明 |
|-----------------|--|---|
| 1. グループの変更 | 「グループ」ウィンドウの「編集 (Edit)」メニューから「変更 (Modify)」を選択して、グループ名またはグループ内のユーザーを変更する。 | 49ページの「グループを変更する方法」 |
| 2. グループの削除 | 「グループ」ウィンドウの「編集 (Edit)」メニューから「削除 (Delete)」を選択して、グループを削除する。 | 50ページの「グループを削除する方法」 |
| 3. ユーザーアカウントの変更 | <p>ユーザーアカウントを無効にする</p> <p>一時的にユーザーアカウントを無効にしたい場合、「変更 (Modify)」ウィンドウの「パスワード (Password)」メニューでそのユーザーアカウントをロックする。</p> <p>ユーザーのパスワードを変更する</p> <p>ユーザーのパスワードを変更したい場合、「変更 (Modify)」ウィンドウの「パスワード (Password)」メニューを使用する。</p> | <p>52ページの「ユーザーアカウントを無効にする方法」</p> <p>54ページの「ユーザーのパスワードを変更する方法」</p> |

表 2-2 作業マップ：ユーザーアカウントの管理 続く

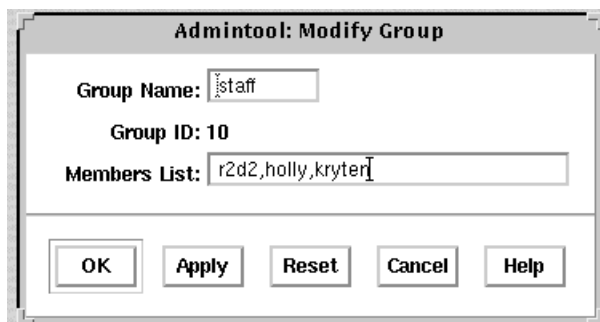
| 作業 | 説明 | 手順の説明 |
|-----------------|--|------------------------------------|
| | パスワード有効期限を変更する 定期的にユーザーのパスワードを変更させたい場合、「変更 (Modify)」ウィンドウでユーザーのパスワード有効期限フィールドを変更する（「アカウントのセキュリティ設定(Account Security)」カテゴリ）。 | 55ページの「ユーザーアカウントのパスワード有効期限を変更する方法」 |
| 4. ユーザーアカウントの削除 | 「ユーザー」ウィンドウの「編集(Edit)」メニューから「削除 (Delete)」を選択して、ユーザーアカウントを削除する。 | 56ページの「ユーザーアカウントを削除する方法」 |

▼ グループを変更する方法

1. **Admintool** を起動して、「ブラウザ (**Browse**)」メニューからグループを選択します。
詳細については 40ページの「Admintool を起動する方法」を参照してください。
2. 「**Admintool: グループ (Groups)**」ウインドウから変更するグループエントリを選択します。
3. 「**編集 (Edit)**」メニューから「**変更 (Modify)**」を選択します。
指定したグループエントリを含む「**変更 (Modify)**」ウィンドウが表示されます。
4. グループ名か、またはグループ内のユーザーを変更します。
ユーザー名はカンマで区切らなければなりません。各フィールドについての説明は、「ヘルプ (Help)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
5. 「**了解 (OK)**」をクリックします。
メインウインドウに表示されるグループ情報が更新されます。

例 — グループを変更する

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



▼ グループを削除する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「グループ (**Groups**)」を選択します。
詳細については 40ページの「Admintool を起動する方法」を参照してください。
2. 「グループ」ウインドウから削除したいグループエントリを指定します。
3. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
ウインドウが表示され、削除してよいかどうかをたずねられます。
4. 「了解 (**OK**)」をクリックします。
「グループ」ウインドウからそのグループエントリが削除されます。

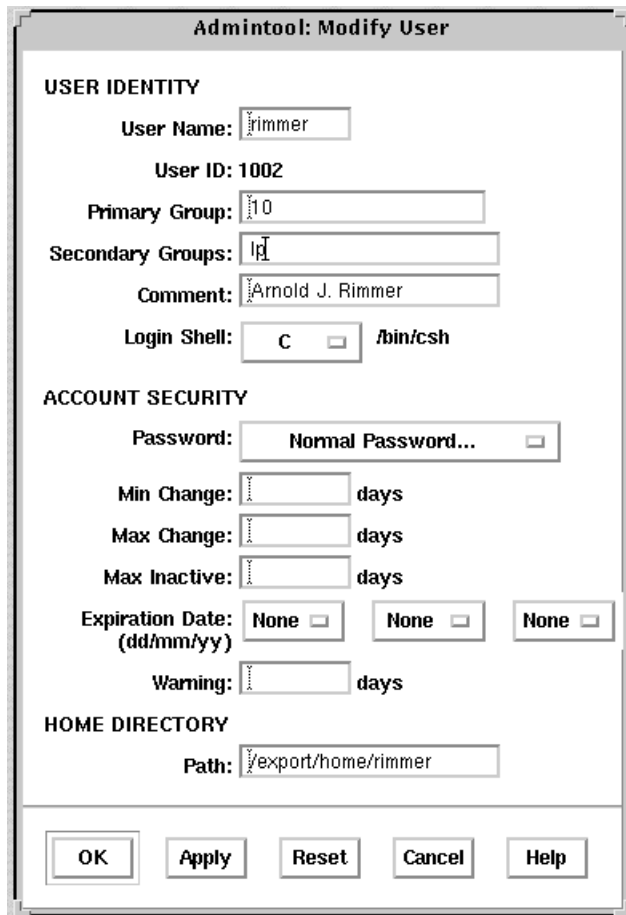
▼ ユーザーアカウントを変更する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細については 40ページの「Admintool を起動する方法」を参照してください。
2. 「ユーザー (**Users**)」ウインドウから変更するユーザーアカウントエントリを選択します。

3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したユーザーアカウントエントリを含む「変更 (**Modify**)」ウィンドウが表示されます。
4. ユーザーアカウントを変更します。
各フィールドについての説明は、「ヘルプ (**Help**)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。パスワードやパスワード有効期限の変更など、「アカウントのセキュリティ設定 (**Account Security**)」フィールドの任意の項目を変更することができます。詳しい手順については、以下の項を参照してください。
 - 52ページの「ユーザーアカウントを無効にする方法」
 - 54ページの「ユーザーのパスワードを変更する方法」
 - 55ページの「ユーザーアカウントのパスワード有効期限を変更する方法」
5. 「了解 (**OK**)」をクリックします。
6. 「ユーザー (**Users**)」ウィンドウで、変更したユーザーアカウントのエントリをダブルクリックして、変更が行われていることを確認してください。確認後、そのまま「取消し(**Cancel**)」をクリックしてウィンドウを閉じてください。

例 — ユーザーアカウントを変更する

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



▼ ユーザーアカウントを無効にする方法

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

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細については 40 ページの「Admintool を起動する方法」を参照してください。
2. 無効にするユーザーアカウントエントリを選択します。

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

例 — ユーザーアカウントを無効にする

次の例は、`rimmer` ユーザーアカウントを無効にしています。

Admintool: Modify User

USER IDENTITY

User Name:

User ID: 1002

Primary Group:

Secondary Groups:

Comment:

Login Shell: /bin/csh

ACCOUNT SECURITY

Password:

Min Change: days

Max Change: days

Max Inactive: days

Expiration Date: (dd/mm/yy)

Warning: days

HOME DIRECTORY

Path:

▼ ユーザーのパスワードを変更する方法

1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。
詳細については 40 ページの「Admintool を起動する方法」を参照してください。
2. パスワード変更が必要なユーザーアカウントエントリを選択します。
3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したユーザーアカウントエントリを含む「変更 (**Modify**)」ウィンドウが表示されます。

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

例 — パスワードの変更

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



▼ ユーザーアカウントのパスワード有効期限を変更する方法

1. **Admintool** を起動します。「ブラウズ (Browse)」メニューから「ユーザー (Users)」を選択します。
詳細については 40ページの「Admintool を起動する方法」を参照してください。
2. パスワード有効期限の変更が必要なユーザーアカウントエントリを選択します。
3. 「編集 (Edit)」メニューから「変更 (Modify)」を選択します。
指定したユーザーアカウントエントリを含む「変更(Modify)」ウィンドウが表示されます。
4. パスワードの有効期限に関係のある以下のフィールドを変更します。
 - 最短有効日数 (Min Change)

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

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

5. 「了解 (OK)」をクリックします。

例 — ユーザーアカウントのパスワード有効期限の変更

次の例では、ユーザーは少なくとも 1 日 (Min Change) だけパスワードを保持しなければなりません。また、ユーザーは 60 日 (Max Change) ごとにパスワードを変更しなければなりません。ユーザーは、アカウントが 10 日 (Max Inactive) を超える期間無効である場合、パスワードを変更しなければなりません。

The screenshot shows a window titled "Admintool: Users" with a menu bar containing "File", "Edit", "Browse", and "Help". The main area displays a table of users with columns for "User Name", "User ID", and "Comment".

| User Name | User ID | Comment |
|-----------|---------|------------------------------|
| adm | 4 | Admin |
| bin | 2 | |
| daemon | 1 | |
| katie | 100 | Katie Smith |
| kryten | 1001 | Kryten Series 4000 Mechanoid |
| listen | 37 | Network Admin |
| lp | 71 | Line Printer Admin |
| noaccess | 60002 | No Access User |
| nobody | 60001 | Nobody |
| nobody4 | 65534 | SunOS 4.x Nobody |
| nuucp | 9 | uucp Admin |
| root | 0 | Super-User |
| smtp | 0 | Mail Daemon User |

Host: venus

▼ ユーザーアカウントを削除する方法

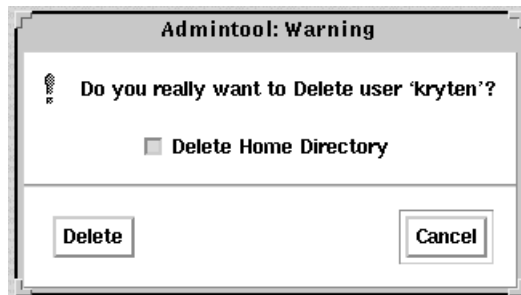
1. **Admintool** を起動します。「ブラウズ (**Browse**)」メニューから「ユーザー (**Users**)」を選択します。

詳細については 40 ページの「Admintool を起動する方法」を参照してください。

2. メインウィンドウから削除するユーザーアカウントエントリを選択します。
3. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
「削除 (Delete)」ウィンドウが表示され、そのユーザーアカウントを削除してもよいかどうかをたずねられます。
4. (省略可能) チェックボックスをクリックして、ユーザーのホームディレクトリとその内容を削除します。
5. ユーザーアカウントを削除する準備が出来ていれば「了解 (**OK**)」をクリックします。そのユーザーアカウントエントリが、「**Admintool: ユーザー (Users)**」メインウィンドウから削除されます。

例 — ユーザーアカウントを削除する

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



Solaris ユーザー登録

Solaris ユーザー登録は、新しい Solaris リリース、アップグレードのオファ、およびプロモーションについての情報を入手するためのツールです。このグラフィカルユーザーインターフェース (GUI) は、ユーザーが最初にユーザーのデスクトップにログインしたときに、自動的に起動されます。この GUI によって、ユーザーは現在登録することも、将来登録することも、あるいは登録しないことも可能です。ユーザー登録により、サンはユーザーの Solaris バージョン、タイプ、プラットフォーム、ハードウェアおよび、ロケールの情報を入手します。

SolarisSM SolveSM へのアクセス

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

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

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

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

4. Solaris Solve サイトにアクセスします。

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

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

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

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

Solaris ユーザー登録の問題の解決

この節では、Solaris ユーザー登録に関連した問題を解決する方法について説明します。

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

表 2-3 登録時の問題と対処方法

| 問題 | 対処方法 |
|--|--|
| 登録フォームが初期化されない。Web ページウィンドウに、システム管理者に登録設定を妨げている問題を解決してもらうように促すメッセージが表示される。 | 登録ファイルが失われていないかチェックする。 |
| フォームを電子メールで送信できない。ダイアログボックスに、システム管理者に問題を解決してもらうように促すメッセージが表示される。 | 電子メールが正しく設定されているかどうかをチェックする。完成した登録フォームを電子メールで送信するには CDE が必要なため、CDE がユーザーのシステム上にあるかを確認する。別の方法としてフォームを印刷して FAX か郵便で送信することもできる。 |
| フォームを印刷できない。ダイアログボックスに、システム管理者に問題を解決してもらうように促すメッセージが表示される。 | プリンタが正しく構成されているかどうかをチェックする。別の方法として電子メールでフォームを送信することもできる。 |
| フォームを保存できない。ダイアログボックスに、登録は成功したが、将来登録を更新するときに登録情報を呼び出せないことが示される。 | ユーザーのホームディレクトリをチェックする。必要な処置は、システムの構成によって異なる。 |
| Solaris Solve のログイン ID とパスワードを忘れてしまった。 | SolarisSolve@sun.com に問題を説明するメールを英語で送るか、59ページの「Solaris ユーザー登録をやり直す方法」を参照のこと。 |
| 登録プロセスをやり直したい。 | 59ページの「Solaris ユーザー登録をやり直す方法」を参照のこと。 |

▼ Solaris ユーザー登録をやり直す方法

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

1. \$HOME/.solregis ディレクトリに移動します。

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

2. uprops ファイルを削除します。

```
% rm uprops
```

3. 登録プロセスを再起動します。

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

▼ ユーザー登録を無効にする方法

表 2-4 に、Solaris ソフトウェアのインストールの前後において、ユーザー登録を無効にする方法を示します。Solaris ユーザー登録を無効にする前に、組織のシステム管理者をユーザー登録しておくことをお勧めします。

表 2-4 ユーザー登録を無効にする方法

| ユーザー登録を無効にするとき | 方法 | 参照先 |
|--------------------------|--|--|
| Solaris ソフトウェアをインストールする前 | <ul style="list-style-type: none"> ■ SUNWsregu パッケージを選択解除する (対話式インストール)。 ■ SUNWsregu パッケージをインストールしないように、カスタム JumpStart プロファイルを変更する。 ■ 1つまたは複数のシステム上の /etc/default ディレクトリに、DISABLE=1 という行を持つ、solregis という名前のファイルを作成する finish スクリプトを作成して実行する。 | <p>『Solaris のインストール (上級編)』</p> <p>solregis (1) のマニュアルページ</p> |
| Solaris ソフトウェアをインストールした後 | <ul style="list-style-type: none"> ■ pkgrm コマンドを使用して SUNWsregu パッケージを削除する。 ■ /etc/default ディレクトリに solregis ファイルを追加する (カスタム JumpStart インストールのみ)。 | <p>第 17 章</p> <p>『Solaris のインストール (上級編)』</p> <p>solregis (1) のマニュアルページ</p> |

パート II サーバーとクライアントサポートの管理

このパートでは、Solaris 環境においてサーバーとクライアントのサポートを管理する方法について説明します。次の章で構成されています。

第 3 章

この章ではネットワークでサーバーとクライアントサポートを管理する場合について、管理者を対象に概要を説明します。サポートを追加できるいろいろなシステムタイプと、選択する場合のガイドラインを説明します。また、Solstice ホストマネージャで行えること、行えないことについても説明します。

第 4 章

この章では Solstice ホストマネージャによりサーバーとクライアントのサポートを追加したり、管理したりする手順を説明します。



サーバーとクライアントサポートの管理の概要

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

この章の内容は次のとおりです。

- 66ページの「サーバーとクライアントの作業についての参照先」
- 66ページの「サーバーとクライアントとは」
- 66ページの「サポートとは」
- 67ページの「システムタイプの概要」
- 70ページの「システムタイプ選択のガイドライン」
- 71ページの「サーバーとクライアントサポートの管理ツール」
- 72ページの「ホストマネージャでできること」
- 77ページの「ホストマネージャの制限事項」

サーバーとクライアントサポートの追加と管理の手順についての詳細は、第 4 章を参照してください。

ネームサービスの設定の概要については、『*Solstice AdminSuite 2.3 管理者ガイド*』を参照してください。

サーバーとクライアントの作業についての参照先

サーバーとクライアントのサービスを設定する手順の説明は、次を参照してください。

- 第4章

サーバーとクライアントとは

ネットワーク上のシステムは、次の2つに分類できます。

- サーバー – ネットワーク上の他のシステムにサービスを提供するシステムです。ファイルサーバー、ブートサーバー、データベースサーバー、ライセンスサーバー、印刷サーバー、インストールサーバー、さらに、特定のアプリケーションのサーバーなどもあります。この章では、サーバーとは、ネットワーク上の他のシステムにファイルシステムとインストールソフトウェアを提供するシステムのことを意味します。
- クライアント – サーバーから提供されるリモートサービスを利用するシステムです。クライアントによってはディスク容量に制限があったり、また、まったくディスクを持たず、サーバーから提供されるファイルシステムに依存するものもあります。ディスクレスクライアント、AutoClient および JavaStation™ システムがこの部類に含まれます。

また別のサーバーは、サーバーが提供するリモートサービス(インストールソフトウェアなど)を利用しながらも、サーバーに依存しなくても機能するものがあります。この種類のクライアントには、ルート(/)、/usr、/export/home ファイルシステムとスワップ空間をそのハードディスクに含むスタンドアロンシステムがあります。

サポートとは

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

- システムをネットワークに認識させる (ホスト名とイーサネットアドレス情報)。
- システムをリモートからインストールおよびブートできるインストールサービスを提供する。
- ディスク領域が限られているか、まったくないシステムに、オペレーティングシステム (OS) のサービスを提供する。

システムタイプの概要

システムタイプは基本的に、ルート (/) と /usr ファイルシステム (スワップ領域を含む) にアクセスする方法によって決まります。たとえば、スタンドアロンとサーバーシステムでは、これらのファイルシステムをローカルディスクからマウントしていますが、その他のクライアントでは、これらのファイルシステムをリモートからマウントし、サーバーから提供されるサービスに依存しています。表 3-1 に、各システムタイプの相違点を要約します。

表 3-1 システムタイプの概要

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

表 3-1 システムタイプの概要 続く

| システムタイプ | ローカルファイルシステム | ローカルスワップ領域 | リモートファイルシステム | ネットワーク利用度 | 相対性能 |
|-----------------|--------------------------------------|------------|--|-----------|------|
| ディスクレスクライアント | なし | なし | ルート (/) スワップ領域 /usr /home | 高 | 低 |
| JavaStation | なし | なし | /home | 低 | 高 |
| AutoClient システム | キャッシュされたルート (/) キャッシュされた /usr | あり | /var | 低 | 高 |

サーバー

サーバーシステム上には、次のファイルシステムがあります。

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

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

- サーバーと異なるリリースを実行したい、またはサーバーと異なるプラットフォームを持つディスクレス、JavaStation または AutoClient に対して提供されるオペレーティングシステム (OS) サービス。
- ネットワークに接続されたシステムがリモートインストールを実行するのに必要な Solaris CD のイメージとブート用ソフトウェア。

- ネットワークに接続されたシステムがカスタム JumpStart インストールを行うのに必要な JumpStart ディレクトリ。

スタンドアロンシステム

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

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

注・スタンドアロンシステムに必要な4つのファイルシステムを保持するには、十分なディスク領域が必要です。

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

ディスクレスクライアント

「ディスクレスクライアント」は自分のディスクを持たないため、サーバー上のソフトウェアおよび記憶領域に全面的に依存します。したがって、ルート (/)、/usr、/home の各ファイルシステムをサーバーからリモートマウントします。

ディスクレスクライアントを使用すると、大量のネットワークトラフィックが発生します。これは、オペレーティングシステムのソフトウェアや仮想メモリ領域をネットワーク経由で継続的に取得する必要があるからです。ディスクレスクライアントは、ネットワークから切り離されたり、サーバーで障害が発生したりすると、機能できなくなります。

JavaStation クライアント

JavaStation は、システム管理が不要になるように設計されたクライアントです。このクライアントは JavaTM を最適化します。つまり、JavaStation クライアントはネットワークの利点を活用して、Java アプリケーションとそのサービス、完全に統合さ

れたシステムとネットワーク管理を提供します。JavaStation にはローカルの管理が必要ありません。つまり、ブート、管理、データの格納は、サーバーが処理します。

AutoClient システム

AutoClient システムは、インストールと管理の面では、ディスクレスクライアントとほとんど同じです。AutoClient システムには次のような特徴があります。

- スワップ空間、およびルート (/) と /usr のファイルシステムをサーバーからキャッシュするために、最低 100M バイトのローカルディスク領域が必要です。
- サーバーが利用できないときに、自分のキャッシュへのアクセスを継続できるように設定できます。
- 他のファイルシステムやアプリケーションにアクセスするのに、サーバーに依存します。
- 保存の必要があるデータを持たないので、フィールド交換ユニット (FRU) として使用できる。

注 - ネットワークに追加したい AutoClient システムごとにライセンスを取得しなければなりません。ライセンスについては、『Solstice AdminSuite 2.3 ご使用にあたって』を参照してください。

システムタイプ選択のガイドライン

次に挙げる特徴に基づいて各システムタイプを比較すれば、環境に適したシステムタイプを決定できます。

- 集中管理
 - システムをフィールド交換ユニット (FRU) として扱うことができるか？ つまり、障害が発生したとき、時間のかかるバックアップ/復元操作をせずに、また、システムデータを喪失したりせずに、新しいシステムと迅速に交換できる。
 - システムにバックアップは必要か？多数のデスクトップシステムをバックアップするには、大量の時間と資源が必要になる。

- システムのデータを中央のサーバーから変更できるか？
- クライアントシステムのハードウェアを操作することなく、中央のサーバーから迅速かつ簡単にシステムをインストールできるか？
- 性能
 - この構成の性能はデスクトップを使用する場合に十分か？
 - システムをネットワークに追加したとき、すでにネットワークに接続されている他のマシンの性能に影響を与えるか？
- 必要なディスク容量
 - この設定を効率的に運用するのに必要なディスク領域はどのくらいか？

表 3-2 は、上記の各項目についてシステムタイプ別に比較したものです。1 が最も効率が高く、4 が最も効率が低いことを意味します。

表 3-2 システムタイプの比較

| システムタイプの比較 | 集中管理 | 性能 | 必要なディスク容量 |
|--------------------|------|----|-----------|
| スタンドアロンシステム | 4 | 1 | 4 |
| ディスクレスクライアント | 1 | 4 | 1 |
| AutoClient システム | 1 | 2 | 2 |
| JavaStation クライアント | 1 | 1 | 1 |

サーバーとクライアントサポートの管理ツール

以前の Solaris リリースでは、システム管理ツールを使用してサーバーとクライアントサポートを管理できました。Solaris 2.5 とその互換バージョンでは、Solstice ホストマネージャを使用しなければなりません。このツールは使いやすく、次のネームサービスをサポートしています。

- NIS+ テーブル
- NIS マップ
- ローカルの /etc 内のファイル

ホストマネージャでできること

ホストマネージャは、ネットワークでサーバーとクライアントサポートを追加したり管理できるグラフィカルユーザーインターフェースです。NIS+ のようなネームサービスを使用すれば、システム情報を集中的に管理できるので、ホスト名などの重要なシステム情報をネットワーク上の各システムに重複して持つ必要がなくなります。

ホストマネージャにより次のことが行えます。

- サポートの追加と変更
- システムタイプの更新
- システムタイプの変換
- OS サービスの追加と削除
- リモートインストールサービスの設定
- タスクを待ち行列に入れる

サポートの追加と管理

ホストマネージャによって、次のシステムタイプのサポートを追加または変更することができます。

- Solaris AutoClient システム
- Solaris ディスクレスクライアント
- Solaris スタンドアロンシステム
- Solaris OS サーバー
- JavaStation (変更のみ)

表 3-3 に、Solstice AdminSuite 2.3 のホストマネージャがサポートするサーバーとクライアントの構成を示します。

表 3-3 サポートされるサーバーとクライアントの構成

| システム | 追加できる OS サービスとサポート | 対象となるリリース |
|--|---------------------------|---|
| Solaris 2.4 およびその互換バージョンが動作している x86 サーバー | SPARC クライアント ¹ | Solaris 2.3 およびその互換バージョン |
| | x86 クライアント | Solaris 2.4 およびその互換バージョン |
| Solaris 2.3 およびその互換バージョンが動作している SPARC サーバー | SPARC クライアント ¹ | SunOS 4.1 およびその互換バージョン、Solaris 2.3 およびその互換バージョン |
| | x86 クライアント | Solaris 2.4 およびその互換バージョン |

1. AutoClient システムは、Solaris 2.4 とその互換バージョンでのみサポートされる。

注 - SunOS 4.1 リリースおよびその互換バージョンは、Sun4、Sun4c、Sun4m プラットフォームグループの SPARC システムでのみサポートされます。

システムタイプの更新

ホストマネージャは最初に、すでにシステムに追加されているシステムタイプに、`generic` というマークを付けます。ただし、「ファイル (File)」メニューの「システムタイプの更新 (Update System Types)」を選択して、すでに追加されているシステムを調べ、自動的にそのシステムタイプを決定することもできます。ホストマネージャがシステムタイプを決定できない場合 (たとえば、システムが Solaris ソフトウェアを実行していない場合)、システムには `generic` というマークが付けられたままです。

注 - ホストマネージャが自動的にシステムタイプを更新できるようにするには、Solaris 2.5 およびその互換バージョンを稼動しているすでに追加されているシステムに、Solstice AdminSuite がインストールされている必要があります。

システムタイプ情報は、ローカルの `/etc` またはネームサービスデータベース内の `bootparams` ファイルに格納されます。ホストマネージャは、既存の `bootparams`

エントリを更新するか、たとえば次のような `mars` という Solaris スタンドアロンシステムを追加します。

```
mars boottype=:st
```

システムタイプの変換

ホストマネージャにより、あるシステムタイプを別のシステムタイプに変換できます。表 3-4 に実行できる変換を示します。

表 3-4 システムタイプの変換

| 変換できるシステム | 変換先のシステム |
|-----------------|---|
| スタンドアロンシステム | AutoClient システム、または OS サーバー |
| データレスシステム | AutoClient システム、または OS サーバー |
| AutoClient システム | スタンドアロンシステム |
| Generic システム | スタンドアロンシステム、AutoClient システム、または OS サーバー |

スタンドアロンシステムから OS サーバーへの変換時、Solaris 7 またはその互換バージョンの OS サービスを追加できます。

OS サービスの追加

Solaris OS サーバーは、クライアントをサポートできるオペレーティングシステムサービスを提供するサーバーです。ホストマネージャを使用すれば、OS サーバーのサポートを追加するか、スタンドアロンシステムを OS サーバーに変換することができます。

サポートしたいプラットフォームと Solaris リリースごとに、特定の OS サービスを OS サーバーに追加しなければなりません。たとえば、Solaris 7 を実行している SPARC Sun4m システムをサポートするには、Sun4m/Solaris 7 OS サービスを OS サーバーに追加しなければなりません。また、Solaris 7 を実行している SPARC Sun4c システムまたは x86 システムをサポートするには、さらに OS サービスを追

加する必要があります。これらのシステムは異なるプラットフォームグループに含まれるためです。

OS サービスを追加するには、適切な Solaris CD イメージを使用できなければなりません。

注 - ホストマネージャを使用して、SunOS 4.0 またはその互換バージョンを実行するディスククライアントのサポートを追加することはできますが、SunOS 4.0 およびその互換バージョンの OS サーバーを追加することはできません。install4x コマンドを使って OS サーバーに OS サービスを追加し、それからホストマネージャを使って SunOS 4.0 またはその互換バージョンのクライアントのサポートを追加しなければなりません。

OS サービスにパッチが適用されているときに OS サービスをサーバーに追加する

OS サービスを OS サーバーに追加するときに、サーバー上で動作している OS のバージョンと、追加しようとしている OS との間に整合性がないというエラーメッセージが表示される場合があります。このメッセージは、インストールされている OS に、以前パッチを適用したパッケージが含まれており、追加しようとしている OS サービスには、パッチを適用したパッケージが含まれていないときに表示されます (これは、パッチがパッケージに統合されているためです)。

たとえば、Solaris 7 またはその互換バージョンを実行しているサーバーがあるとします。また、このサーバーには、以前パッチを適用した Solaris 2.6 SPARC Sun4m OS サービスを含む OS サービスもロードされているものとします。このサーバーに、Solaris 2.6 SPARC Sun4c OS サービスを CD-ROM から追加しようとする、次のようなエラーメッセージが表示されます。

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

OS サービスの削除

OS サービスは、ホストマネージャを使用して、OS サーバーから削除できます。たとえば、Solaris 7 またはその互換バージョンを実行している SPARC Sun4m システ

ムをサポートする必要がなくなった場合、ホストマネージャを使用してこれらの OS サービスをサーバーから削除できます。

リモートインストールサービスの設定

ホストマネージャを使用すれば、Solaris インストールサービスをネットワーク上の他のシステムに提供できるようにシステムを設定できます。次のタイプのインストールサービスをシステムに設定できます。

- インストールサーバー – インストール用に Solaris CD イメージを (CD-ROM ドライブからか、ハードディスク上のコピーから) 他のシステムに提供する、ネットワーク上のシステム。
- ブートサーバー – ネットワーク上の他のシステムにブート情報を提供するシステム。通常、ブートサーバーとインストールサーバーは同じシステムです。
- プロファイルサーバー – カスタム JumpStart インストールを実行するための JumpStart ファイルを含むシステム。

注 - ブートサーバーとインストールサーバーは通常同じシステムを使用します。ただし、インストールするシステムがインストールサーバーと異なるサブネット上にある場合、ブートサーバーはインストール先のサブネット上になければなりません。

タスクを待ち行列に入れる

ホストマネージャを使用すれば、システムタイプの変換や OS サービスの追加などのタスクを待ち行列に入れることができます。これらのタスクを処理するには数分必要です。ホストマネージャを使用すれば、各タスクが完了するのを待たずに、タスクを実行するように設定できます。タスクの設定後、「ファイル (File)」メニューから「変更を保存 (Save Changes)」を選択します。ホストマネージャの進捗状況は、各タスクが処理されるごとに、ウィンドウの一番下にあるメッセージバーに表示されます。

root パスワードの設定

ホストマネージャを使用して Solstice AutoClient または Solaris ディスクレスクリアライアントを追加するときに、グループまたはユーザーのパスワードを設定するときと同じように GUI を使用して、root のパスワードを設定できます。

スクリプトの有効化

ホストマネージャを使用して Solstice AutoClient を追加する場合、AutoClient をサーバーに追加する前後にサーバーで実行するように、あるいは、キャッシュを AutoClient 上で構成する前後にクライアントで実行するようにスクリプトを有効にすることも可能です。

これらのスクリプトは、AutoClient システムの追加または削除をカスタマイズするために作成したスクリプトです。AdminSuite ソフトウェアが読み取るので、これらのスクリプトは /opt/SUNWadmd/Scripts ディレクトリになければなりません。

マルチホームホストの追加

ホストマネージャを使用すれば、複数のネットワークインタフェースを持つサーバー用に、マルチホームホストの別名を追加できます。たとえば、サーバーが複数のネットワーク上にあるために複数の IP アドレスを持っている場合、このサーバーは、マルチホームホストであると考えられます。ホストマネージャを使用すれば、複数の IP アドレスを 1 つのホストに指定して、そのホストをマルチホームホストとすることができます。

ホストマネージャの制限事項

表 3-5 にホストマネージャの制限事項と、その対処方法を示します。

表 3-5 ホストマネージャの制限事項と対処方法

| 制限 | 対処方法 |
|--|--|
| ホストマネージャは、すでに追加されているシステムタイプをすべて自動的に認識できるとは限らない。 | ホストマネージャを初めて使用するときは、「ファイル (File)」メニューから「システムタイプの更新 (Update System Type)」オプションを選択する。このオプションはネットワーク上のシステムを調べ、システムタイプを識別する。 |
| ホストマネージャは、SunOS 4.1 または互換バージョンのサービスを OS サーバーに追加できない。 | SunOS 4.1 またはその互換バージョンの CD イメージをマウントし、install4x コマンドを使って OS サービスを追加する。 |

表 3-5 ホストマネージャの制限事項と対処方法 続く

| 制限 | 対処方法 |
|---|--|
| ホストマネージャは SunOS 4.1 およびその互換バージョンのシステムにはリモートインストールサービスを提供できない。 | SunOS 4.1 およびその互換バージョンのシステムではローカルの CD-ROM ドライブからインストールする。 |
| ホストマネージャでは、既存のクライアントとサーバーに、パッチをインストールできない。(ただし、admclientpatch コマンドを使用してパッチスプールディレクトリを設定しておけば、ホストマネージャはこのスプールディレクトリを参照して、適切なパッチをすべての新しいホストに追加できる。) | 既存のサーバーとクライアントに最新のパッチを適用して更新するには、admclientpatch コマンドを使用してパッチスプールディレクトリを設定しておく。 |

ホストマネージャをスーパーユーザーとして実行する

ホストマネージャをスーパーユーザーで実行するとき、動作が若干異なることに気がつくでしょう。次のリストに、ホストマネージャをスーパーユーザーで実行するときの制限を説明します。

- ホストマネージャをスーパーユーザーで起動するとき、制限について説明するダイアログボックスが表示されます。
- ネームサービス選択ダイアログは強制的にローカルホストになり、テキストフィールドは編集できなくなります。
- ホストを追加するとき、ファイルサーバーは強制的にローカルホストになり、編集できなくなります。
- 「リモートインストール (Remote Install)」が「追加 (Add)」、「変更 (Modify)」、または「変換 (Convert)」ウィンドウで有効なとき、ブートサーバーは強制的にローカルホストになり、編集できなくなります。また、「インストールサーバー (Install Server)」は強制的にローカルホストになり、編集できなくなります。

サーバーとクライアントサポートの管理の手順

この章では、Solstice ホストマネージャを使用してサーバーとクライアントサポートを設定し、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。

- 81ページの「Solstice ホストマネージャを起動する方法」
- 83ページの「システムタイプを更新する方法」
- 84ページの「サポート追加時のデフォルトを設定する方法」
- 84ページの「スタンドアロンシステムまたは OS サーバーのサポートを追加する方法」
- 87ページの「システムを OS サーバーに変換する方法」
- 89ページの「SunOS 4.0 およびその互換バージョンの OS サービスを OS サーバーに追加する方法」
- 94ページの「Solaris 7 OS サービスを OS サーバーに追加する方法」
- 96ページの「AutoClient システムのサポートを追加する方法」
- 99ページの「ディスクレスクライアントサポートを追加する方法」
- 103ページの「システムサポートを変更する方法」
- 104ページの「システムサポートを削除する方法」

サーバーとクライアントサポートの管理の概要については、第 3 章を参照してください。Solstice ホストマネージャの使用方法については、『*Solstice AdminSuite 2.3 管理者ガイド*』を参照してください。

サーバーとクライアントサポートの追加

表 4-1 作業マップ: サーバーとクライアントサポートの追加

| 作業 | 説明 | 手順の説明 |
|------------------------|---|--|
| 1. システムタイプの更新 | (省略可能) ホストマネージャが、すでに追加されているすべてのシステムタイプを認識できることを確認する。 | 83ページの「システムタイプを更新する方法」 |
| 2. サポート追加時のデフォルトの設定 | (省略可能) クライアントサポートを追加する前に、ホストマネージャの「編集 (Edit)」メニューから「デフォルトの設定 (Set Defaults)」を選択して、「追加 (Add)」ウィンドウのデフォルト値を設定する。 | 84ページの「サポート追加時のデフォルトを設定する方法」 |
| 3. スタンドアロンシステムのサポートの追加 | ホストマネージャの「編集 (Edit)」メニューから「追加 (Add)」を選択して、スタンドアロンシステムのサポートを追加する。また、「編集 (Edit)」メニューから「スタンドアロンへの変換 (Convert to Standalone)」を選択して、AutoClient システムまたは generic システムをスタンドアロンシステムに変換できる。 | 84ページの「スタンドアロンシステムまたは OS サーバーのサポートを追加する方法」 |
| 4. OS サーバーのサポートの追加 | OS サーバーのサポートを追加する。 ホストマネージャの「編集 (Edit)」メニューから「追加 (Add)」を選択して、OS サーバーのサポートを追加する。 スタンドアロンシステムを OS サーバーに変換する。 ホストマネージャの「編集 (Edit)」メニューから「変換 (Convert)」を選択して、スタンドアロンシステムを OS サーバーに変換する。変換時に Solaris 7 OS サービスを追加できる。 | 84ページの「スタンドアロンシステムまたは OS サーバーのサポートを追加する方法」 87ページの「システムを OS サーバーに変換する方法」 |

表 4-1 作業マップ: サーバーとクライアントサポートの追加 続く

| 作業 | 説明 | 手順の説明 |
|-------------------------|---|---|
| 5. システムを OS サーバーに変換する方法 | <p><i>SunOS 4.0</i> およびその互換バージョンの OS サービスを追加する。</p> <p><i>SunOS 4.0</i> およびその互換バージョンのディスクレスクライアントサポートを追加する必要があるときは、OS サーバーには適切な <i>SunOS 4.0</i> およびその互換バージョンの OS サービスが追加されていないなければならない。</p> <p><i>Solaris 7</i> OS サービスを追加する。</p> <p><i>Solaris 7</i> ディスクレスまたは <i>AutoClient</i> システムのサポートを追加する必要があるときは、OS サーバーには適切な <i>Solaris 7</i> OS サービスが追加されていないなければならない。</p> | <p>89ページの「<i>SunOS 4.0</i> およびその互換バージョンの OS サービスを OS サーバーに追加する方法」</p> <p>94ページの「<i>Solaris 7</i> OS サービスを OS サーバーに追加する方法」</p> |
| 6. クライアントのサポートの追加 | <p><i>AutoClient</i> システムのサポートを追加する。</p> <p>ホストマネージャの「編集 (Edit)」メニューから「追加 (Add)」を選択して、<i>AutoClient</i> システムのサポートを追加する。</p> <p>ディスクレスクライアントのサポートを追加する。</p> <p>ホストマネージャの「編集 (Edit)」メニューから「追加 (Add)」を選択して、ディスクレスクライアントのサポートを追加する。</p> | <p>96ページの「<i>AutoClient</i> システムのサポートを追加する方法」</p> <p>99ページの「ディスクレスクライアントサポートを追加する方法」</p> |

▼ Solstice ホストマネージャを起動する方法

1. 次の条件がそろっていることを確認します。**Solstice** ホストマネージャを使用するには、次のものがが必要です。

- Solstice AdminSuite ソフトウェアがインストールされていること。
- ビットマップディスプレイモニター。Solstice AdminSuite ソフトウェアは、Sun ワークステーションの標準ディスプレイモニターなど、ビットマップ画面のコンソールを使用するシステムでだけ使用できます。
- コンソールとして ASCII 端末を使用するシステムで管理作業を行うには、Solaris コマンドを使用してください。
- X Window システムがインストールおよび実行されていること。
- スーパーユーザー (root) 権限、または sysadmin グループ (グループ 14) のメンバシップと、NIS または NIS+ データベースを管理するのに必要なアクセス権。

注 - ネームサービスが NIS+ の場合、NIS+ の admin グループのメンバーでなければなりません。

2. **Solstice** 起動ツールを起動します。

```
$ solstice &
```

Solstice 起動ツールが表示されます。

3. 「**Host Manager**」アイコンをクリックします。



Host Manager

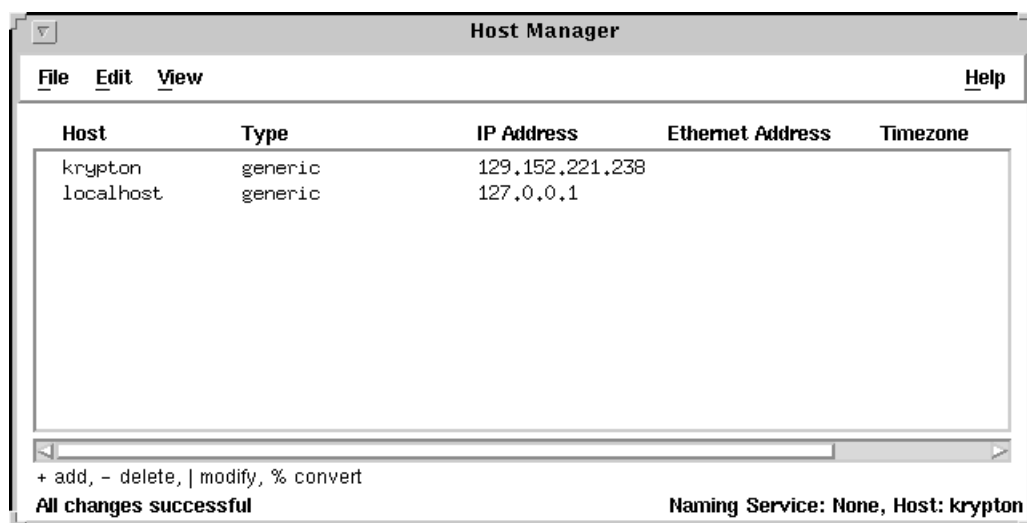
「読み込み (Load)」ウィンドウが表示されます。

4. ネットワークで使用するネームサービスを選択します。
5. ドメインまたはホストの名前が正しいかどうかをチェックします。

正しくない場合、アクセスする必要があるドメインかホストの名前を入力します。

6. 「了解 (OK)」をクリックします。
Solstice ホストマネージャのメインウィンドウが表示されます。

例 — ホストマネージャのメインウィンドウの例



▼ システムタイプを更新する方法

次の手順で、generic とみなされたシステムタイプを調べます。

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. ホストマネージャのメインウィンドウの「ファイル (File)」メニューから「システムタイプの更新 (Update System Types)」を選択します。
「システムタイプの更新 (Update System Types)」ウィンドウが表示されます。
3. 「更新 (Update)」をクリックして、**generic** のマークが付いているシステムを更新します。

4. ホストマネージャのメインウィンドウで、システムが **generic** に変換されていることを確認します。

▼ サポート追加時のデフォルトを設定する方法

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. 「編集 (**Edit**)」メニューから「デフォルトの設定 (**Set Defaults**)」を選択します。
「デフォルトの設定 (**Set Defaults**)」ウィンドウが表示されます。
3. 「デフォルトの設定 (**Set Defaults**)」ウィンドウに入力します。
ここで入力したデフォルトは、「追加 (**Add**)」ウィンドウの初期値になります。
各フィールドについての説明は、「ヘルプ (**Help**)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
4. 「了解 (**OK**)」をクリックします。

▼ スタンドアロンシステムまたは OS サーバーのサポートを追加する方法

概要と手順は次のとおりです。

- システム情報を追加します。
 - (省略可能) システムのリモートインストール機能を設定します。
 - (省略可能) システムをインストールします。
1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
 2. ホストマネージャのメインウィンドウの「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。

「追加 (Add)」ウィンドウが表示されます。

3. 「**Solaris Standalone**」または「**Solaris OS Server**」をシステムタイプとして選択し、システム情報を入力します。
システム情報には、「ホスト名 (Host Name)」から「時間帯 (Timezone)」まですべてのフィールドが含まれます。各フィールドについての説明は、「ヘルプ (Help)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
4. システムのリモートインストール機能を設定するには、85ページの手順 5 に進みます。それ以外の場合は、86ページの手順 10 に進みます。
5. 「リモートインストールを有効にする (**Enable Remote Install**)」をクリックします。
6. インストールサーバーを選択します。
デフォルトのインストールサーバーは、現在のホストです。別のホストをインストールサーバーに指定するには、「インストールサーバー (Install Server)」メニューから「その他 (Other)」を選択してください。
7. 「パスの設定 (**Set Path**)」をクリックして、インストールサーバー上の **Solaris CD** イメージへのパスを指定します
インストールサーバーがリモートシステムの場合、管理対象のシステムとして、最小限の設定が行われていなければなりません。

| 使用する媒体 | ボリュームマネージャによる管理 | 入力するパス |
|--|--------------------|---|
| Solaris CD イメージとして Solaris CD を使用 | Solaris CD は管理される | /cdrom/cdrom0、 /cdrom/cdrom0/s0、 /cdrom/cdrom0/s2 のいずれか |
| | Solaris CD は管理されない | Solaris CD のマウント先 |
| インストールサーバーのハードディスク上の Solaris CD のコピー (setup_install_server)。 | | Solaris CD のコピー先 |

8. 「**OS リリース (OS Release)**」メニューから、システムのアーキテクチャタイプと **OS** リリースを選択します。

アーキテクチャタイプはシステムのアーキテクチャタイプと一致していなければなりません。また、OS リリースは、システム上にリモートからインストールする Solaris のリリースと一致していなければなりません。

9. 必要に応じて、ブートサーバーまたはプロファイルサーバーを指定します。

デフォルト以外のサーバーを指定するには、メニューから「その他 (Other)」を選択します。プロファイルサーバーは、「プロファイルサーバー (Profile Server)」プルダウンメニューから選択します。ブートサーバー上のブート用ソフトウェアへのパス、またはプロファイルサーバー上のカスタム JumpStart ディレクトリを指定する必要があります。

カスタム JumpStart インストールの他の構成要素を設定して、ネットワークとシステムの情報を事前構成する方法については、『Solaris のインストール (上級編)』を参照してください。

10. 「追加 (**Add**)」ウィンドウ上の「了解 (**OK**)」をクリックします。

11. 「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択して、スタンドアロンシステムサポートまたは **OS** サーバーサポートを追加します。

ホストマネージャのメインウィンドウ内に、スタンドアロンシステムまたは OS サーバーが表示されます。

12. ホストマネージャのメインウィンドウで、**OS** サーバー情報が正しいことを確認します。

13. (省略可能) スタンドアロンシステムまたは **OS** サーバーをブートして、インストールします。

ブートおよびインストールについての詳細は、『Solaris のインストール (上級編)』を参照してください。

注 - OS サーバーをインストールする場合は、クライアントの数に応じて /export または /export/swap 用の領域を割り当てる必要があります。

例 — スタンドアロンシステム追加ウィンドウの入力例

次の例は、スタンドアロンシステム venus を追加するウィンドウです。

Host Manager: Add

Host Name: venus

IP Address: 129.152.225.2

Ethernet Address: 8:0:20:b:40:e9

System Type: Solaris Standalone

Timezone Region: United States

Timezone: Mountain

Remote Install: Enable Remote Install

Install Server: krypton

OS Release:

Boot Server: none

Profile Server: none

OK Apply Reset Cancel Help

次に進む手順

OS サーバーをインストールしたあと OS サービスを追加する場合は、94ページの「Solaris 7 OS サービスを OS サーバーに追加する方法」を参照してください。

▼ システムを OS サーバーに変換する方法

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。

81ページの「Solstice ホストマネージャを起動する方法」を参照してください。

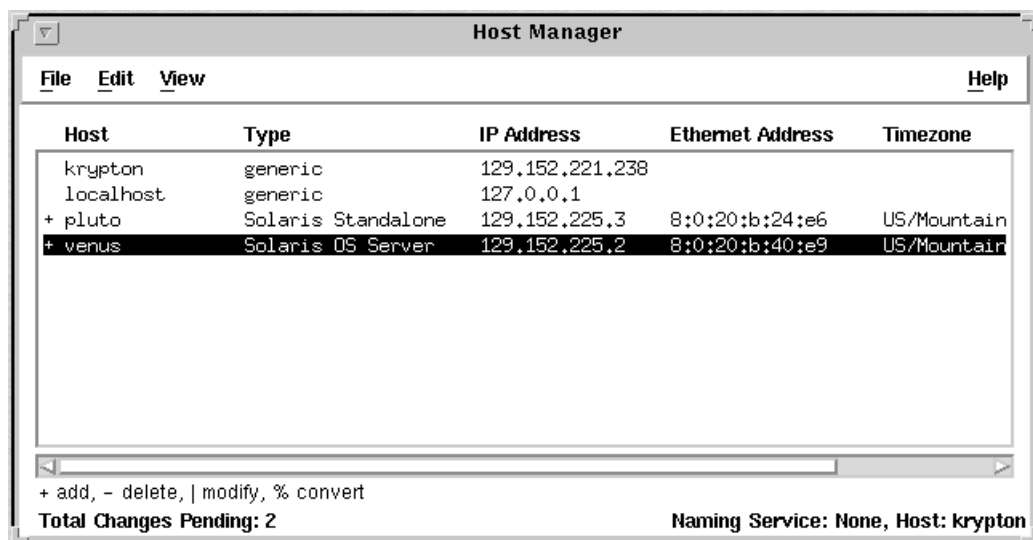
2. ホストマネージャのメインウィンドウからスタンドアロンシステムのホストエン
トリーを選択します。
スタンドアロンシステムと generic システムは、OS サーバーに変換できます。
3. 「編集 (Edit)」メニューから「変換 (Convert)」の「OS サーバーに (To OS
Server)」を選択します。
「変換 (Convert)」ウィンドウが表示されます。「ホスト名 (Host Name)」
フィールドには、選択したスタンドアロンシステムが表示されています。
4. 「OS サービス (OS Services)」ウィンドウの「追加 (Add)」ボタンをクリック
して、サービスを追加します。
5. 「パスの設定 (Set Path)」をクリックして、クライアントサービスを追加するた
めの Solaris CD イメージへのパスを指定します。
デフォルトのインストールサーバーは、現在のホストです。別のホストをインス
トールサーバーに指定するには、「インストールサーバー (Install Server)」メ
ニューから「その他 (Other)」を選択してください。リモートシステムは、管理
対象のシステムとして、最小限の設定が行われていなければなりません。

| 使用する媒体 | ボリュームマネージャによる管理 | 入力するパス |
|--|--------------------|---|
| Solaris CD イメージとして Solaris CD を使用 | Solaris CD は管理される | SPARC プラットフォーム上の /cdrom/cdrom0/s0 または /cdrom/cdrom0/s2、x86 プラット フォーム上の /cdrom/cdrom0 |
| | Solaris CD は管理されない | Solaris CD のマウント先 |
| インストールサーバーのハード ディスク上の Solaris CD のコ ピー setup_install_server を使用)。 | | setup_install_server で指定した Solaris CD のコピー先。 |

6. 追加するサービスのタイプを指定して、「追加 (Add)」をクリックします。
OS サービスは、OS サービスリストに追加されて、プラス記号 (+) のマークが
付けられます。これは、変更を保存するときに、追加されることを意味します。
「削除 (Delete)」ボタンを使用すれば、変更を保存する前に、OS サービスをリ
ストから削除できます。

7. 「変換 (Convert)」 ウィンドウ上の「了解 (OK)」 をクリックします。
8. 「ファイル (File)」 メニューから「変更を保存 (Save Changes)」 を選択して、スタンドアロンシステムを OS サーバーに変換します。
変換されたスタンドアロンシステムは、ホストマネージャのメインウィンドウ内に OS サーバーとして表示されます。
9. ホストマネージャのメインウィンドウで、ホストタイプが更新されていることを確認します。

例 — 変換ウィンドウの入力



▼ SunOS 4.0 およびその互換バージョンの OS サービスを OS サーバーに追加する方法

注 - SunOS 4.0 およびその互換バージョンの OS サービスを、x86 OS サーバーに追加することはできません。

概要と手順は次のとおりです。

- ディスクレスとデータレスクライアント用に SunOS 4.0 およびその互換バージョンの OS サービスを追加する OS サーバーを決定します。
 - 必要な SunOS 4.0 またはその互換バージョン (install4x コマンド) をインストールします。
1. ディスクレスクライアント用に **SunOS 4.0** およびその互換バージョンの **OS** サービスを追加する **OS** サーバーにスーパーユーザーとしてログインします。

2. **Solstice AdminSuite 2.3 CD** を **CD-ROM** ドライブに挿入します。

この手順では、システムがボリュームマネージャを実行していると仮定していません。

3. ディレクトリをソフトウェアのある位置に変更します。

```
# cd /cdrom/Solstice_sysmgt_2.3/4.x
```

4. **SunOS 4.0** またはその互換バージョンの、異機種インストールソフトウェアをインストールします。

```
# pkgadd -d `pwd` SUNWhinst
```

5. **Solstice AdminSuite 2.3 CD** を取り出します。

```
# cd  
# eject cd
```

6. **SunOS 4.0** またはそれと同等の **Solaris** リリースの **CD** を **CD-ROM** ドライブに挿入します。

この手順では、システムがボリュームマネージャを実行していて、CD-ROM ドライブが直接サーバーに接続されていると仮定しています。ボリュームマネージャは CD ディレクトリを /cdrom/volume1/s0 に自動的にマウントします。

7. **SunOS 4.0/4.1** のソフトウェアインストールプログラムを起動します。

```
# /usr/sbin/install4x -m /cdrom/volume1/s0 -e /export
```

メインメニューが表示されます。

```
*** 4.1* Install Main Menu ***
Choose an Architecture (then select modules to load):

Modules
                                Loaded   Selected
[a] sun4.sun4c.sunos.4.1.2      0           0
[b] sun4.sun4.sunos.4.1.2      0           0
[c] sun4.sun4m.sunos.4.1.2     0           0

or begin the loading process for all selected modules:

[L] Load selected modules

or abort without loading any modules:

[Q] Quit without loading

                                +-----+
                                | Disk Usage: |
                                |   0K Selected |
                                | 53634K Free  |
                                +-----+

Type any bracketed letter to select that function.

Type ? for help.
```

8. メインメニューで、鍵かっこで囲まれている文字を入力することによりサポートしたいアーキテクチャを指定します。

「Module Selection」メニューが表示されます。

```
Select sun4.sun4c.sunos.4.1.2 modules:
+[a] R proto root.....240K | [o] User_Diag.....6352K
+[b] R usr.....26240K | [p] Manual.....7456K
+[c] R Kvm.....4832K |+[q] D TLI..... 48K
+[d] R Install.....936K | [r] D RFS..... 912K
[e] D Networking.....1040K | [s] D Debugging.....2928K
[f] D System_V.....4008K | [t] SunView_Programmers....1840K
[g] D Sys.....5288K | [u] Shlib_Custom.....1376K
[h] C SunView_Users.....2664K | [v] Graphics.....1784K
[i] SunView_Demo.....512K |+[w] uucp.....608K+
[j] Text.....712K |+[x] Games..... 3136K
[k] Demo.....4264K | [y] Versatec.....5960K
[l] C OpenWindows_Users....25936K | [z] Security.....312K
[m] C OpenWindows_Demo.....4288K | [A] OpenWindows_Programmers.10200K
[n] C OpenWindows_Fonts ...7840K |

Module + = already loaded R = Required C= Common
```

(続く)

続き

```
Legend:  ** = selected for loading  D = Desirable  Others are optional
Select [a-A] or a Quick-Pick Option:

[1] All Required Modules  [4] All Optional Modules  | Disk Usage:  |
[2] All Desirable Modules [5] All Modules          |         0K Selected  |
[3] All Common Modules   |                          |         53634K Free  |
or [D] (done) to return to the main screen  +-----+
```

9. 鍵かっこで囲まれている文字を入力することにより、ロードするモジュールを指定します。

「Module Selection」画面では、ロードするモジュールのグループをすぐに選択できます。1を入力すると、必要なすべてのモジュールをロード用にマークします。2を入力すると、推奨されるモジュールをすべてマークします。3を入力すると、一般的にロードするモジュールをすべてマークします。4を入力すると、オプションのモジュールをすべてマークします。5を入力すると、「Module Selection」画面に表示されるすべてのモジュールをマークします。

10. Dを入力してメインメニューに戻ります。

メインメニューが表示されます。

```
*** 4.1* Install Main Menu ***

Choose an Architecture (then select modules to load):

                Modules
                Loaded  Selected
[a] sun4.sun4c.sunos.4.1.2      0      4
[b] sun4.sun4.sunos.4.1.2      0      3
[c] sun4.sun4m.sunos.4.1.2     0      1

or begin the loading process for all selected modules:

[L] Load selected modules

or abort without loading any modules:

[Q] Quit without loading

                +-----+
                | Disk Usage:  |
                |         0K Selected  |
                |         53634K Free  |
                +-----+
```

(続く)

```
Type any bracketed letter to select that function.
Type ? for help.
```

11. **L** と入力してソフトウェアをインストールします。
前に選択したモジュールがインストールされます。

```
Installing module 'proto root' [size: 248K]
      in directory /export/exec/proto.root.sunos.4.1.2 ...

Updating server databases ...

Press any key to continue:
```

12. モジュールがインストールされたあと、任意のキーを押してメインメニューに戻ります。
ロードされたモジュールがメインメニューに戻ります。
13. その他のアーキテクチャのサポートを追加するには、91ページの手順 8 から 93 ページの手順 11 までを繰り返します。終了するには、**Q** を入力します。

注 - SunOS 4.0 またはその互換バージョンの OS サービスを OS サーバーに追加する機能は、コマンド行にはありません。

14. `/export/SunOS_4.x` ディレクトリの内容をリストして、**SunOS 4.0** またはその互換バージョンの **SunOS OS** サービスが追加されていることを確認します。

次に進む手順

ディスクレスクライアント用に SunOS 4.0 またはそれと同等のサポートを追加するには、99ページの「ディスクレスクライアントサポートを追加する方法」を参照してください。

▼ Solaris 7 OS サービスを OS サーバーに追加する方法

注 - 今回の Solaris リリースの名称は「Solaris 7」ですが、コード、パス名、パッケージパス名などで、「Solaris 2.7」または「SunOS 5.7」という名称が使用されていることがあります。コード、パス、パッケージパスなどを実際に入力または使用するときには、必ずマニュアル中に記述されている名称に従ってください。

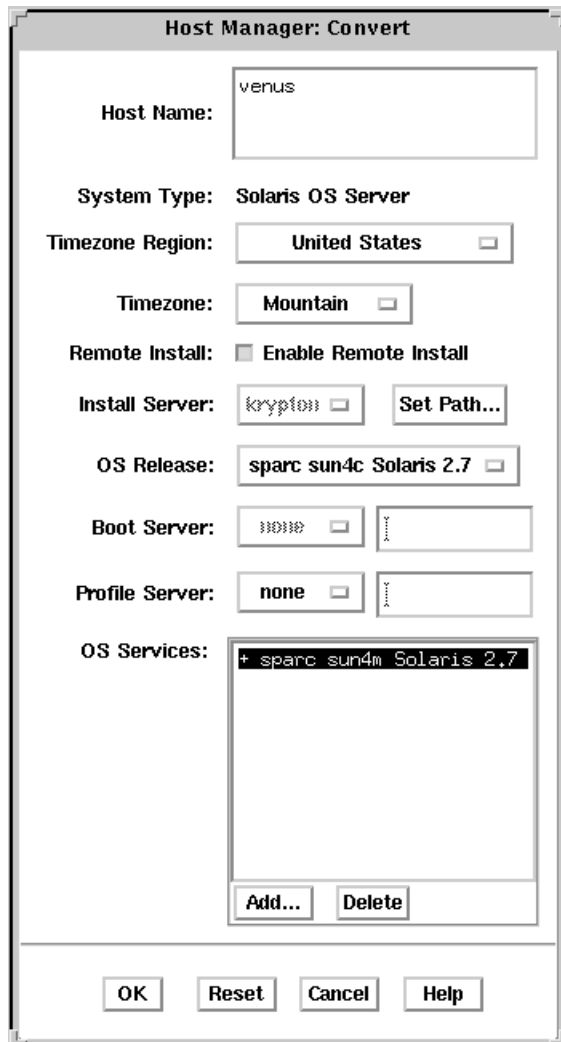
1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. ホストマネージャのメインウィンドウから変更する **OS** サーバーを選択します。
3. ホストマネージャのメインウィンドウの「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
「変更 (**Modify**)」ウィンドウが表示されます。
4. 「**OS** サービス (**OS Services**)」ウィンドウにある「追加 (**Add**)」ボタンをクリックして、サービスを追加します。
5. 「パスの設定 (**Set Path**)」をクリックして、クライアントサービスを追加するための **Solaris CD** イメージへのパスを指定します。
デフォルトのインストールサーバーは、現在のホストです。別のホストをインストールサーバーに指定するには、「インストールサーバー (**Install Server**)」メニューから「その他 (**Other**)」を選択してください。リモートシステムは、管理対象のシステムとして、最小限の設定が行われていなければなりません。

| 使用する媒体 | ボリュームマネージャによる管理 | 入力するパス |
|--|--------------------|---|
| Solaris CD イメージとして Solaris CD を使用 | Solaris CD は管理される | SPARC プラットフォーム上の /cdrom/cdrom0/s0 または /cdrom/cdrom0/s2、x86 プラットフォーム上の /cdrom/cdrom0 |
| インストールサーバーのハードディスク上の Solaris CD のコピー (setup_install_server を使用)。 | Solaris CD は管理されない | Solaris CD のマウント先 setup_install_server で指定した Solaris CD のコピー先 |

6. 追加するサービスのタイプを指定して、「追加 (**Add**)」をクリックします。
OS サービスは、OS サービスリストに追加されて、プラス記号 (+) のマークが付けられます。これは、変更を保存するときに、追加されることを意味します。「削除 (Delete)」ボタンを使用すれば、変更を保存する前に、OS サービスをリストから削除できます。
7. 「追加 (**Add**)」ウィンドウ上の「了解 (**OK**)」をクリックします。
8. 「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択して、サービスを追加します。
9. /export/Solaris_2.7 ディレクトリの内容をリストして、OS サービスディレクトリが利用できることを確認します。

例 — OS サービスの追加 (**Add OS Services**) ウィンドウの入力例

次の例では、OS サーバー venus に Solaris 2.7 SPARC sun4m サービスを追加します。



▼ AutoClient システムのサポートを追加する方法

注 - この手順は、AutoClient サーバーがすでに OS サーバーとして設定されていて、追加する AutoClient システムのカーネルアーキテクチャでインストールされていることを仮定しています。OS サーバーを追加する方法、あるいは、既存のシステムを OS サーバーに変換する方法については、オンラインヘルプ、または『*Solstice AdminSuite 2.3 管理者ガイド*』を参照してください。

1. **Solstice Launcher** からホストマネージャを起動して、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。「追加 (**Add**)」ウィンドウが表示されます。デフォルトのシステムタイプが「**Solaris** スタンドアロン (**Solaris Standalone**)」であることに注意してください。
3. 「システムタイプ (**System Type**)」メニューから「**Solstice AutoClient**」を選択します。
Solstice AutoClient システム用の「追加 (**Add**)」ウィンドウが表示されます。
4. **AutoClient** システムについてのシステム情報を入力します。
5. 必要な情報を入力した後、「**OK**」をクリックします。
Solstice AutoClient 機能のライセンスを有効にしていなかった場合、ソフトウェアがライセンスを取得できなかったことを示すメッセージが表示されます。ライセンスを有効にする方法については、『*Solstice AutoClient 2.1* ご使用にあたって』を参照してください。
AutoClient システムは、追加される AutoClient システムのリストの一部になります。そして、ホストマネージャのメインウィンドウに、プラス記号 (+) 付きで表示されます。+ は、システムが「追加を保留している」ことを意味します。
6. 手順 2 から手順 5 までを繰り返して、次の **AutoClient** システムをユーザー保留になっている変更の「バッチ」に追加します。
システムを追加するたびに、「変更の保留の合計 (Total Changes Pending)」の値が増加します。
7. ウィンドウにリストされているすべての **AutoClient** システムの追加を確認する用意ができたとき、「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択します。
「変更を保存 (Saving Changes)」メッセージウィンドウが表示されます。
「ファイル (File)」メニューから「変更を保存 (Save Changes)」を選択したとき、すべての AutoClient システムが追加されます。

各クライアントを追加するには数分必要です。この時間は、サーバーの速度、現在の負荷、および、自動的に追加されるパッチの数とタイプによって異なります。

各 AutoClient システムの追加が成功するたびに (「変更を保存 (Saving Changes)」ウィンドウに表示される)、ホストマネージャのメインウィンドウでは、その追加に対応するエントリが、「追加を保留」として表示されなくなります。つまり、ホスト名の横にあった + が消えます。



注意 - AutoClient システムが正しく動作するためには、その `/export/root` ディレクトリへのスーパーユーザー権限でのアクセスが必要です。ホストマネージャが、`/export` ディレクトリがすでに共有されていて、要求されているものとは異なる共有オプションを持っているというメッセージを表示する場合、AutoClient システムが適切に機能するためには、クライアントの `root` 領域へのスーパーユーザー権限でのアクセスを許可する必要があります。通常、クライアント `root` のアクセスモードは、`rw=clientname, root=clientname` です。

ホストマネージャが、`/usr` ディレクトリがすでに共有されているというメッセージを表示するのは、読み取り専用の `/usr` を共有しようとしたためです。このディレクトリを読み書きアクセス権で共有していた場合、`/usr` は正しく設定され、何も変更することはありません。

8. AutoClient システムをネットワークからブートします。

AutoClient システムをブートする方法については、『*Solstice AutoClient 2.1 管理者ガイド*』を参照してください。

9. 初期ブートプロセス中、プロンプトが表示されたら、AutoClient システムのシステム構成情報を提供します。

10. プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。

例 — AutoClient システム用のホストマネージャの「追加 (Add)」ウィンドウの入力例

次の例は、Solstice AutoClient システム `mars` について入力した「追加 (Add)」ウィンドウです。

Host Manager: Add

Host Name: mars

IP Address: 129.152.225.4

Ethernet Address: 8:0:20:c:28:e

System Type: Solstice AutoClient

Timezone Region: United States

Timezone: Mountain

File Server: krypton

OS Release:

Root Path: /export/root

Swap Size: 32 megabytes

Disk Config:

Disconnectable: Enable Disconnectability

Script Features: Enable Scripts ...

Root Password...

OK Apply Reset Cancel Help

▼ ディスクレスクライアントサポートを追加する方法

概要と手順は次のとおりです。

- ディスクレスクライアントに関するシステム情報を追加します。
- ディスクレスクライアントの OS サービスを選択します。
- ディスクレスクライアントをブートします。

注 - この手順では、サービスを提供するシステム (ファイルサーバー) が、すでに OS サーバーとして構成されているものと仮定しています。

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. ホストマネージャのメインウィンドウの「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。
「追加 (Add)」ウィンドウが表示されます。
3. 「**Solaris Diskless**」をシステムタイプとして選択し、システム情報を入力します。
システム情報には、「ホスト名 (Host Name)」から「時間帯 (Timezone)」まですべてのフィールドが含まれます。各フィールドについての説明は、「ヘルプ (Help)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
4. ファイルサーバーを選択します。
デフォルトのファイルサーバーは、現在のホストです。別のホストをインストールサーバーに指定するには、「インストールサーバー (Install Server)」メニューから「その他 (Other)」を選択してください。
5. 「**OS リリース (OS Release)**」メニューから、クライアントのアーキテクチャタイプと **OS** リリースを選択します。
アーキテクチャタイプはディスクレスクライアントのアーキテクチャタイプと一致していなければなりません。また、OS リリースは、ディスクレスクライアント上で実行したい Solaris のリリースと一致していなければなりません。
6. システムのルートパス、スワップパス、スワップサイズを指定します。
7. 「追加 (**Add**)」ウィンドウ上の「了解 (**OK**)」をクリックします。
8. 「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択して、ディスクレスクライアントのサポートを追加します。
ホストマネージャのメインウィンドウ内に、ディスクレスクライアントが表示されます。システムルートとスワップ領域の作成、および `admclientpatch` コマンドによるパッチの適用には時間がかかるので、ディスクレスクライアントサポートを追加するには数分かかります。



注意 - ディスクレスクライアントが正しく動作するためには、その `/export/root` ディレクトリへのスーパーユーザー権限でのアクセスが必要です。ホストマネージャが、`/export` ディレクトリがすでに共有されていて、要求されているのとは異なる共有オプションを持っているというメッセージを表示する場合、ディスクレスシステムが適切に機能するためには、クライアントのルート領域へのスーパーユーザー権限でのアクセスを許可する必要があります。通常、クライアントルートのアクセスモードは、`rw=clientname, root=clientname` です。

ホストマネージャが、`/usr` ディレクトリがすでに共有されているというメッセージを表示するのは、読み取り専用の `/usr` を共有しようとしたからです。このディレクトリを読み書きアクセス権で共有していた場合、`/usr` は正しく設定され、何も変更することはありません。

9. ホストマネージャのメインウィンドウで、システムがディスクレスクライアントとして追加されていることを確認します。
10. ディスクレスクライアントをブートします。
11. 最初のブートプロセス時に、ディスクレスクライアント用に、次のシステム設定情報をプロンプトに従って入力します。
 - 地理的位置
 - 時間帯
 - 日付と時刻
12. プロンプトに従って、スーパーユーザーのパスワードを入力します。

例 — ディスクレスクライアント追加ウィンドウの入力

次の例は、ディスクレスクライアント `neptune` について「追加 (Add)」ウィンドウに入力します。

Host Manager: Add

Host Name: neptun³

IP Address: 129.152.225.1¹

Ethernet Address: 8:0:20:6c:d8:6²

System Type: Solaris Diskless

Timezone Region: United States

Timezone: Mountain

File Server: krypton

OS Release:

Root Path: /export/root

Swap Path: /export/swap

Swap Size: 32 megabytes

Terminal Type: sun

Root Password...

OK Apply Reset Cancel Help

サーバーとクライアントサポートの管理

表 4-2 作業マップ: サーバーとクライアントサポートの管理

| 作業 | 説明 | 手順の説明 To |
|----------------|--|--------------------------|
| 1. システムサポートの変更 | ホストマネージャの「編集 (Edit)」メニューから「変更 (Modify)」を選択して、システムサポートを変更します。 | 103ページの「システムサポートを変更する方法」 |
| 2. システムサポートの削除 | ホストマネージャの「編集 (Edit)」メニューから「削除 (Delete)」を選択して、システムサポートを削除します。 | 104ページの「システムサポートを削除する方法」 |

▼ システムサポートを変更する方法

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. ホストマネージャのメインウィンドウから、変更するシステムエントリを選択します。
3. 「編集 (**Edit**)」メニューから「変更 (**Modify**)」を選択します。
選択したシステムエントリが「変更 (**Modify**)」ウィンドウに表示されます。
4. システムサポートを変更します。
各フィールドについての説明は、「ヘルプ (**Help**)」ボタンをクリックして、このウィンドウのフィールド定義を表示してください。
5. 「変更 (**Modify**)」ウィンドウ上で「了解 (**OK**)」をクリックします。
6. 「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択して、システムサポートを変更します。
7. ホストマネージャのメインウィンドウで、システムエントリが変更されていることを確認します。

▼ システムサポートを削除する方法

1. **Solstice Launcher** からホストマネージャを起動し、ネームサービスを選択します。
詳細は 81ページの「Solstice ホストマネージャを起動する方法」を参照してください。
2. ホストマネージャのメインウィンドウから、削除するシステムエントリを選択します。
3. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
削除してよいかをたずねるウィンドウが表示されます。
4. 「了解 (**OK**)」をクリックします。
5. 「ファイル (**File**)」メニューから「変更を保存 (**Save Changes**)」を選択して、システムサポートを削除します。
ホストマネージャのメインウィンドウから、選択したシステムエントリが削除されます。
6. システムエントリが、ホストマネージャのメインウィンドウから削除されていることを確認します。

ホストマネージャのコマンド行インタフェースを使用して設定作業を自動化する

ホストマネージャの機能に相当するコマンド行を使用すると、新しいディスクレスシステムや AutoClient システムの作成に関連する、多くの設定作業を自動化できます。この自動化は、JumpStart を使用して Solaris をスタンドアロンシステムにインストールするときに行うことと似ています。ユーザー独自のシェルスクリプトを書いて、コマンド行機能を使用すると、1回の操作でクライアント環境を自動的にカスタマイズできます。

表 4-3 に、Solstice AdminSuite の代わりに使用して管理作業を実行できる、ホストマネージャのコマンドを要約します。

表 4-3 Solstice AdminSuite コマンド行相当機能の要約

| コマンド | 目的 | /opt/SUNWadm/2.x/man/ の 参照先 |
|--|--------------------------------------|--------------------------------|
| <code>admhostmod -x type=type host</code> | generic のマークが付いているシステムタイプを更新する。 | admhostmod |
| <code>admhostadd -D -x [file serv=server] [-x root=directory] [-x swap=swap_file] [-x swappsize=size] [-x os=version] [-x ns=NIS+ NIS NONE] [-x domain=domain rhost=host]</code> | クライアントサポートを追加するためのデフォルトを設定する。 | admhostadd |
| <code>admhostadd -i ip_address -e ethernet_addr [-x type= type] [-x tz=timezone] [-x install=Y N] [-x installpath=server:/path] host</code> | システムを追加して、ネットワークインストールを有効にする (省略可能)。 | admhostadd |
| <code>admhostmod -x type=OS_SERVER host</code> | スタンドアロンシステムを OS サーバーに変換する。 | admhostmod |
| <code>admhostmod -x mediapath=server:/path -x platform=platform host</code> | OS サービスを OS サーバーに追加する。 | admhostmod |
| <code>admhostdel [-x ns=NIS+ NIS NONE] [-x domain=domain rhost=host ...]host ...</code> | 既存のシステムまたは OS サーバーを削除する。 | admhostdel |

パート III システムのシャットダウンとブート

このパートでは、Solaris システムのシャットダウンとブートの手順について説明します。次の章で構成されています。

| | |
|--------|--|
| 第 5 章 | システムのシャットダウンとブートの概要とガイドラインを示します。 |
| 第 6 章 | 実行レベルとブートファイルについて説明します。 |
| 第 7 章 | システムのシャットダウン手順について説明します。 |
| 第 8 章 | SPARC システムのブート手順について説明します。 |
| 第 9 章 | x86 システムのブート手順について説明します。 |
| 第 10 章 | ブートプロセスの概要を説明します。SPARC システムおよび x86 システムのブートに使用するプラットフォーム固有のハードウェアについても説明します。 |



システムのシャットダウンとブートの概要

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

この章の内容は次のとおりです。

- 109ページの「システムのシャットダウンとブートに関する新機能」
- 110ページの「シャットダウンとブートについての参照先」
- 110ページの「用語」
- 111ページの「システムのシャットダウンに関するガイドライン」
- 112ページの「システムのブートに関するガイドライン」
- 112ページの「再構成用ブートの実行」
- 113ページの「システムをシャットダウンする場合」
- 114ページの「システムをブートする場合」

システムのシャットダウンとブートに関する新機能

この節では、Solaris 7 におけるシステムのシャットダウンとブートに関する新機能について説明します。

レベル S (シングルユーザーモード) でのシステムの起動

バグ ID 1154696 は Solaris 7 で修正されました。つまり、`shutdown -s` または `init -s` コマンドで、システムを正常にシャットダウンし、レベル S (シングルユーザーモード) に移行できるようになりました。 `inittab` ファイルと、`/etc/init.d` および `/etc/rcn.d` ディレクトリ内の `rc` スクリプトが修正され、システム起動レベルの変更が適切かつ効率的に行われるようになりました。

32 ビットまたは 64 ビット Solaris 7 オペレーティング環境のシステムのブート

32 ビットまたは 64 ビットの Solaris 7 オペレーティング環境のシステムをブートする方法については、690 ページの「64 ビット Solaris ブート時の問題の解決」を参照してください。

シャットダウンとブートについての参照先

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

- 第 7 章
- 第 8 章
- 第 9 章

用語

ここではシャットダウンとブートに関する用語について説明します。

- 実行レベルと `init` 状態 – 「実行レベル」とは、システムの状態を表す文字または数字のことで、どのシステムサービスを使用できるのかを示します。システムは常に定義済み実行レベルの 1 つで動作します。実行レベルを変更するために `init` プロセスが使用されるため、実行レベルは「`init` 状態」と呼ばれることもあります。システム管理者は `init(1M)` コマンドを使用して、実行レベルを変更します。

- ブートタイプ - 「ブートタイプ」とはシステムのブート方法を指すもので、次のようなブートタイプがあります。
- 対話式ブート - システムのブート方法に関する情報 (カーネルやデバイスのパス名など) を入力するプロンプトが表示されます。
- 再構成用ブート - システムが再構成され、新しく追加したハードウェアや新しい擬似デバイスがサポートされます。
- 回復ブート - システムがハング状態になったとき、無効なエントリがあるためシステムが正常にブートできないとき、またはユーザーがログインできないときに使用します。

システムのシャットダウンに関するガイドライン

システムをシャットダウンするときは次の点に注意してください。

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

- 実行レベル 0 - オペレーティングシステムがシャットダウンされ、安全に電源切断できる状態。システムの設置場所を変更したり、ハードウェアを追加または削除する場合は、システムを実行レベル 0 にする必要があります。

実行レベルについての詳細は、第 6 章を参照してください。

システムのブートに関するガイドライン

システムをブートするときは、次の点に注意してください。

- シャットダウン後にシステムをブートするには、SPARC システムの場合は、PROM レベルで boot コマンドを使用します。x86 システムの場合は、一次ブートサブシステムメニューで boot コマンドを使用します。
- 電源を切断したあと再投入すればシステムをリブートできます。ただし、この方法では、システムサービスやプロセスが突然終了してしまうので、適切なシャットダウンとは言えません。緊急時のリブート以外には使用しないようにします。
- SPARC および x86 システムでは、ブート時に使用するハードウェアが異なります。これらのハードウェアの違いについては、第 10 章を参照してください。

再構成用ブートの実行

再構成用ブートは、システムに新しいハードウェアを追加したときや擬似デバイスのサポートを再構成したとき (pty 数の増加など) に行います。表 5-1 に、どの場合にどの再構成手順を使用するかを示します。

表 5-1 再構成手順

| システム再構成の内容 | 参照先 |
|------------|--|
| ディスクの追加 | 第 23 章または第 24 章 |
| 周辺装置の追加 | 316ページの「周辺デバイスを追加する方法」 |
| 擬似デバイス数の変更 | 『Solaris のシステム管理 (第 2 巻)』の「システム情報の確認と変更」 |

システムをシャットダウンする場合

表 5-2 に、システム管理作業とそれに伴って必要となるシャットダウンの種類を示します。

表 5-2 システムのシャットダウン

| 管理作業 | 以下の実行レベルに変更 | 参照先 |
|--------------------------------------|---------------------------|--|
| 停電のためシステムの電源を切断する。 | 実行レベル 0。安全に電源を切れる状態。 | 第 7 章 |
| /etc/system ファイル内のカーネルパラメータを変更する。 | 実行レベル 6 (システムのリブート)。 | 第 7 章 |
| ファイルシステムを保守する (システムデータのバックアップや復元など)。 | 実行レベル S (シングルユーザーモード)。 | 第 7 章 |
| /etc/system などのシステム構成ファイルを修正する。 | 114ページの「システムをブートする場合」を参照。 | なし |
| /etc/system ファイル内の擬似デバイスパラメータを変更する。 | 再構成用ブート。 | 『Solaris のシステム管理 (第 2 巻)』の「カーネルパラメータの調整手順」 |

表 5-2 システムのシャットダウン 続く

| 管理作業 | 以下の実行レベルに変更 | 参照先 |
|---|-------------------------------------|------------------------------------|
| システムにハードウェアを追加する (または、システムからハードウェアを削除する)。 | 再構成用ブート (ハードウェアを追加または削除したら電源を切断する)。 | 384ページの「SPARC: 二次ディスクを接続してブートする方法」 |
| ブート失敗の原因となっている重要なシステムファイルを修正する。 | 114ページの「システムをブートする場合」を参照。 | なし |
| カーネルデバッガ (kadb) をブートして、システムの障害を調査する。 | 実行レベル 0 (可能な場合)。 | 第 7 章 |
| ハング状態から回復する。または、クラッシュダンプを強制する。 | 114ページの「システムをブートする場合」を参照。 | なし |

サーバーまたはスタンドアロンシステムのシャットダウンの例については、第 7 章を参照してください。

システムをブートする場合

表 5-3 に、システム管理作業とそれに伴って必要となるブートの種類を示します。

表 5-3 システムのブート

| リポート前に実行した管理作業 | 使用するブート種類 | SPARC のブート手順の参照先 | x86 のブート手順の参照先 |
|-----------------------------------|--|--|--|
| 停電のためシステムの電源を切断した。 | システムの電源を再投入する。 | 第 7 章 | 第 7 章 |
| /etc/system ファイル内のカーネルパラメータを変更した。 | システムを実行レベル 3 にリポートする (NFS 資源を共有できる状態のマルチユーザーモード) | 148ページの「SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法」 | 161ページの「x86: システムを実行レベル 3 (マルチユーザー状態) にする方法」 |

表 5-3 システムのブート 続く

| リポート前に実行した管理作業 | 使用するブート種類 | SPARC のブート手順の参照先 | x86 のブート手順の参照先 |
|---|---|--|--|
| ファイルシステムを保守した (システムデータのバックアップや復元など)。 | 実行レベル S で Ctrl-d を押して、システムを実行レベル 3 に戻す。 | 149ページの「SPARC: システムを実行レベル S (シングルユーザー状態) にする方法」 | 162ページの「x86: システムを実行レベル S (シングルユーザー状態) にする方法」 |
| /etc/system などのシステム構成ファイルを修正した。 | 対話式ブート。 | 150ページの「SPARC: システムを対話式でブートする方法」 | 163ページの「x86: システムを対話式でブートする方法」 |
| /etc/system ファイル内の擬似デバイスパラメータを変更した。 | 再構成用ブート。 | 『Solaris のシステム管理 (第 2 巻)』の「カーネルパラメータの調整手順」 | 『Solaris のシステム管理 (第 2 巻)』の「カーネルパラメータの調整手順」 |
| システムにハードウェアを追加した (または、システムからハードウェアを削除した)。 | 再構成用ブート (ハードウェアを追加または削除したら電源を投入する)。 | 384ページの「SPARC: 二次ディスクを接続してブートする方法」 | 第 24 章 |
| カーネルデバッグ (kadb) をブートして、システムの障害を調査した。 | kabd をブートする。 | 156ページの「SPARC: カーネルデバッグ (kadb) を使ってシステムをブートする方法」 | 170ページの「x86: クラッシュダンプを強制して、システムをリブートする方法」 |
| ブート失敗の原因となっていた重要なシステムファイルを修正した。 | 回復ブート。 | 165ページの「x86: システムを復元するためにブートする方法」 | 165ページの「x86: システムを復元するためにブートする方法」 |
| ハング状態から回復した。または、クラッシュダンプを強制した。 | 回復ブート。 | 170ページの「x86: クラッシュダンプを強制して、システムをリブートする方法」の例を参照 | 170ページの「x86: クラッシュダンプを強制して、システムをリブートする方法」の例を参照 |

システムブートの例については、第 8 章または第 9 章を参照してください。

実行レベルとブートファイルの手順

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

この章で説明する手順は次のとおりです。

- 119ページの「システムの実行レベルを確認する方法」
- 124ページの「実行制御スクリプトを使用してサービスを起動または停止する方法」
- 125ページの「実行制御スクリプトを追加する方法」
- 126ページの「実行制御スクリプトを無効にする方法」

この章で説明する主な内容は次のとおりです。

- 117ページの「実行レベル」
- 119ページの「/etc/inittab ファイル」
- 123ページの「実行制御スクリプト」
- 127ページの「実行制御スクリプトのまとめ」

実行レベル

システムの「実行レベル」(init 状態ともいう)は、ユーザーが使用できるサービスと資源を定義します。システムが1度に持つことのできる実行レベルは1つだけです。

Solaris ソフトウェア環境には 8 つの実行レベルがあります (表 6-1 参照)。デフォルトの実行レベル 3 は、`/etc/inittab` ファイルに指定されています。

表 6-1 Solaris 実行レベル

| 実行レベル | init 状態 | 種類 | 目的 |
|---------|--------------------|----------|--|
| 0 | 電源切断状態 | 電源切断 | オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにする。 |
| s または S | シングルユーザー状態 | シングルユーザー | すべてのファイルシステムがマウントされ使用可能な状態で、シングルユーザーとして動作する。 |
| 1 | システム管理状態 | シングルユーザー | ユーザーがログインできる状態ですべての使用可能なファイルシステムにアクセスする。 |
| 2 | マルチユーザー状態 | マルチユーザー | 通常の運用に使用する。複数のユーザーがシステムと全ファイルシステムにアクセスできる。NFS サーバーデーモンを除く、すべてのデーモンが動作する。 |
| 3 | NFS 資源を共有したマルチユーザー | マルチユーザー | 通常の運用に使用する。NFS の資源共有が可能な状態。 |
| 4 | マルチユーザー状態 (予備) | | このレベルは現在使用されていない。 |
| 5 | 電源切断状態 | 電源切断 | オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにする。可能であれば、この機能をサポートしているシステムでは電源を自動的に切断する。 |
| 6 | リブート状態 | リブート | システムをシャットダウンして実行レベル 0 にしたあと、マルチユーザー状態 (または、 <code>inittab</code> ファイルに指定されたデフォルトのレベル) でリブートする。 |

▼ システムの実行レベルを確認する方法

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

```
$ who -r
```

例—システムの実行レベルを確認する

```
$ who -r
.      run-level 3  Oct 26 15:04      3      0  S
$
```

| | |
|--------------|---------------------------|
| 実行レベル 3 | 現在の実行レベル。 |
| Oct 26 15:04 | 実行レベルが最後に変更された日時。 |
| 3 | 現在の実行レベル。 |
| 0 | 最後にリブートしてからこの実行レベルになった回数。 |
| S | 以前の実行レベル。 |

/etc/inittab ファイル

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

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

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

id: rstate: action: process

表 6-2 に、`inittab` エントリの各フィールドを要約します。

表 6-2 `inittab` ファイルのフィールド

| フィールド | 説明 |
|----------------|---|
| <i>id</i> | エントリに固有の (一意の) 識別子。 |
| <i>rstate</i> | このエントリが適用される実行レベルのリスト。 |
| <i>action</i> | プロセスフィールドに指定されたプロセスの実行方法。指定できる値は、 <code>initdefault</code> 、 <code>sysinit</code> 、 <code>boot</code> 、 <code>bootwait</code> 、 <code>wait</code> 、および <code>respawn</code> 。 |
| <i>process</i> | 実行するコマンド。 |

例 — デフォルトの `inittab` ファイル

以下の例は、デフォルトの `inittab` ファイルです。

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit >/dev/console 2<>/dev/console </dev/
console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/console 2<>/dev/
console
sS:s:wait:/sbin/rcS >/dev/console 2<>/dev/console </dev/
console
s0:0:wait:/sbin/rc0 >/dev/console 2<>/dev/console </dev/
console
s1:1:respawn:/sbin/rc1 >/dev/console 2<>/dev/console </dev/
console
s2:23:wait:/sbin/rc2 >/dev/console 2<>/dev/console </dev/
console
s3:3:wait:/sbin/rc3 >/dev/console 2<>/dev/console </dev/
console
s5:5:wait:/sbin/rc5 >/dev/console 2<>/dev/console </dev/
console
s6:6:wait:/sbin/rc6 >/dev/console 2<>/dev/console </dev/
console
fw:0:wait:/sbin/uadmin 2 0 >/dev/console 2<>/dev/console </dev/
console
```

(続く)

```
of:5:wait:/sbin/uadmin 2 6      >/dev/console 2<>/dev/console </dev/  
console  
rb:6:wait:/sbin/uadmin 2 1      >/dev/console 2<>/dev/console </dev/  
console  
sc:234:respawn:/usr/lib/saf/sac -t 300  
co:234:respawn:/usr/lib/saf/  
ttymon -g -h -p "`uname -n` console login: " -T sun  
-d /dev/console -l console -m ldterm,ttcompat
```

システムが実行レベル 3 になると実行される処理

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

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

表 6-3 に、実行レベル 3 の `action` フィールドで使用するキーワードについて説明します。

表 6-3 実行レベル 3 の action キーワードの説明

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

表 6-4 に、実行レベル 3 で実行されるプロセス (またはコマンド) について説明します。

表 6-4 実行レベル 3 のコマンドの説明

| コマンドまたはスクリプト名 | 説明 |
|--|--|
| <code>/usr/sbin/shutdown</code> | システムをシャットダウンする。 <code>init</code> プロセスは、システムが <code>powerfail</code> シグナルを受信した場合にのみ <code>shutdown</code> コマンドを実行する。 |
| <code>/sbin/rcS</code> | ルート (<code>/</code>)、 <code>/usr</code> 、 <code>/var</code> 、 <code>/var/adm</code> のファイルシステムをマウントしてチェックする。 |
| <code>/sbin/rc2</code> | 標準のシステムプロセスを起動して、システムを実行レベル 2 (マルチユーザーモード) に移行する。 |
| <code>/sbin/rc3</code> | 実行レベル 3 で使用される NFS 資源共有を開始する。 |
| <code>/usr/lib/saf/sac -t 30</code> | ポートモニターと UUCP 用のネットワークアクセスを起動する。このプロセスは失敗すると再起動される。 |
| <code>/usr/lib/saf/ttymon -g -h -p ``uname -n` console login: " -T terminal_type -d /dev/console -l console</code> | コンソールでのログイン要求を監視する <code>ttymon</code> プロセスを起動する。 このプロセスは失敗すると再起動される。SPARC システムの <code>terminal_type</code> は <code>sun</code> である。 x86 システムの <code>terminal_type</code> は <code>AT386</code> である。 |

実行制御スクリプト

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

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

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

```
# ls /etc/rc2.d
K20spc@           S70uucp*         S80lp*
K60nfs.server*   S71rpc*          S80spc@
K76snmpdx*       S71sysid.sys*   S85power*
K77dmi*          S72autoinstall* S88sendmail*
README           S72inetsvc*     S88utmpd*
S01MOUNTFSYS*   S73nfs.client*  S89bdconfig@
S05RMTMPFILES*  S74autofs*      S91leoconfig*
S20syssetup*    S74syslog*      S92rtvc-config*
S21perf*        S74xntpd*       S92volmgt*
S30sysid.net*   S75cron*        S93cacheos.finish*
S47asppp*       S76nscd*        S99audit*
S69inet*        S80PRESERVE*    S99dtlogin*
```

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

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

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

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

各実行制御スクリプトの処理について、表 6-5 から表 6-10 に要約を示します。

実行制御スクリプトを使用してサービスを起動または停止する

実行レベルごとに対応するスクリプトを持つことの利点は、`/etc/init.d` ディレクトリ内の個々のスクリプトを実行することにより、システムの実行レベルを変更しないで (現在の実行レベルの) 機能を停止できる点です。

▼ 実行制御スクリプトを使用してサービスを起動または停止する方法

1. スーパーユーザーになります。
2. 機能を停止します。

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

3. 機能を再開します。

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

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

```
# pgrep -f service
```

例—実行制御スクリプトを使用してサービスを起動または停止する

NFS サーバーの機能を停止するには、次のように入力します。

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

NFS サービスを再開するには、次のように入力します。

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

実行制御スクリプトの追加

サービスを起動または停止するための実行制御スクリプトを追加するには、そのスクリプトを `/etc/init.d` ディレクトリにコピーして、`rcn.d` ディレクトリ内の適切なファイルへのリンクを作成します。

実行制御スクリプトの命名法についての詳細は、`/etc/rcn.d` ディレクトリ内の `README` ファイルを参照してください。以下に、実行制御スクリプトの追加方法を説明します。

▼ 実行制御スクリプトを追加する方法

1. スーパーユーザーになります。
2. スクリプトを `/etc/init.d` ディレクトリにコピーします。

```
# cp filename /etc/init.d
# chmod 0744 /etc/init.d/filename
# chown root:sys /etc/init.d/filename
```

3. 適切な `rcn.d` ディレクトリへのリンクを作成します。

```
# cd /etc/init.d
# ln filename /etc/rc2.d/Snnfilename
# ln filename /etc/rcn.d/Knnfilename
```

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

```
# ls /etc/init.d/ /etc/rc2.d/ /etc/rcn.d/
```

例—実行制御スクリプトを追加する

```
# cp xyz /etc/init.d
# cd /etc/init.d
# ln xyz /etc/rc2.d/S100xyz
# ln xyz /etc/rc0.d/K100xyz
# ls /etc/init.d /etc/rc2.d /etc/rc0.d
```

実行制御スクリプトを無効にする

実行制御スクリプトを無効にするには、スクリプト名の先頭にドット (.) を付けます。ドットで始まるファイルは実行されません。接尾辞を追加してファイルをコピーすると、両方のファイルが実行されます。

▼ 実行制御スクリプトを無効にする方法

1. スーパーユーザーになります。
2. スクリプト名の先頭に下線 (`_`) を付けて、スクリプト名を変更します。


```
# cd /etc/rcn.d
# mv filename _filename
```

3. スクリプト名が変更されたことを確認します。

```
# ls -a
_filename
```

例—実行制御スクリプトを無効にする

次の例は、S100datainit というスクリプト名を変更しますが、元のスクリプトも保存しています。

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

実行制御スクリプトのまとめ

表 6-5 /sbin/rc0 スクリプト

| スクリプト名 | 説明 |
|-----------|--|
| /sbin/rc0 | 以下の作業を実行する。 <ul style="list-style-type: none">■ システムサービスとデーモンの終了■ 実行中の全プロセスの停止■ 全ファイルシステムのマウント解除 |

表 6-6 /sbin/rc1 スクリプト

| スクリプト名 | 説明 |
|-----------|---|
| /sbin/rc1 | <p>/etc/rc1.d ディレクトリ内のスクリプトを実行して以下の作業を実行する。</p> <ul style="list-style-type: none"> ■ システムサービスとデーモンの終了 ■ 実行中の全プロセスの停止 ■ 全ファイルシステムのマウント解除 ■ システムをシングルユーザーモードに移行 |

表 6-7 /sbin/rc2 スクリプト

| スクリプト名 | 説明 |
|-----------|--|
| /sbin/rc2 | <p>/etc/rc2.d ディレクトリ内のスクリプトを実行して以下の作業を実行する。</p> <ul style="list-style-type: none"> ■ すべてのローカルファイルシステムのマウント ■ quota オプションを指定してマウントされたファイルシステムが 1 つでも存在する場合は、ディスク割り当てを有効にする。 ■ エディタの一時ファイルを /usr/preserve 内に保存する。 ■ /tmp ディレクトリにあるすべてのファイルの削除 ■ システムアカウントの設定 ■ デフォルトのルータの設定 ■ NIS ドメインと ifconfig ネットワークマスクの設定 ■ インストール媒体またブートサーバー (/ .PREINSTALL または / AUTOINSTALL のどちらかが存在する場合) からシステムをリブートする。 ■ 必要に応じて、inetd、rpcbind、named を起動する。 ■ Kerberos のクライアント側デーモン kerbd を起動する。 ■ NIS デーモン (ypbind) および NIS+ デーモン (rpc.nisd) を起動する。どちらを起動するかは、システムが NIS または NIS+ のどちら用に設定されているか、またシステムがクライアントとサーバーのどちらであるかによって異なる。 ■ keyserv、statd、lockd、xntpd、utmpd の起動 ■ すべての NFS エントリのマウント ■ ncsd (ネームサービスキャッシュデーモン) の起動 ■ automount、cron、LP 印刷サービス、sendmail、utmpd、vold の各デーモンの起動 |

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

表 6-8 /sbin/rc3 スクリプト

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

表 6-9 /sbin/rc5 スクリプトと /sbin/rc6 スクリプト

| スクリプト名 | 説明 |
|-------------------------|--|
| /sbin/rc5 および /sbin/rc6 | <p>/etc/rc0.d/k* ディレクトリ内のスクリプトを実行して以下の作業を行う。</p> <ul style="list-style-type: none"> ■ すべてのアクティブなプロセスを停止する。 ■ ファイルシステムのマウント解除 |

表 6-10 /sbin/rcS スクリプト

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

システムのシャットダウンの手順

この章では、システムのシャットダウン手順について説明します。この章で説明する手順は次のとおりです。

- 134ページの「システムにログインしているユーザーを知る方法」
- 134ページの「サーバーをシャットダウンする方法」
- 138ページの「スタンドアロンシステムをシャットダウンする方法」
- 140ページの「すべてのデバイスの電源を落とす方法」

この章で説明する主な内容は次のとおりです。

- 131ページの「システムをシャットダウンする場合」
- 132ページの「システムをシャットダウンするには」
- 133ページの「デバイスの電源を落とす場合」
- 133ページの「システムのダウンについてユーザーに通知する」

使用可能な実行レベルについての概要は、第 6 章を参照してください。

システムをシャットダウンする場合

Solaris システムソフトウェアは、電子メールとネットワーク資源をいつでも利用できるように停止することなく動作するように設計されています。しかし、システム管理作業を行う場合や緊急事態が発生した場合は、システムをシャットダウンして安全に電源を切断できる状態にするか、一部のシステムサービスしか提供しない中間の実行レベルまで移行する必要があります。たとえば次のような場合です。

- ハードウェアを追加または削除する。
- 予告済みの停電に備える。
- ファイルシステムの保守を行う (バックアップなど)。

システムをシャットダウンする必要があるシステム管理作業についての詳細は、第 5 章を参照してください。

システムをシャットダウンするには

システムをシャットダウンする第 1 の方法は、`init` コマンドおよび `shutdown` コマンドを使用する方法です。どちらのコマンドもシステムを「正常に シャットダウン」します。つまり、すべてのファイルシステムに対する変更はディスクに書き出され、すべてのシステムサービス、プロセス、オペレーティングシステムが正常に終了します。

システムのアポルトキーシーケンスを使用したり、電源をオフにしてからオンにする方法では、システムサービスが突然終了してしまうので、正常なシャットダウン方法とはいえません。しかし、緊急時には、これらの方法を使用しなければならない場合もあります。システムの復元手順については、第 8 章または第 9 章を参照してください。

表 7-1 に、いくつかのシャットダウンコマンドとその用途を要約します。

表 7-1 シャットダウンコマンド

| コマンド | 説明 | 用途 |
|-----------------------|---|---|
| <code>shutdown</code> | <code>init</code> を呼び出してシステムをシャットダウンする実行可能なシェルスクリプト。デフォルトでは、システムは実行レベル S に移行する。 | 実行レベル 3 で動作しているサーバーで使用する。サーバーにログインしているユーザー、およびサーバー資源をマウントしているシステムに、サーバーが間もなくシャットダウンされることが通知される。 |
| <code>init</code> | すべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能ファイル。 | 他のユーザーが影響を受けないスタンドアロンシステムで使用する。ユーザーはまもなく行われるシャットダウンについて通知されないで、シャットダウンにかかる時間は短くて済む。 |

表 7-1 シャットダウンコマンド 続く

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

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

デバイスの電源を落とす場合

次のような場合は、すべてのシステムデバイスの電源を落とす必要があります。

- ハードウェアを置換または追加する。
- システムの設置場所を変更する。
- 予測可能な停電や自然災害 (接近中の雷雨など) に備える。

システムデバイスとは、CPU、モニター、外部デバイス (ディスク、テープ、プリンタ) などを意味します。

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

システムのダウンについてユーザーに通知する

`shutdown` コマンドは起動時に、ログインしているすべてのユーザーおよびシステム資源をマウントしているすべてのシステムに、警告と最終メッセージという形でシャットダウンを通知します。

サーバーをシャットダウンする場合に、`init` コマンドではなく `shutdown` コマンドを使用することを推奨するのはこのためです。どちらを使用するにしても、ユー

ザーには予定されているシャットダウンについてあらかじめ電子メールで知らせておくようにしてください。

システム上のどのユーザーに通知する必要があるかを知るには、who コマンドを使用します。who コマンドは、システムの現在の実行レベルを知りたい場合にも使用できます。システム上のどのユーザーに通知する必要があるかを知るには、who コマンドを使用します。who(1) コマンドは、119ページの「システムの実行レベルを確認する方法」で説明されているシステムの現在の実行レベルを知りたい場合にも使用できます。

▼ システムにログインしているユーザーを知る方法

1. シャットダウンするシステムにログインします。
2. who コマンドを使用して、ログイン中のユーザーを表示します。

```
$ who
```

例—システムにログインしているユーザーを知る

who コマンドの出力例を以下に示します。

```
$ who
1 holly2          console May  7 07:30
3 kryten         pts/0    May  7 07:35 (starbug)
4 lister         pts/1    May  7 07:40 (bluemidget)
```

▼ サーバーをシャットダウンする方法

1. スーパーユーザーになります。
2. システムにユーザーがログインしているかどうか調べます。

```
# who
```

1. ログインしているユーザーのユーザー名。
2. ログインしているユーザーの端末回線。
3. ユーザーがログインした日時。
4. (省略可能) リモートシステムからログインしているユーザーのホスト名。

ログインしているすべてのユーザーが表示されます。システムがシャットダウンされることを、メールかブロードキャストメッセージで知らせることもできます。

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

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

| | |
|-----------------------------|--|
| <code>-iinit-state</code> | システムをデフォルトの S 以外の init 状態にする。0、1、2、5、6 のいずれかを指定できる。 |
| <code>-ggrace-period</code> | シャットダウンするまでの時間 (秒) を指定する。デフォルトは 60 秒。 |
| <code>-y</code> | ユーザーの介入なしにシャットダウンを継続する。このオプションを指定しないと、シャットダウンを継続するかどうか 60 秒後にたずねられる。 |

4. シャットダウンを継続するかどうかたずねられたら、y を入力します。

```
Do you want to continue? (y or n): y
```

-y オプションを指定した場合、このプロンプトは表示されません。

5. プロンプトが表示されたら、スーパーユーザー のパスワードを入力します。

```
Type Ctrl-d to proceed with normal startup,  
(or give root password for system maintenance): xxx
```

6. システム管理作業を終了したら、**Ctrl-d** を押してデフォルトの実行レベルに戻ります。
7. システムが、shutdown コマンドで指定した実行レベルに移行したことを確認する方法を以下の表に要約します。

| 移行後の実行レベル | SPARC システムの場合 | x86 システムの場合 |
|---------------------------------|-------------------------|--------------------------|
| 実行レベル S (シングルユーザーモード) | # | # |
| 実行レベル 0 (電源切断状態) | ok または > | type any key to continue |
| 実行レベル 3 (リモート資源が共有されたマルチユーザー状態) | hostname console login: | hostname console login: |

例 — SPARC システムを実行レベル S にする (サーバー)

次の例では、shutdown コマンドを使用して、3 分後に、SPARC システムを実行レベル S (シングルユーザーモード) にしています。

```
# who
root      console      Jun 10 14:15
# shutdown -g180 -y

Shutdown started.      Wed Jun 10 14:15:25 MDT 1998

Broadcast Message from root (console) on mars Wed Jun 10 14:15:26...
The system mars will be shut down in 3 minutes
.
.
.
Broadcast Message from root (console) on mars Wed Jun 10 14:17:58...
The system mars will be shut down in 30 seconds
.
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Print services stopped.
syslogd: going down on signal 15
Killing user processes: done.

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode ...
#
```

例 — SPARC システムを実行レベル 0 にする

次の例では、shutdown コマンドを使用して、SPARC システムを 5 分後に実行レベル 0 にしています。確認用プロンプトが表示されないように `-y` オプションを指定しています。

```
# who
kryten      console      Jun 10 14:22
rimmer     pts/1        Jun 10 14:23      (starbug)
pmorph     pts/2        Jun 10 14:24      (bluemidget)
(ログインしているユーザーにメールを送る)
# shutdown -i0 -g300 -y
Shutdown started.      Wed Jun 10 14:30:32 MDT 1998
Broadcast Message from root (console) on pluto Wed Jun 10 14:30:32...
The system will be shut down in 5 minutes
.
.
.
INIT: New run level: 0
The system is coming down.  Please wait.
.
.
.
The system is down.
syncing file systems... [11] [9] [5] done
Program terminated
Type help for more information
ok
```

システムを実行レベル 0 にしてすべてのデバイスの電源を落とす場合は、140ページの「すべてのデバイスの電源を落とす方法」を参照してください。

例 — SPARC システムをリブートして実行レベル 3 にする (サーバー)

次の例では、shutdown コマンドを使用して SPARC システムをリブートし、2 分後に実行レベル 3 にしています。確認用プロンプトが表示されないように `-y` オプションを指定しています。

```
# who
kryten      console      Jun 10 14:35
rimmer     pts/1        Jun 10 14:40      (starbug)
pmorph     pts/2        Jun 10 14:45      (bluemidget)
(ログインしているユーザーにメールを送る)
# shutdown -i6 -g120 -y
Shutdown started.      Wed Jun 10 14:34:26 MDT 1998
Broadcast Message from root (console) on pluto Wed Jun 10 14:34:26 MDT 1998
```

(続く)

```
The system will be shut down in 1 minute
Changing to init state 6 - please wait
#
INIT: New run level: 6
The system is coming down. Please wait.
.
.
.
The system is down.
syncing file systems... [11] [9] [5] done
rebooting...
.
.
.
pluto console login:
```

次に進む手順

システムをシャットダウンした理由が何であれ、最終的には、すべてのファイル資源が使用でき、ユーザーがログインできる実行レベル 3 に戻すことになるでしょう。システムをマルチユーザー状態に移行する手順については、第 8 章または第 9 章を参照してください。

▼ スタンドアロンシステムをシャットダウンする方法

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

```
# init run-level
```

run-level

新しい実行レベル

3. システムが、`init` コマンドで指定した実行レベルに移行したことを確認する方法を以下の表に要約します。

| 移行後の実行レベル | SPARC システムの場合 | x86 システムの場合 |
|---------------------------------|-------------------------|--------------------------|
| 実行レベル S (シングルユーザーモード) | # | # |
| 実行レベル 2 (マルチユーザー状態) | # | # |
| 実行レベル 0 (電源切断状態) | ok または > | type any key to continue |
| 実行レベル 3 (リモート資源が共有されたマルチユーザー状態) | hostname console login: | hostname console login: |

例 — x86 システムを実行レベル 0 にする (スタンドアロン)

次の例では、init コマンドを使用して、x86 スタンドアロンシステムを安全に電源を落とせるレベルにします。

```
# init 0
#
INIT: New run level: 0
The system is coming down. Please wait.
.
.
.
The system is down.
syncing file systems... [11] [10] [3] done
Type any key to continue
```

システムを実行レベル 0 に移行してすべてのデバイスの電源を落とす場合は、140 ページの「すべてのデバイスの電源を落とす方法」を参照してください。

例 — SPARC システムを実行レベル S にする (スタンドアロン)

次の例では、init コマンドとを使用して、SPARC スタンドアロンシステムを実行レベル S (シングルユーザー状態) にしています。

```

# init S
#
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Print services stopped.
syslogd: going down on signal 15
Killing user processes: done.
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode

#

```

次に進む手順

システムをシャットダウンした理由が何であれ、最終的には、すべてのファイル資源が使用でき、ユーザーがログインできる実行レベル 3 に戻すことになるでしょう。システムをマルチユーザー状態に移行する手順については、第 8 章または第 9 章を参照してください。

▼ すべてのデバイスの電源を落とす方法

1. 以下の表を参考にして、該当するシャットダウン作業を実行します。

| サーバーをシャットダウンする場合 | スタンドアロンシステムをシャットダウンする場合 |
|------------------------------|-------------------------------------|
| 134ページの「サーバーをシャットダウンする方法」を参照 | 138ページの「スタンドアロンシステムをシャットダウンする方法」を参照 |

2. システムをシャットダウンしたら、すべてのデバイスの電源を落とします。必要なら、電源ケーブルをコンセントから引き抜きます。
3. 電源が回復したら、次の手順に従ってシステムとデバイスの電源を投入します。
 - a. 電源ケーブルをコンセントに差し込みます。
 - b. モニターの電源を入れます。
 - c. ディスクドライブ、テープドライブ、プリンタの電源を入れます。

d. **CPU** の電源を入れます。

CPU の電源が入ると、システムは実行レベル 3 になります。

SPARC: システムのブートの手順

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

この章で説明する手順は次のとおりです。

- 144ページの「SPARC: ok プロンプトに切り換える方法」
- 144ページの「SPARC: システムの PROM リリースを確認する方法」
- 145ページの「SPARC: デフォルトのブートデバイスを変更する方法」
- 146ページの「SPARC: システムをリセットする方法」
- 148ページの「SPARC: システムを実行レベル 3 (マルチユーザー状態) にする方法」
- 149ページの「SPARC: システムを実行レベル S (シングルユーザー状態) にする方法」
- 150ページの「SPARC: システムを対話式でブートする方法」
- 152ページの「SPARC: システムを復元するためにブートする方法」
- 154ページの「SPARC: 復元のためにシステムを停止する方法」
- 155ページの「SPARC: クラッシュダンプを強制して、システムをリブートする方法」
- 156ページの「SPARC: カーネルデバッガ (kadb) を使ってシステムをブートする方法」

ブート手順の概要については、第 10 章を参照してください。

SPARC: ブート PROM の使用方法

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

永久に、または一時的に新しいドライブをシステムに追加したいときは、デフォルトのブートデバイスを必ず変更してください。あるいは、スタンドアロンシステムをディスクレスクライアントに変換する場合は、必ずネットワークからブートしてください。

すべての PROM コマンドについては、`monitor(1M)` または `eeprom(1M)` のマニュアルページを参照してください。

▼ SPARC: ok プロンプトに切り換える方法

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

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

```
> n
ok
```

この節で示す例では、すべて `ok` プロンプトを使います。

▼ SPARC: システムの PROM リリースを確認する方法

システムの PROM リリースレベルを `banner` コマンドで表示します。

```
ok banner
SPARCstation 2, Type 4 Keyboard
ROM Rev. 2.2, 16 MB memory installed, Serial #nnnnnnn
Ethernet address 8:0:20:f:fd:6c HostID nnnnnnnn
```

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

▼ SPARC: デフォルトのブートデバイスを変更する方法

次の手順で、デフォルトのブートデバイスを変更してください。

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

```
# init 0
```

PROM プロンプトが表示されます。

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

```
> n  
ok
```

ok PROM プロンプトが表示されます。

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

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

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

`disk[n]` `boot-device` 値を指定する。この場合、`n` はディスク番号。

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

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

```
ok printenv boot-device
```

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

```
ok reset
```

boot-device 設定が、PROM に書き込まれます。

SPARC: 例 — デフォルトのブートデバイスを変更する

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... [11] [10] [5] done
Program terminated
Type help for more information
ok setenv boot-device disk
boot-device =          disk
ok printenv boot-device
boot-device          disk          disk
ok reset
SPARCstation 10 (1 X 390Z50), No Keyboard
ROM Rev. 2.14, 32 MB memory installed, Serial #3383708.
Ethernet address 8:0:20:1f:33:9f, Host ID: 7233a19e.
Boot device: /iommu/sbus/espdma@f,400000/esp@f,800000/
sd@3,0 File and args: kadb -v
.
.
.
pluto console login:
```

▼ SPARC: システムをリセットする方法

ok プロンプトで reset コマンドを実行します。

```
ok reset
```

セルフテストプログラムが実行され、ハードウェアで診断テストを行なった後、システムがリブートされます。

SPARC システムのブート

表 8-1 に、この章で説明するブート方法の要約を示します。

表 8-1 ブート方法の説明

| システムのブート方法 | このブート方法が必要な場合 | 参照先 |
|----------------------------------|---|--|
| 実行レベル 3 (NFS 資源を共有するマルチユーザー状態) | システムを停止するか、またなんらかのシステムハードウェアの保守作業を行なった後。これはデフォルトのブートレベルで、すべての資源が利用でき、そしてユーザーがシステムにログインできる。 | 148ページの「例 — システムを実行レベル 3 (マルチユーザー状態) にする」 |
| 実行レベル S (シングルユーザー状態) | ファイルシステムのバックアップなどのなんらかのシステム保守作業を行なった後。このレベルでは、一部のファイルシステムだけがマウントされ、ユーザーはシステムにログインできない。 | 149ページの「SPARC: 例 — システムを実行レベル S (マルチユーザー状態) にする」 |
| 対話式でブート | テストのために、システムファイルまたはカーネルを一時的に変更した後。このブート方法を使えば、システムファイルまたはカーネルに問題がある場合、プロンプトが表示されたときに、これらのファイルに対して別のパス名を指定すれば簡単に復元できる。他のシステムプロンプトにデフォルトの設定を使う。 | 151ページの「SPARC: 例 — システムを対話式でブートする」 |
| ローカル CD-ROM またはネットワークからブートして復元する | システムの正常なブートを妨げている重要なシステムファイルを修復する。また、この形式のブートはオペレーティングシステムの新しいリリースをインストール (あるいは、アップグレード) するのにも使用する。 | 154ページの「SPARC: 例 — システムを復元するためにブートする」 |
| kadb を使ってブート | オペレーティングシステムのコアダンプを保存して、システムの障害を追跡する場合。 | 157ページの「SPARC: 例 — カーネルデバッグ (kadb) を使ってシステムをブートする」 |

システムの電源を切ってから入れ直すと、マルチユーザーのブートシーケンスが開始されます。このあとに示す手順では、ok PROM プロンプトからさまざまな状態でブートする方法を説明します。

who -r コマンドを使って、システムが指定した実行レベルになっていることを確認します。

実行レベルについての説明は、第 6 章を参照してください。

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

1. ブートコマンドを使って実行レベル 3 にします。

```
ok boot
```

自動ブート処理では、一連のスタートアップメッセージ表示して、システムを実行レベル 3 にします。

2. システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面か、ログインプロンプトが表示されます。

```
hostname console login:
```

例 — システムを実行レベル 3 (マルチユーザー状態) にする

次に、システムを実行レベル 3 にするメッセージの例を示します。

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

Boot device: /iommu/sbus/espdma@f,400000/esp@f,800000/sd@3,0
File and args:
SunOS Release 5.7 Version generic [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1998, Sun Microsystems, Inc.
configuring network interfaces: le0.
Hostname: venus
The system is coming up. Please wait.
add net default: gateway 129.152.75.248
NIS domainname is solar.com
starting rpc services: rpcbind keyserver ypbind done.
Setting netmask of le0 to 255.255.255.0
Setting default interface for multicast: add net 224.0.0.0: gateway venus
```

(続く)

続き

```
syslog service starting.  
Print services started.  
volume management starting.  
The system is ready.  
  
venus console login:
```

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

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

```
ok boot -s
```

2. 次のメッセージが表示されたら、**root** パスワードを入力します。

```
INIT: SINGLE USER MODE  
Type Ctrl-d to proceed with normal startup,  
  
(or give root password for system maintenance): xxx
```

3. `who -r` コマンドを使って、システムが実行レベル **S** になっていることを確認します。

```
# who -r  
.          run-level S          May  2 07:39    3          0  S
```

4. システム保守作業の後に、システムをマルチユーザー状態にするには、**Control-d** を押します。

SPARC: 例 — システムを実行レベル S (マルチユーザー状態)にする

次に、実行レベル S になるシステムの表示例を示します。

```
ok boot -s
.
.
.
SunOS Release 5.7 Version [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1998, Sun Microsystems, Inc.
configuring network interfaces: le0.
Hostname: mars
INIT: SINGLE USER MODE
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Sun Microsystems Inc. SunOS 5.7 August 1998
# who -r
. run-level S June 2 07:45 S 0 ?
<保守作業を行う>
# <Control-d を押す>
```

▼ SPARC: システムを対話式でブートする方法

1. boot -a コマンドを使って、システムを対話式でブートします。

```
ok boot -a
```

2. 表 8-2 に示すように、システムプロンプトに答えてください。

表 8-2 SPARC: 対話式によるブート処理手順

| システムの表示 | 操作 |
|--|--|
| Enter filename [kernel/unix]: | ブートに使用する別のカーネルの名前を入力する。 あるいは、そのまま Return キーを押してデフォルトカーネルを使う (/platform/'uname -m'/kernel/unix)。 |
| Name of default directory for modules [/platform/'uname -m'/kernel /kernel /usr/kernel]: | モジュールディレクトリに別のパスを入力して Return キーを押す。 あるいは、そのまま Return キーを押してデフォルトのモジュールディレクトリパスを使う。 |

表 8-2 SPARC: 対話式によるブート処理手順 続く

| システムの表示 | 操作 |
|---|---|
| Name of system file [/etc/system]: | <p>代わりのシステムのファイルの名前を入力して Return キーを押す。 /etc/system ファイルが破損している場合、ファイルとして /dev/null を入力する。</p> <p>あるいは、そのまま Return キーを押してデフォルトの /etc/system ファイルを使う。</p> |
| root filesystem type [ufs]: | <p>そのまま Return キーを押してデフォルトのルートファイルシステム形式を使う。つまり、ローカルディスクからのブートの場合は UFS、ディスクレスクライアントの場合は NFS を使う。</p> |
| Enter physical name of root device [physical_device_name]: | <p>代わりのデバイス名を入力して、Return キーを押す。</p> <p>あるいは、そのまま Return キーを押してルートデバイスのデフォルトの物理名を使う。</p> |

3. 表 8-2 の質問に答えるようなプロンプトが表示されない場合、boot -a コマンドを正しく入力していることを確認してください。

SPARC: 例 — システムを対話式でブートする

次の例では、利用できるデフォルトの選択例 ([] で囲まれています) を示します。

```

ok boot -a
.
.
.
Resetting ...
Rebooting with command: -a
Boot device: /iommusbus/espdma@f,400000/esp@f,800000/sd@3,0
File and args: -a
Enter filename [kernel/unix]: Return キーを押す
Enter default directory for modules
[/platform/SUNW,SPARCstation-10/kernel /platform/sun4m/kernel /kernel /usr/
kernel]:
Return キーを押す
SunOS Release 5.7 Version generic [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1998, Sun Microsystems, Inc.
Name of system file [etc/system]: Return キーを押す
root filesystem type [ufs]: Return キーを押す

```

(続く)

```

Enter physical name of root device
[/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000
/sd@3,0:a]: Return キーを押す
configuring network interfaces: le0.
Hostname: earth
The system is coming up. Please wait.
.
.
.
The system is ready.
earth console login:
    
```

▼ SPARC: システムを復元するためにブートする方法

この手順は、`/etc/passwd` などの重要なファイルに、無効なエントリがあり正常にブートできない場合に必要です。

システムのデバイス名を調べたい場合は、第 20 章を参照してください。

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

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

2. ファイル内に無効なエントリがあるファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /a
```

3. 新しくマウントしたディレクトリに移動します。

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

4. 端末タイプを設定します。

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

5. エディタを使って、ファイルから無効なエントリを削除します。

```
# vi filename
```

6. ルート (/) ディレクトリに移動します。

```
# cd /
```

7. /a ディレクトリのマウントを解除します。

```
# umount /a
```

8. システムをリブートします。

```
# init 6
```

9. システムが実行レベル **3** になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

SPARC: 例 — システムを復元するためにブートする

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

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=sun
# export TERM
# vi passwd
(無効なエントリを削除する)
# cd /
# umount /a
# init 6
```

▼ SPARC: 復元のためにシステムを停止する方法

アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、Stop-A または L1-A を押します。端末では、Break キーを押します。

1. システムのアボートキーシーケンスを入力します。
モニターに `ok PROM` プロンプトが表示されます。

```
ok
```

2. `sync` コマンドを使って、ディスクを同期させます。

```
ok sync
```

3. `syncing file systems. . .` というメッセージが表示されたら、システムのアボートキーシーケンスをもう一度押します。
4. 該当する `boot` コマンドを入力して、ブートプロセスを起動します。
5. システムが指定した実行レベルになっていることを確認します。

```
# who -r
.          run-level 3  May  2 07:39    3    0  S
```

SPARC: 例 — 復元のためにシステムを停止する

```
<Stop-a を押す>
ok sync
syncing file systems...
<Stop-a を押す>
ok boot
```

SPARC: クラッシュダンプを強制してシステムをリブートする

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

▼ SPARC: クラッシュダンプを強制して、システムをリブートする方法

1. システムのアボートキーシーケンスを入力します。アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、**Stop-A** または **L1-A** を押します。端末では、**Break** キーを押します。
モニターに ok PROM プロンプトが表示されます。
2. ok プロンプトで、sync コマンドを使ってディスクを同期させ、クラッシュダンプを書き出します。

```
> n  
ok sync
```

クラッシュダンプがディスクに書き出されると、システムはそのままリブートします。

3. システムが実行レベル **3** になっていることを確認します。ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

SPARC: 例 — クラッシュダンプを強制して、システムをリブートする

```
<Stop-a を押す>  
ok sync
```

▼ SPARC: カーネルデバッガ (kadb) を使ってシステムをブートする方法

1. システムのアボートキーシーケンスを入力します。アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、**Stop-A** または **L1-A** を押します。端末では、**Break** キーを押します。
モニターに、ok PROM プロンプトが表示されます。
2. ok プロンプトで、sync コマンドを使ってディスクを同期させ、クラッシュダンプを書き出します。

```
> n
ok sync
```

3. syncing file systems. . .メッセージが表示されたら、もう一度システムのアボートキーシーケンスを入力します。

4. カーネルデバッグを使ってシステムをブートします。

```
ok boot kadb
```

5. ブートメッセージで、システムがカーネルデバッグ (kadb) を使用してブートしていることを確認します。

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

SPARC: 例 — カーネルデバッグ (kadb) を使ってシステムをブートする

```
<Stop-a を押す>
ok sync
syncing file systems...
<Stop-a を押す>
ok boot kadb
```


x86: システムのブートの手順

この章では、x86 システムをブートする手順を説明します。

この章で説明する手順は次のとおりです。

- 161ページの「x86: システムを実行レベル 3 (マルチユーザー状態) にする方法」
- 162ページの「x86: システムを実行レベル S (シングルユーザー状態) にする方法」
- 163ページの「x86: システムを対話式でブートする方法」
- 165ページの「x86: システムを復元するためにブートする方法」
- 169ページの「x86: 復元を目的としてシステムを停止する方法」
- 170ページの「x86: クラッシュダンプを強制して、システムをリブートする方法」

ブート手順の概要については、第 10 章を参照してください。

x86 システムのブート

表 9-1 に、この章で説明するブート方法の要約を示します。

表 9-1 ブート方法の説明

| システムのブート方法 | このブート方法が必要な場合 | 参照先 |
|----------------------------------|--|---|
| 実行レベル 3 (マルチユーザー状態) | システムを停止するか、またなんらかのシステムハードウェアの保守作業を行なった後。これはデフォルトのブートレベルで、すべての資源が利用でき、そしてユーザーがシステムにログインすることができる。 | 161ページの「x86: 例 — システムを実行レベル 3 (マルチユーザー状態) にする」 |
| 実行レベル S (シングルユーザー状態) | ファイルシステムのバックアップなどのなんらかのシステム保守作業を行なった後。このレベルでは、一部のファイルシステムだけがマウントされ、ユーザーはシステムにログインできない。 | 162ページの「x86: 例 — システムを実行レベル S (シングルユーザー状態) にする」 |
| 対話式でブート | テストのために、システムファイルまたはカーネルを一時的に変更した後。この形式のブートを使えば、システムファイルまたはカーネルに問題がある場合、プロンプトが表示されたときに、これらのファイルに対して別のパス名を指定すれば簡単に復元できる。他のシステムプロンプトにデフォルトの設定を使う。 | 164ページの「x86: 例 — システムを対話式でブートする」 |
| ローカル CD-ROM またはネットワークからブートして復元する | システムの正常なブートを妨げている重要なシステムファイルを修復する。また、この形式のブートは、オペレーティングシステムの新しいリリースをインストール (またはアップグレード) するのにも使用する。 | 167ページの「x86: 例 — システムを復元する目的でブートする」 |
| kadb を使ってブート | オペレーティングシステムのコアダンプを保存して、システムの障害を追跡する場合。 | 170ページの「x86: クラッシュダンプを強制して、システムをリブートする方法」 |

次に、リセットボタンを使ってシステムを再起動する手順を示します。システムにリセットボタンがない場合は、電源のオン・オフスイッチを使ってシステムを再起動します。システムの状態によっては **Control-Alt-Del** キーを使って、システムの動作に割り込むことができます。

▼ x86: システムを実行レベル 3 (マルチユーザー状態) にする方法

1. Type any key to continue プロンプトが表示された場合は、任意のキーを押してシステムをリブートするか、リセットボタンを押してシステムを再起動します。システムがシャットダウンしている場合は、電源スイッチを押してシステムを再起動します。

「Current Boot Parameters」メニューが、数分後に表示されます。

2. システムを実行レベル 3 にするには、b を入力します。Return キーを押してください。

5 秒以内に選択しないと、システムは自動的に実行レベル 3 になります。

3. システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

x86: 例 — システムを実行レベル 3 (マルチユーザー状態) にする

```
Type any key to continue
.
.
.

          <<< Current Boot Parameters >>>
Boot path: /eisa/eha@1,4000/sd@0,0:a
Boot args:

Type      b [file-name] [boot-flags] <ENTER>   to boot with options
or       i <ENTER>                               to enter boot interpreter
or       <ENTER>                                 to boot with defaults

          <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b
.
.
.
```

(続く)

```
venus console login:
```

▼ x86: システムを実行レベル S (シングルユーザー状態) にする方法

1. Type any key to continue プロンプトが表示された場合は、どれかキーを押してシステムをリブートします。このプロンプトが表示されているときは、リセットボタンを押すこともできます。システムがシャットダウンしている場合は、電源スイッチを使用してシステムの電源を入れます。
数分後、「Current Boot Parameters」メニューが表示されます。
2. `b -s` と入力し、システムを実行レベル **S** でブートします。**Return** キーを押します。
5 秒以内に選択しないと、システムは自動的に実行レベル 3 でブートします。
3. プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。
4. `who -r` を使用して、システムが実行レベル **S** になっていることを確認します。

```
# who -r  
.          run-level S  Nov 10 13:59      S    0  3
```

5. 実行レベルを **S** に移行して行う必要がある保守作業を行います。
6. システムを実行レベル **3** に戻すには、**Control-d** を押します。

x86: 例 — システムを実行レベル S (シングルユーザー状態) にする

```
Type any key to continue
.
.
.

<<< Current Boot Parameters >>>
Boot path: /eisa/eha@1,4000/sd@0,0:a
Boot args:

Type   b [file-name] [boot-flags] <ENTER>   to boot with options
or     i <ENTER>                             to enter boot interpreter
or     <ENTER>                               to boot with defaults

<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
.
.
.
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode
.
.
.
# who -r
.      run-level S  Aug  4 13:11      S      0  3
(保守作業を行う)
# Control-d を押す
```

▼ x86: システムを対話式でブートする方法

1. Type any key to continue プロンプトが表示されているときは、任意のキーを押してリブートするか、リセットボタンを押してシステムを再起動します。システムがシャットダウンしているときは、電源スイッチを押してシステムをリブートします。
「Current Boot Parameters」メニューが、数分後に表示されます。
2. b -a と入力して、システムを対話式でブートします。
5 秒以内に選択しないと、システムは自動的に実行レベル 3 でブートします。

3. 表 9-2 で説明するように、システムプロンプトに応答します。

表 9-2 x86: 対話式によるブート処理手順

| システムの表示 | 操作 |
|--|---|
| Type any key to continue | このプロンプトが表示されているときには、任意のキーを押してシステムをリブートする。このプロンプトが表示されているときにリセットボタンを使用することもできる。システムがシャットダウンしているときは、電源スイッチを使用してシステムの電源を入れる。 Primary Boot Subsystemメニューが、数分後に表示される。 |
| Primary Boot Subsystem メニュー | ブートデバイスとして Active Solaris スライスを選択する。Return キーを押す。30 秒以内に選択しないと、アクティブなブートスライスが自動的に選択される。 |
| Select (b)oot or (i)nterpreter: | b -a と入力して、Return キーを押す。 |
| Enter default directory for modules: [/platform/i86pc/kernel /kernel /usr/kernel]: | modules ディレクトリの代替パスを指定して Return キーを押すか、Return キーを押してデフォルトの modules ディレクトリパスを使用する。 |
| Name of system file [etc/system]: | 代わりのシステムファイル名を入力して Return キーを押すか、そのまま Return キーを押してデフォルトの /etc/system ファイルを使う。/etc/system ファイルが破損している場合は、ファイルとして /dev/null を入力する。 |
| root filesystem type [ufs]: | そのまま Return キーを押してデフォルトのルートファイルシステム形式を使う。つまり、ローカルディスクからのブートの場合は UFS、ディスクレスクライアントの場合は NFS を使う。 |
| Enter physical name of root device [physical_device_name]: | 代わりのデバイス名を入力して、Return キーを押すか、そのまま Return キーを押してルートデバイス bootpath のデフォルトの物理名を使う。 |

x86: 例 — システムを対話式でブートする

デフォルトを選択 ([]で囲まれています) した例を示します。

```
Type any key to continue
.
.
.

<<< Current Boot Parameters >>>
Boot path: /eisa/eha@1,4000/sd@0,0:a
Boot args:

Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                  to boot with defaults

<<< timeout in 5 seconds >>>>
Select (b)oot or (i)nterpreter: b -a
Enter default directory for modules [/platform/i86pc/kernel
/kernel /usr/kernel]: Return キーを押す
Name of system file [etc/system]:Return キーを押す
(著作権表示)
root filesystem type [ufs]: Return キーを押す
Enter physical name of root device
[/eisa/dpt@5c88,0/cmdk@0,0:a]: Return キーを押す
Configuring network interfaces: smc0
Hostname: venus
(fsck メッセージ)
The system is coming up. Please wait
(その他のメッセージ)
venus console login:
```

▼ x86: システムを復元するためにブートする方法

システムの復元を目的としたブート例として、無効な `/etc/passwd` ファイルを復元する方法について説明します。

以下の手順では、認識された `devicename` 変数を修復するためにファイルシステムのデバイス名に置き換えます。システムのデバイス名については、第 20 章を参照してください。

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

1. **Solaris** インストール **CD** (またはネットワーク) から、手順 **a** から **f** に従ってシングルユーザーモードでブートします。
ネットワークからブートしている場合は、手順 **a** と **b** を省略します。
 - a. **Solaris** インストール **CD** を **CD** キャディに挿入します。
 - b. **CD** キャディを **CD-ROM** ドライブに挿入します。

- c. (省略可能) ブート元のディスクに **Intel 版 Solaris 2.6** またはその互換バージョンが含まれていない場合、フロッピーディスクドライブ (**DOS ドライブ A**) に **Configuration Assistant/Boot (構成用補助)** フロッピーディスクを挿入します。
- d. Type any key to continue プロンプトが表示されている場合は、どれかキーを押してシステムをリブートします。このプロンプトが表示されているときは、リセットボタンを押すこともできます。システムがシャットダウンしている場合は、電源スイッチを使用してシステムの電源を入れます。
- e. 「**Solaris Device Configuration Assistant**」画面で、**F2 キー (F2_Continue)** を押します。
デバイスが識別され、識別されたデバイスを示す画面が表示されます。
- f. 「**Identified Devices**」画面で、**F2 キー (F2_Continue)** を押します。
ブート可能なドライバが読み込まれます。
- g. 「**Boot Solaris**」画面から **CD** または **NET** をブートデバイスとして選択します。次に **F2 キー (F2_Continue)** を押します。
「Current Boot Parameters」メニューが表示されます。
- h. Select type of installation: プロンプトで **b -s** と入力します。
数分後に、シングルユーザーモードの **#** プロンプトが表示されます。

- 2. 無効な `passwd` ファイルがあるルート (`/`) ファイルシステムをマウントします。

```
# mount /dev/dsk/devicename /a
```

- 3. 新しくマウントした `etc` ディレクトリに移動します。

```
# cd /a/etc
```

- 4. 端末タイプを設定します。


```
# TERM=AT386
# export TERM
```

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

```
# vi passwd
```

6. ルート (/) ディレクトリに移動します。

```
# cd /
```

7. /a ディレクトリのマウントを解除します。

```
# umount /a
```

8. システムをリブートします。

```
# init 6
```

9. システムが実行レベル **3** でブートしたことを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

x86: 例 — システムを復元する目的でブートする

```
Type any key to continue
SunOS Secondary Boot version 3.00
```

```
Solaris Intel Platform Edition Booting System

Running Configuration Assistant...
Autobooting from bootpath: /eisa/eha@1,4000/sd@0,0:a

If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.
.
.
.
Boot Solaris

Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[ ] DISK: Target 0, IMPRIMIS 94241-7 0888
    on Adaptec 1740/1742 SCSI controller in EISA Slot 4
[ ] CD : Target 2, TOSHIBA CD-ROM XM-3501TA 3054
    on Adaptec 1740/1742 SCSI controller in EISA Slot 4
[ ] NET : SMC EtherCard Elite32C Ethernet adapter
    in EISA Slot 6

F2_Continue F3_Back F4_Boot Tasks F6_Help
.
.
.
<<< Current Boot Parameters >>>
Boot path: /eisa/smceu@0,0
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.
```

(続く)

```
Select type of installation: b -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=AT386
# export TERM
# vi passwd
(無効なエントリの削除)
# cd /
# umount /a
# init 6
```

▼ x86: 復元を目的としてシステムを停止する方法

可能であれば、次のコマンドのどちらかを使用してシステムを停止してみます。

- システムが動作している場合、スーパーユーザーになって `init 0` を実行してシステムを停止します。「Type any key to continue」というメッセージが表示されたら、任意のキーを押してシステムをリブートします。
- システムが動作している場合、スーパーユーザーになって `init 6` を実行してシステムをリブートします。

マウスまたはキーボードからの入力にシステムが応答しない場合、リセットキーがあればそのキーを押してシステムをリブートします。あるいは、電源スイッチを使用してシステムをリブートします。

x86: クラッシュダンプを強制してシステムをリブートする

問題に対処するために、オペレーティングシステムのコアダンプを保存しておく必要があります。

`savecore` 機能とその設定方法については、『Solaris のシステム管理 (第 2 巻)』の「システムクラッシュ情報の生成と保存」で説明しています。この節では、`savecore` 機能が有効な場合に、システムをリブートする方法だけを説明します。

▼ x86: クラッシュダンプを強制して、システムをリブートする方法

kadb[0]: プロンプトを表示して、クラッシュダンプを強制するには、カーネルデバッグオプション kadb を指定してシステムをブートします。

x86 - カーネルデバッグ kadb に入るにはテキストモードでなければなりません。したがって、まずウィンドウシステム (CDE または OpenWindows) を終了してください。

1. **Control-Alt-d** を押します。

```
kadb[0]:
```

kadb[0]: プロンプトが表示されます。

2. 次のコマンドを kadb[0]: プロンプトで入力します。

```
Press <Control-Alt-d>
kadb[0]: vfs_syncall/W ffffffff
kadb[0]: 0>eip
kadb[0]: :c
kadb[0]: :c
kadb[0]: :c
```

最初の :c を入力すると、システムはパニックを起こします。そこでもう一度 :c を入力します。システムは再度パニックを起こすため、3 度目の :c を入力し、クラッシュダンプを強制して、システムをリブートします。

クラッシュダンプがディスクに書き込まれた後、システムはリブートします。

3. コンソールログインプロンプトでログインして、システムがリブートされていることを確認します。

ブートプロセスの参照情報

この章では、SPARC や x86 システムでブートするのに使用するハードウェア、および各プラットフォームのブートプロセスの概要を説明します。

この章の内容は次のとおりです。

- 171ページの「SPARC: ブート PROM」
- 172ページの「SPARC: ブートプロセス」
- 172ページの「SPARC: ブートプロセスの詳細」
- 173ページの「x86: PC BIOS」
- 174ページの「x86: ブートサブシステム」
- 179ページの「x86: ブートプロセス」
- 180ページの「x86: ブートプロセスの詳細」

システムのブート手順については、第 8 章または、第 9 章を参照してください。

SPARC: ブート PROM

SPARC システムごとに、「モニター」と呼ばれるプログラムを格納している PROM (プログラム可能な読み取り専用メモリー) チップがあります。モニターは、カーネルが使用される前に、システムの動作を制御します。システムをオンにすると、モニターはシステムのハードウェアやメモリーなどを検査するセルフテスト手順を実行します。エラーが検出されなければ、システムは、自動ブートプロセスを開始します。

SPARC - Solaris システムが動作する前に、以前のシステムの一部には PROM アップグレードが必要なものもあります。詳細については、購入先にお問い合わせください。

SPARC: ブートプロセス

表 10-1 SPARC: ブートプロセス

| | |
|---------------|--|
| ブート PROM フェーズ | ブート PROM が、セルフテスト診断を実行する。 |
| | ブート PROM が、bootblock プログラムを読み込む。 |
| ブートプログラムフェーズ | ブートブロックプログラムが、ufsboot プログラムを読み込む。 |
| | ufsboot プログラムが読み込まれてから、ブート PROM がカーネルを読み込む |
| カーネル初期設定フェーズ | カーネルが自身を初期設定し、ルート (/) ファイルシステムをマウントするのに必要なモジュールをロードする。 |
| | カーネルが init プロセスを起動する。 |
| init フェーズ | init プロセスが、実行制御スクリプトを起動する。 |

SPARC: ブートプロセスの詳細

表 10-2 に、上で図示した SPARC ブートプロセスについて説明します。

表 10-2 SPARC ブートプロセスの説明

| ブートフェーズ | 説明 |
|-----------|---|
| ブート PROM | <p>1. PROM は、システム識別情報を表示し、セルフテスト診断を実行してシステムのハードウェアとメモリーを検査する。</p> <p>2. PROM は一次ブートプログラム bootblk を読み込む。このプログラムの目的は、デフォルトのブートデバイスから ufs ファイルシステムにある二次ブートプログラムを読み込むことにある。</p> |
| ブートプログラム | <p>3. bootblk プログラムは二次ブートプログラム ufsboot を検索して実行し、それをメモリーに読み込む。</p> <p>4. ufsboot プログラムが読み込まれたあと、ufsboot プログラムはカーネルを読み込む。</p> |
| カーネルの初期設定 | <p>5. カーネルが自身を初期設定し、ファイルの読み込みに ufsboot を使ってモジュールのロードを開始する。カーネルはルートファイルシステムをマウントするのに十分なモジュールをロードすると、/ufsboot プログラムの対応づけを解除し、それ自身のリソースを使って動作を続ける。</p> |
| init | <p>6. カーネルがユーザープロセスを作成し、/sbin/init プロセスを起動する。このプロセスは /etc/inittab ファイルを読んで他のプロセスを起動する。</p> <p>7. /sbin/init プロセスが実行制御 (rc) スクリプトを起動し、このスクリプトは一連の他のスクリプトを実行する。それらのスクリプト (/sbin/rc*) はファイルシステムをマウントして検査し、いろいろなプロセスを開始し、システム保守作業を実行する。</p> |

x86: PC BIOS

カーネルが起動される前は、システムは、PC 上のファームウェアインタフェースである、読み取り専用メモリー (ROM) の Basic Input/Output System (BIOS) によって制御されます。

ハードウェアアダプタにオンボード BIOS を搭載することができ、それによってデバイスの物理的特性を表示したり、デバイスにアクセスしたりすることができます。

起動シーケンスの間、PC の BIOS はアダプタ BIOS があるかどうか調べ、あれば、それぞれをロードして、実行します。アダプタの BIOS はそれぞれセルフテスト診断を実行して、デバイス情報を表示します。

x86: ブートサブシステム

Solaris のブート時に、ブートシステムについて次の選択を行えます。

- 一次ブートサブシステム (「Partition Boot」メニュー) — この最初のメニューは、ブート可能な fdisk パーティションがディスク上に複数ある場合に表示されます。このメニューで、どの fdisk パーティションからブートするかを指定できます。何も指定しなければ、デフォルトではアクティブなパーティションからブートします。

Solaris のパーティション以外のパーティションからブートするように指定すると、次の 2 つのメニューは表示されません。

- 自動ブートプロセスへの割り込み — 自動ブートプロセスに割り込むと、Configuration Assistant (構成用補助) にアクセスできます。

Configuration Assistant を使用すると、Solaris システムを別のブートデバイスからブートしたり、新しいハードウェアの構成や構成変更が行え、またデバイスやブートに関連する作業を実行できます。

- 「Current Boot Parameters」メニュー — このメニューには 2 つの形式があります。1 つは、通常の Solaris ブート用で、もう 1 つは Solaris インストールブート用です。
 - 通常の「Current Boot Parameters」メニューでは、オプションを付けて Solaris システムをブートしたり、ブートインタプリタを実行できます。
 - インストール用の「Current Boot Parameters」メニューでは、インストールの種類を選択したり、ブートをカスタマイズできます。

表 10-3 に、主な x86 ブートインタフェースの目的を示します。以降の節では、各ブートサブシステムを詳細に説明し、例を示します。

表 10-3 x86 ブートサブシステム

| ブートサブシステム | このサブシステムメニューの目的 |
|---|--|
| 一次ブートサブシステム | ブートしているディスク上に Solaris fdisk パーティション以外にも複数の fdisk パーティションがある場合に使用する。 |
| 二次ブートサブシステム | Solaris リリースをブートするときに使用。自動ブートプロセスに割り込んで、Solaris Device Configuration Assistant を実行することを選択していない限り、Solaris リリースが自動的にブートされる。 |
| Solaris Device Configuration Assistant ブートフロッピーディスク | 「Solaris Device Configuration Assistant」メニューを表示するには次の 2 つの方法がある。 <ol style="list-style-type: none"> 1. Solaris Device Configuration Assistant フロッピーディスクを使用してシステムをブートする。 2. インストールしたディスクから Solaris をブートするときに自動ブートプロセスに割り込む。 |
| 「Current Boot Parameters」 | Solaris オペレーティング環境がインストールされているディスクからブートするときに使用。あるいは、Solaris インストール CD から、またはネットワーク経由で Solaris リリースをインストールするときに使用。どちらの場合も、このメニューにブートオプションのリストが表示される。 |

ブートプロセス中は、ブートサブシステムメニューは異なるデバイスとブートオプションを表示します。システムは何回かのタイムアウトの後で応答を受け付けなくなった場合、デフォルトの設定値を使って自動的にブートを継続します。ブートサブシステムメニューが表示されるたびに、ブートプロセスを停止することができます。また、自動的に継続させることもできます。

以下の節では、各サブシステムの画面の例を示します。

x86: Solaris のブート

デバイスを識別するフェーズでは、Configuration Assistant (構成用補助) は次の処理を実行します。

- システムにインストールされているデバイスを走査する。
- 識別されたデバイスを表示する。
- キーボードの選択やデバイスとその資源の編集などの追加作業を可能にする。

ブート中に、システムは次の処理を行います。

- ブートするデバイスのリストを表示する。アスタリスク (*) が付いたデバイスがデフォルトのブートデバイスです。
- 自動ブートやプロパティの編集などの追加作業を可能にする。

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

x86: デバイス識別段階で表示されるメニュー

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

x86: Configuration Assistant メニュー

このメニューは、Configuration Assistant を実行するたびに表示されます。

```
Solaris Device Configuration Assistant

The Solaris(TM) (Intel Platform Edition) Device Configuration Assistant
scans to identify system hardware, lists identified devices, and can
boot the Solaris software from a specified device. This program must be
used to install the Solaris operating environment, add a driver,
or change the hardware on the system.

> To perform a full scan to identify all system hardware, choose Continue.
> To diagnose possible full scan failures, choose Specific Scan.
> To add new or updated device drivers, choose Add Driver.

About navigation...
- The mouse cannot be used.
- If the keyboard does not have function keys or they do not respond,
  press ESC. The legend at the bottom of the screen will change to
  show the ESC keys to use for navigation.
- The F2 key performs the default action.

F2_Continue F3_Specific Scan F4_Add Driver F6_Help
```

x86: Bus Enumeration メニュー

このメニューは、Configuration Assistant が自動検出できるデバイスのハードウェア構成データを集めているときに一時的に表示されます。

```
Bus Enumeration

Determining bus types and gathering hardware configuration data ...

Please wait ...
```

x86: Scanning Devices メニュー

このメニューは、特別なドライバだけで検出できるデバイスを Configuration Assistant が手動で走査しているときに表示されます。

```
Scanning Devices

The system is being scanned to identify system hardware.

If the scanning stalls, press the system's reset button. When the
system reboots, choose Specific Scan or Help.

Scanning: Flpppy disk controller

#####
|          |          |          |          |
0         20        40        60        80        100

Please wait ...
```

x86: Identified Devices メニュー

このメニューは、システムで識別されたデバイスを表示します。このメニューからは、「Boot Solaris」メニューに移動できます。また、キーボード構成の設定、デバイスの表示や編集、シリアルコンソールの設定、構成の保存や削除など、付加的な作業も実行できます。

```
Identified Devices

The following devices have been identified on this system. To identify
devices not on this list or to modify device characteristics, such as
keyboard configuration, choose Device Tasks. Platform types may be
included in this list.

EISA: Adaptec 1740/1742 SCSI controller
```

(続く)

```
EISA: Motherboard
EISA: SMC EtherCard Elite32C Ethernet adapter
ISA: Floppy disk controller
ISA: Game port (Joy stick)
ISA: PCMCIA controller
ISA: Parallel port
ISA: Serial port
ISA: System keyboard (US-English)
ISA: VGA w/ 8514/A compatible graphics adapter
```

```
F2_Continue  F3_Back  F4_Device Tasks  F6_Help
```

x86: ブート段階で表示されるメニュー

この段階では、システムをブートする方法を指定できます。

x86: Boot Solaris メニュー

Solaris ブートメニューからは、Solaris リリースをブートするデバイスを選択できます。また、autoboot やプロパティ設定の表示や編集など、オプションの作業も実行できます。ブートデバイスを選択し、[Continue] を選択すると、Solaris カーネルがブートを開始します。

```
Boot Solaris

Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[ ] DISK: Target 0, IMPRIMIS 94241-7 0888
    on Adaptec 1740/1742 SCSI controller in EISA Slot 4
[ ] CD : Target 2, TOSHIBA CD-ROM XM-3501TA 3054
    on Adaptec 1740/1742 SCSI controller in EISA Slot 4
[ ] NET : SMC EtherCard Elite32C Ethernet adapter
    in EISA Slot 6
```

(続く)

```
F2_Continue F3_Back F4_Boot Tasks F6_Help
```

x86: Solaris ブートオプションメニュー

このメニューは、ローカルディスクから Solaris をブートするたびに表示されます。デフォルトの Solaris カーネルからブートしたい場合は、5 秒間のタイムアウトが経過するのを待ちます。別のオプションでブートしたい場合は、タイムアウト期間が経過する前に、適切なオプションを選択します。

```

                <<< Current Boot Parameters >>>
Boot path: /eisa/eha@1,4000/sd@0,0:a
Boot args:

Type      b [file-name] [boot-flags]      to boot with options
or        i                               to enter boot interpreter
or                               to boot with defaults

                <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:

```

x86: ブートプロセス

表 10-4 x86: ブートプロセス

| | |
|--------------|--|
| BIOS フェーズ | PC BIOS が、すべてのハードウェアデバイスの BIOS を読み込んで実行する。 |
| | BIOS ブートプログラムが、マスターブートレコード mboot を読み込んで実行する。 |
| ブートプログラムフェーズ | mboot が、Solaris パーティションブートプログラム pboot を読み込む。 |
| | pboot が一次ブートプログラム bootblk を読み込む。 |

表 10-4 x86: ブートプロセス 続く

| | |
|--------------|---|
| | bootblk が、fdisk テーブルを読み込み、デフォルトのブートパーティションを探す。この時点で、「一次ブートサブシステム」メニューが表示される。 |
| | bootblk が、二次ブートプログラム boot.bin または ufsboot を読み込む。ここで ESC キーを押すと、「Configuration Assistant (構成用補助)」メニューを表示できる。 |
| | 二次ブートプログラム boot.bin または ufsboot が、/etc/bootrc スクリプトを読み込み、このスクリプトがカーネルを読み込む。ここで、Solaris ブートオプションメニューが表示される。 |
| カーネル初期設定フェーズ | カーネルが自身を初期設定し、ルート (/) ファイルシステムをマウントするのに必要なモジュールをロードする。 |
| | カーネルが init プロセスを起動する。 |
| init フェーズ | init プロセスが、実行制御スクリプトを起動する。 |

x86: ブートプロセスの詳細

表 10-5 に、上記の図に示した x86 ブートプロセスについて説明します。

表 10-5 x86 ブートプロセスの説明

| ブートフェーズ | 説明 |
|---------|---|
| BIOS | <p>1. システムの電源をオンにすると、PC BIOS がセルフテスト診断を実行してシステムのハードウェアとメモリーを検査する。エラーが検出されなければ、システムは自動的にブートを開始する。エラーが検出されると、エラーメッセージが表示されて、復元オプションが示される。</p> <p>追加のハードウェアデバイスの BIOS がここで実行される。</p> <p>2. BIOS ブートプログラムが、ブートデバイス (フロッピーディスクかハードドライブ) の最初の物理セクターを読もうとする。ブートデバイス上のこの最初のディスクセクターにはマスターブートレコード mboot が格納されており、このレコードが読み込まれて実行される。mboot ファイルが見つからなかった場合は、エラーメッセージが表示される。</p> |

表 10-5 x86 ブートプロセスの説明 続く

| ブートフェーズ | 説明 |
|----------|--|
| ブートプログラム | <p>3. mboot には、アクティブなパーティションと、Solaris ブートプログラム pboot の位置を見つけるのに必要なディスク情報が、格納されており、mboot は、pboot を読み込んで実行する。</p> <p>4. pboot が一次ブートプログラム bootblk を読み込む。このプログラムは、ufs ファイルシステムにある二次ブートプログラムを読み込む。</p> <p>5. ブート可能なパーティションが複数ある場合、bootblk は fdisk テーブルを読んでデフォルトのブートパーティションをさがし、利用可能なパーティションのメニューを表示する。ブートするパーティションを選択する際に、30 秒経過するとタイムアウトになる。これは、ブート可能なパーティションが複数ある場合にのみ発生する。</p> <p>6. bootblk はルートファイルシステムで二次ブートプログラム boot.bin または ufsboot をさがし、実行する。Configuration Assistant を起動するときは、5 秒以内に自動ブートに割り込みをかける。</p> <p>7. 二次ブートプログラム boot.bin または ufsboot が /etc/bootrc スクリプトを実行するコマンドインタプリタを起動する。ここでシステムのブート方法を選択するメニューが表示される。デフォルトでは、カーネルが読み込まれ、実行される。ブートオプションを指定したり、ブートインタプリタを起動するときは、5 秒以内に行う。</p> |
| カーネル初期設定 | <p>8. カーネルは二次ブートプログラム boot.bin または ufsboot を使用してファイルを読み取り、カーネル自身を初期化して、モジュールの読み込みを開始する。ルートファイルシステムをマウントするのに十分なモジュールをカーネルが読み込むと、二次ブートプログラムのマッピングを解除し、それ自身の資源を使用してブートを継続する。</p> <p>9.カーネルはユーザープロセスを作成し、/sbin/init プロセスを起動する。このプロセスは /etc/inittab ファイルを読み込んで他のプロセスを起動する。</p> |
| init | <p>10. /sbin/init プロセスは実行制御 (rc) スクリプトを開始し、このスクリプトが一連の他のスクリプトを実行する。それらのスクリプト (/sbin/rc*) はファイルシステムを検査、マウントし、さまざまなプロセスを実行して、システム保守作業を実行する。</p> |

パート IV 取り外し可能な媒体の管理

このパートでは、取り外し可能な媒体を Solaris の環境で使用する方法を説明します。次の章で構成されています。

| | |
|--------|---|
| 第 11 章 | 自動マウントと手作業によるマウントの比較を含めて、CD とフロッピーディスクの使用方法についての概要を説明します。 |
| 第 12 章 | コマンド行から CD を使用する手順と、ボリューム管理の起動と終了について説明します。 |
| 第 13 章 | コマンド行からフロッピーディスクをフォーマットして使用する手順を説明します。 |
| 第 14 章 | コマンド行から PCMCIA メモリカードをフォーマットして使用する手順を説明します。 |
| 第 15 章 | Solaris 環境によって、フロッピーディスクおよび CD へのアクセスを容易にするための特別なマウントポイントとシンボリックリンクがどのように作成されるかについて、詳しく説明します。 |



CD とフロッピーディスクの使用方法的概要

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

この章の内容は次のとおりです。

- 185ページの「取り外し可能な媒体の管理についての参照先」
- 186ページの「機能と利点」
- 186ページの「自動マウントと手作業によるマウントの比較」
- 187ページの「フロッピーディスクと CD で行える操作」

取り外し可能な媒体の管理についての参照先

取り外し可能な媒体を管理する手順については、次を参照してください。

- 第 12 章
- 第 13 章
- 第 14 章

OpenWindows 環境または共通デスクトップ環境のファイルマネージャで取り外し可能な媒体を使用する方法については、次のドキュメントを参照してください。

- 『OpenWindows ユーザーズガイド』
- 『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』

機能と利点

Solaris 環境には、ユーザーとソフトウェア開発者用に、フロッピーディスクと CD を扱うための標準インタフェースが用意されています。ボリューム管理と呼ばれるこのインタフェースには、主に次の 3 つの利点があります。

- フロッピーディスクと CD を自動的にマウントすることによって、操作を簡単にします。(手作業によるマウントと自動マウントの比較については、表 11-1 を参照してください。)
- スーパーユーザーでなくても、フロッピーディスクと CD にアクセスできるようにします。
- ネットワーク上の他のシステムが、システムに挿入されたフロッピーディスクと CD に自動的にアクセスできるようにします(第 12 章と第 13 章を参照してください)。

自動マウントと手作業によるマウントの比較

次の表 11-1 は、手作業によるマウント (ボリューム管理を使用しない場合) と自動マウント (ボリューム管理を使用する場合) の手順を比較したものです。

表 11-1 手作業によるマウントと自動マウントの比較

| 手順 | 手作業によるマウント | 自動マウント |
|----|-------------------------------|--|
| 1 | 媒体を挿入する | 媒体を挿入する |
| 2 | スーパーユーザーになる | フロッピーディスクの場合は、 <code>volcheck</code> コマンドを使用する |
| 3 | 媒体デバイスの位置を確認する | ボリューム管理は、CD およびフロッピーディスクを手作業でマウントする場合の大半の作業を、自動的に実行する。 |
| 4 | マウントポイントを作成する | |
| 5 | マウントポイントが現在のディレクトリではないことを確認する | |

表 11-1 手作業によるマウントと自動マウントの比較 続く

| 手順 | 手作業によるマウント | 自動マウント |
|----|---------------------------------|---------------|
| 6 | mount に適切なオプションを付けて、デバイスをマウントする | |
| 7 | スーパーユーザーを終了する | |
| 8 | 媒体上のファイルを操作する | 媒体上のファイルを操作する |
| 9 | スーパーユーザーになる | |
| 10 | 媒体デバイスのマウントを解除する | |
| 11 | 媒体を取り出す | 媒体を取り出す |
| 12 | スーパーユーザーを終了する | |

フロッピーディスクと CD で行える操作

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

表 11-2 フロッピーディスクと CD 上のデータにアクセスする方法

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

表 11-2 フロッピーディスクと CD 上のデータにアクセスする方法 続く

システムに複数のフロッピーディスクや CD-ROM ドライブがある場合は、そのアクセスポイントについて、次の表 11-3 を参照してください。

表 11-3 フロッピーディスクと CD にアクセスする場所

| 媒体デバイス | ファイルシステムにアクセスするための場所 | raw データにアクセスするための場所 |
|--------------------|----------------------|--------------------------|
| 最初のフロッピーディスクドライブ | /floppy/floppy0 | /vol/dev/aliases/floppy0 |
| 2 番目のフロッピーディスクドライブ | /floppy/floppy1 | /vol/dev/aliases/floppy1 |
| 最初の CD-ROM ドライブ | /cdrom/cdrom0 | /vol/dev/aliases/cdrom0 |
| 2 番目の CD-ROM ドライブ | /cdrom/cdrom1 | /vol/dev/aliases/cdrom1 |

コマンド行での CD の使用方法

この章では、Solaris 環境においてコマンド行から CD を使用方法について説明します。この章で説明する手順は次のとおりです。

- 191ページの「CD をロードする方法」
- 191ページの「CD の内容を調べる方法」
- 191ページの「CD から情報をコピーする方法」
- 193ページの「CD が使用中かどうかを調べる方法」
- 193ページの「CD を取り出す方法」
- 194ページの「他のシステム上の CD にアクセスする方法」
- 196ページの「ローカル CD を他のシステムで使用可能にする方法」
- 199ページの「音楽 CD を演奏するようにシステムを設定する方法」
- 200ページの「新しい CD-ROM ドライブ用にシステムを準備する方法」
- 201ページの「ボリューム管理を終了させる方法」
- 201ページの「ボリューム管理を再起動する方法」

CD の使用方法

表 12-1 作業マップ: 一般的な CD の使用方法

| 作業 | 説明 | 手順の説明 |
|-------------------|--|----------------------------|
| CD のロード | CD-ROM ドライブに CD を挿入する。 | 191ページの「CD をロードする方法」 |
| CD の内容の確認 | 省略可能。 CD の内容を調べるには、/cdrom の下にある適切なディレクトリを調べる。 | 191ページの「CD の内容を調べる方法」 |
| ファイルまたはディレクトリのコピー | 省略可能。ファイルシステムの任意の場所からコピーをするのと同様に、CD からファイルまたはディレクトリをコピーする。 | 191ページの「CD から情報をコピーする方法」 |
| CD が使用中かどうかの確認 | 省略可能。CD を取り出す前に、CD が使用中かどうかを確認する。 | 193ページの「CD が使用中かどうかを調べる方法」 |
| CD の取り出し | 作業が終了したら、CD-ROM ドライブから CD を取り出す。 | 193ページの「CD を取り出す方法」 |

CD 名の使用方法

CD を操作する場合、名前または以下の表 12-2 に示す呼び名によって、それらを識別できます。説明を簡単にするため、作業の説明では `cdrom0` を使用しますが、CD 名または別の呼び名を使用することもできます。

表 12-2 CD の識別方法

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

合、そのユーザーはファイルの所有者になりますが、書き込み権は与えられません (これは、CD 上のファイルには書き込み権がないためです)。ユーザー自身がアクセス権を変更する必要があります。

1. **CD** がマウントされていることを確認します。

```
$ ls /cdrom
```

ls コマンドは、マウントされた CD の内容を表示します。内容が表示されない場合は、191ページの「CD をロードする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

| コピーするもの | 使用するコマンド |
|---------|----------|
| ファイル | cp |
| ディレクトリ | cp -r |

例 — CD から情報をコピーする

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

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

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

```
-r--r--r--
```

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

▼ CD が使用中かどうかを調べる方法

1. スーパーユーザーになります。
2. `fuser(1M)` コマンドを入力します。
`fuser` コマンドは、指定する CD に現在アクセス中のプロセスを表示します。

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

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

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

例 — CD が使用中かどうかを調べる

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

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

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

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

▼ CD を取り出す方法

1. **CD** が使用中でないことを確認します。

CD は、シェルまたはアプリケーションがそ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には「使用中」であることを忘れないください。CD のすべてのユーザーを検出したかどうかを確認できない場合は (デスクトップツールの背後に隠れているシェルがアクセスしている可能性がある場合は)、193ページの「CD が使用中かどうかを調べる方法」に説明されている `fuser` コマンドを使用してください。

2. **CD** を取り出します。

```
# eject cdrom0
```

▼ 他のシステム上の CD にアクセスする方法

他のシステム上の CD を各自のファイルシステムに手作業でマウントすることによって、その CD にアクセスできます。ただしこれは、他のシステムが、196ページの「ローカル CD を他のシステムで使用可能にする方法」の指示に従って CD-ROM をエクスポートしている場合にかぎります。

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

```
$ mkdir directory
```

directory

他のシステムの CD のマウントポイントとして作成するディレクトリの名前

2. マウントしたい **CD** の名前を確認します。

手作業でリモート CD をマウントする場合は、ローカル CD で使用可能な `cdrom0` または `cdrom1` 変数を使用できません。CD 名そのものを使用する必要があります。この名前を確認するには、リモートシステムの `/cdrom` ディレクトリで `ls` コマンドを使用してください。オートマウンタが実行されている場合は、マウントしたい CD を持つシステムに `cd` コマンドで移動してから、`ls` コマンドを使用できます。オートマウンタが実行されていない場合は、リモートからログインするなどの別の方法を使用する必要があります。

3. スーパーユーザー権限で、**CD** をマウントします。

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

| | |
|--------------------------|---------------------------|
| <i>system-name</i> | マウントする CD を持つシステムの名前 |
| <i>cd-name</i> | マウントしたい CD の名前 |
| <i>local-mount-point</i> | リモート CD のマウント先のローカルディレクトリ |

4. スーパーユーザーをログアウトします。
5. **CD** が実際にマウントされたかどうかを確認するには、`ls` コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /cdrom
```

例 — 他のシステム上の **CD** にアクセスする

注 - 今回の Solaris リリースの名称は「Solaris 7」ですが、コード、パス名、パッケージパス名などには、「Solaris 2.7」または「SunOS 5.7」という名称が使用されていることがあります。コード、パス、パッケージパスなどを実際に入力または使用ときには、必ずマニュアル中に記述されている名称に従ってください。

次の例では、リモートシステム `mars` 上の `Solaris_2.7_sparc` という名前の CD を、ローカルシステムの `/cdrom` ディレクトリにマウントしています。

```
$ cd /net/mars
$ ls /cdrom
cdrom0      Solaris_2.7_sparc
$ su
Password: password
# mount -F nfs ro mars:/cdrom/Solaris_2.7_sparc /cdrom
# exit
$ ls /cdrom
Solaris_2.7_sparc
```

▼ ローカル CD を他のシステムで使用可能にする方法

システムを設定して、その CD-ROM をエクスポートすることができます。つまり、これらのドライブ上の CD (音楽 CD を除く) を、他のシステムで使用できるようにすることができます。CD-ROM ドライブがエクスポートされると、他のシステムは、194ページの「他のシステム上の CD にアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれる CD にアクセスできます。

1. スーパーユーザーになります。
2. NFS デーモン (nfsd) が実行されているかどうかを確認します。

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

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

3. 以下のうちの該当する手順に進みます。

| 条件 | 次の手順 |
|------------------|-------------|
| nfsd が実行されている場合 | 197ページの手順 8 |
| nfsd が実行されていない場合 | 196ページの手順 4 |

4. nfsd がエクスポートするダミーディレクトリを作成します。

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

dummy-dir

たとえば、*dummy* などの任意のディレクトリ名。このディレクトリにはファイルは含まれない。この目的は、NFS デーモンを「呼び起こして」、エクスポートされた CD-ROM を認識させることにある。

5. 次のエントリを `/etc/dfs/dfstab` に追加します。

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

NFS デーモンを起動すると、このエントリを参照して、エクスポートされたフロッピーディスクドライブを認識します。コメント (`-d` が前に付く) はオプションです。

6. **NFS** デーモンを起動します。

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

7. **NFS** デーモンが実際に実行されていることを確認します。

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

8. 現在ドライブにある **CD** を取り出します。

```
# eject cdrom0
```

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

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

10. 次の行を `/etc/rmmount.conf` に追加します。

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

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

11. /etc/rmmount.conf から書き込み権を削除します。

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

この手順により、ファイルはそのデフォルトのアクセス権に戻ります。

12. CD をロードします。

ここでロードする CD は他のシステムで使用できるようになります。必ずドライブのランプの点滅が消えるまで待って、この作業を確認するようにしてください。

CD にアクセスするために、リモートユーザーは、194ページの「他のシステム上の CD にアクセスする方法」の指示に従って、名前によりその CD をマウントする必要があります。

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

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

```
# share
- /dummy ro "dummy dir to wake up NFS daemon"
- /Solaris_2.7_sparc ro ""
```

例 —ローカル CD を他のシステムで使用可能にする

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

```
# ps -ef | grep nfsd
  root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
  root 10118    1  0 08:24:39 ?          0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(次の行を追加する)
```



```

share -F nfs -o ro /dummy
# eject cdrom0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(次の行をファイルシステム共有セクションに追加する)
share cdrom*
# chmod 444 /etc/rmmount.conf
(CD をロードする)
# share
- /dummy ro ""
- /cdrom/solaris_2_6_sparc/s5 ro ""
- /cdrom/solaris_2_6_sparc/s4 ro ""
- /cdrom/solaris_2_6_sparc/s3 ro ""
- /cdrom/solaris_2_6_sparc/s2 ro ""
- /cdrom/solaris_2_6_sparc/s1 ro ""
- /cdrom/solaris_2_6_sparc/s0 ro ""

```

▼ 音楽 CD を演奏するようにシステムを設定する方法

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

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

1. スーパーユーザーになります。
2. /etc/rmmount.conf を編集します。

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

```

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

```

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

例 — 音楽 CD を演奏するようにシステムを設定する

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

```
# @(#)rmmount.conf 1.3      96/05/10 SMI
#
# Removable Media Mounter configuration file.
#
# File system identification
ident hsfs ident_hsfs.so cdrom
ident ufs ident_ufs.so cdrom floppy rmscsi pcmem
ident pcfs ident_pcfs.so floppy rmscsi pcmem
# Actions
action cdrom action_workman.so /usr/dist/exe/workman
action cdrom action_filemgr.so
action floppy action_filemgr.so
action rmscsi action_filemgr.so
# File System Sharing
share cdrom*
share floppy*
```

▼ 新しい CD-ROM ドライブ用にシステムを準備する方法

`boot -r` コマンドによって正しくブートされたシステムでは、ボリューム管理は新しい CD-ROM ドライブを自動的に認識します。ただし、ボリューム管理が常に新しいドライブを認識するようにするには、`/reconfigure` ファイルを作成する必要があります。

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

```
# touch /reconfigure
```

/reconfigure ファイルは、システムの電源を入れたとき、またはシステムをブートしたときに、Solaris 環境が、新たにインストールされた周辺デバイスがあるかどうかをチェックするように設定するものです。この後で、ボリューム管理は、新しいデバイスを検出し、ユーザーに代わってそれを自動的に管理します。

ボリューム管理の設定

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

▼ ボリューム管理を終了させる方法

1. フロッピーディスクまたは **CD** が使用中でないことを確認します。
フロッピーディスクまたは CD を使用中のすべてのユーザーを確認したかどうかを調べるには、193ページの「CD が使用中かどうかを調べる方法」に説明されている `fuser` コマンドを使用してください。
2. スーパーユーザーになります。
3. `volmgt stop` コマンドを入力します。

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

▼ ボリューム管理を再起動する方法

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

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

コマンド行でのフロッピーディスクのフォーマットと使用方法

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

- 208ページの「UFS フロッピーディスクをフォーマットする方法」
- 211ページの「UFS ファイルシステムをフロッピーディスクに作成する方法」
- 213ページの「DOS フロッピーディスクをフォーマットする方法」
- 216ページの「フロッピーディスクをロードする方法」
- 218ページの「フロッピーディスクの内容を調べる方法」
- 219ページの「フロッピーディスクから情報をコピーまたは移動する方法」
- 220ページの「フロッピーディスクへ情報をコピーまたは移動する方法」
- 221ページの「フロッピーディスクが使用中かどうかを調べる方法」
- 222ページの「フロッピーディスクを取り出す方法」
- 223ページの「他のシステム上のフロッピーディスクにアクセスする方法」
- 225ページの「ローカルフロッピーディスクを他のシステムで使用可能にする方法」

フロッピーディスクのフォーマット

表 13-1 作業マップ: フロッピーディスクのフォーマット方法

| 作業 | 説明 | 手順の説明 |
|---------------------------|---|--|
| フォーマットされていないフロッピーディスクのロード | フロッピーディスクをドライブに挿入して、volcheck コマンドを実行する。 | 216ページの「フロッピーディスクをロードする方法」 |
| フロッピーディスクのフォーマット | UFS 用にフロッピーディスクをフォーマットする。 DOS 用にフロッピーディスクをフォーマットする。 | 208ページの「UFS フロッピーディスクをフォーマットする方法」 213ページの「DOS フロッピーディスクをフォーマットする方法」 |
| UFS ファイルシステムの作成 | UFS の場合のみ。 省略可能。フロッピーディスクをファイルの保存に使用する場合は、UFS ファイルシステムを作成する。文字の保存に使用する場合に、この手順は不要。 | 211ページの「UFS ファイルシステムをフロッピーディスクに作成する方法」 |
| フロッピーディスクの取り出し | フォーマットが終了したら、すぐに使用する場合でも、必ずそのフロッピーディスクを取り出す。 | 222ページの「フロッピーディスクを取り出す方法」 |

フロッピーディスクの名前について

フロッピーディスクを操作する場合は、名前、または次の表 13-2 の呼び名によって識別できます。説明を簡単にするために、作業の説明では floppy0 を使用しますが、フロッピーディスク名または別の呼び名も使用できます。

表 13-2 フロッピーディスクを識別する方法

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

注 - 名前の付いていない (つまり「ラベル」がない) フロッピーディスクには、noname というデフォルト名が割り当てられます。

ハードウェアの考慮点

Solaris システムは、Solaris システム用と DOS システム用にフロッピーディスクをフォーマットできます。ただし、ハードウェアプラットフォームによっていくつかの制限があります。次の表はこの制限を要約したものです。

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

UFS 用にフォーマットされたフロッピーディスクは、それらがフォーマットされたハードウェアプラットフォームに制限されます。つまり、SPARC システムでフォーマットされた UFS フロッピーディスクは、x86 システム上の UFS には使用できません。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の場合も同様です。これは、SPARC と x86 の 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 のいずれかのファイルシステムをサポートする構造からなります。フロッピーディスクを準備するために必要な手順は、ファイルシステムによって異なります。したがって、フロッピーディスクをフォーマットする前には、どの作業が必要かを決めてください。203ページの「フロッピーディスクのフォーマット」を参照してください。

Solaris システム (SPARC または x86 のいずれか) では、7つの異なる密度のフロッピーディスクをフォーマットできます (適切なドライブが必要です)。

| フロッピーディスクのサイズ | フロッピーディスクの密度 | 容量 |
|---------------|--------------|-----------|
| 3.5" | 拡張密度 | 2.88M バイト |
| 3.5" | 高密度 (HD) | 1.44M バイト |
| 3.5" | 中密度 (DD) | 1.2M バイト |
| 3.5" | 低密度 | 720M バイト |
| 5.25" | 高密度 (HD) | 1.2M バイト |
| 5.25" | 中密度 (DD) | 720K バイト |
| 5.25" | 低密度 | 360K バイト |

デフォルトにより、フロッピーディスクドライブは、それに近い密度にフロッピーディスクをフォーマットします。つまり、1.44M バイトのドライブは、フロッピーディスクが実際に 1.44M バイトのフロッピーディスクかどうかに関係なく、特に指示しないかぎり、そのフロッピーディスクを 1.44M バイト用にフォーマットしようとします。1.44M バイトドライブに、フロッピーディスクを、たとえば 720K バイトにフォーマットするように指示できます。しかし、720K バイトドライブに、フロッピーディスクを 1.44M バイト用にフォーマットするよう指示することはできません。つまり、フロッピーディスクもドライブも、その容量以下にフォーマットすることは可能です。

ドライブに対して、フロッピーディスクをデフォルト以外の密度にフォーマットするよう指示するには、次の作業に示すように、`fdformat` コマンドを使用してください。この場合、次の表 13-3 から適切な密度オプションを使用するようにしてください。

表 13-3 密度のオプション

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

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

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

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

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

▼ UFS フロッピーディスクをフォーマットする方法

最初の説明にあるように、SPARC システムでフォーマットされた UFS フロッピーディスクは、SPARC システムでしか使用できません。また、x86 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、Solaris を実行する x86 システムでしか使用できません。



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

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていないフロッピーディスクを挿入すると、フォーマット用ウィンドウを自動的に表示します。ただし、以下で説明するフォーマット方法が、ファイルマネージャによるフォーマットよりも確実です。このウィンドウが表示されないようにするには、ファイルマネージャを終了してください。ファイルマネージャを開いておきたい場合は、フォーマットウィンドウが表示されたときに、それを終了してください。

2. フロッピーディスクが書き込み可能になっていることを確認します。

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

3. フロッピーディスクを挿入します。

フロッピーディスクが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

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

| | |
|----------------------------|---|
| -v | フロッピーディスクが正しくフォーマットされたかどうかを確認する。 |
| -U | フロッピーディスクがマウントされている場合は、それを解除する。 |
| <i>density-options</i> | ドライブ密度が 1.44M バイトの場合、密度オプションは次のようになる。 |
| — なし — | 1.44M バイトにフォーマットする。 |
| -D | 720K バイトにフォーマットする。 |
| | <i>density-options</i> のリストは、表 13-3 を参照。 |
| <i>convenience-options</i> | |
| -e | フォーマットが終了すると、フロッピーディスクを取り出す。 |
| -f | フォーマットの前に確認を要求しない。 |
| -b <i>label</i> | フロッピーディスクに名前を付ける。名前は、8 文字以下にする必要がある。大文字小文字は区別されない。 |
| -z | <code>fdformat</code> コマンドのすべてのオプションを一覧表示するが、フロッピーディスクはフォーマットしない。 |

注 - 720K バイト (DD) のフロッピーディスクを 1.44M バイト用にフォーマットしようとしたときに、`-v` オプションが指定されていない場合は `fdformat` の処理は中止されません。`-v` オプションがされている場合には、`fdformat` はフロッピーディスクをフォーマットしますが、検査でエラーが検出されて次のメッセージが出力されます。`fdformat: check diskette density, I/O error`

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

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



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

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

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

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

```
$ volrmmount -i floppy0
```

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

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

```
$ ls /floppy
floppy0
```

例 — UFS ファイルシステムをフロッピーディスクに作成する

```
$ volcheck -v
media was found
$ /usr/sbin/newfs -v /vol/dev/aliases/floppy0
newfs: construct a new file system /vol/dev/aliases/floppy0: (y/n)? y
mkfs -F ufs /vol/dev/aliases/floppy0 2880 18 2 8192 1024 16 10 5 2048
t 0 -1 8 15
/vol/dev/aliases/floppy0: 2880 sectors in 80 cylinders of 2 tracks,
18 sectors
      1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 640, 1184, 1792, 2336,
$ volrmmount -i floppy0
$ ls /floppy
```

(続く)

```
floppy0
```

▼ DOS フロッピーディスクをフォーマットする方法

SPARC および x86 のどちらの Solaris システムでも、DOS フロッピーディスクをフォーマットできます。両者の手順は同じですが、SPARC の場合はフロッピーディスク上に SunOS ファイルシステムが作成されるのに対して、x86 の場合はファイルシステム上に DOS ファイルシステム (MS-DOS または NEC-DOS) が作成されます。



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

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていないフロッピーディスクを挿入すると、フォーマットウィンドウを自動的に表示します。ただし、以下で説明するフォーマット方法が、ファイルマネージャによるフォーマットよりも確実です。このウィンドウが表示されないようにするには、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、それを終了してください。

2. フロッピーディスクが書き込み保護になっていないことを確認します。

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

3. フロッピーディスクを挿入します。

フロッピーディスクが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

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

| | |
|----------------------------|--|
| -v | フロッピーディスクが正しくフォーマットされたかどうかを確認する。 |
| -U | フロッピーディスクがマウントされている場合は、それを解除する。 |
| <i>density-options</i> | ドライブ密度が 1.44M バイトの場合、密度オプションは次のようになる。 |
| -d | 1.44M バイトで MS-DOS 用にフォーマットする。 |
| -d -D | 720K バイトで MS-DOS 用にフォーマットする。 |
| -t nec -M | 1.2M バイトで NEC-DOS 用にフォーマットする。 |
| | <i>density-options</i> のリストは、表 13-3 を参照。 |
| <i>convenience-options</i> | |
| -e | フォーマットが終了すると、フロッピーディスクを取り出す。 |
| -f | フォーマットの前に確認を要求しない。 |
| -b <i>label</i> | フロッピーディスクに名前を付ける。名前は、8 文字以下にする必要がある。大文字小文字は区別されない。 |
| -z | fdformat コマンドのすべてのオプションを一覧表示するが、フロッピーディスクはフォーマットしない。 |

注 - 720K バイト (DD) のフロッピーディスクを、1.44M バイト用にフォーマットしようとした場合、fdformat は、-v オプションが指定されないかぎり、処理を中止しない。-v オプションを使用すると、fdformat は、フロッピーディスクをフォーマットするが、検査でエラーが検出されて、次のメッセージによって通知される。fdformat: check diskette density, I/O error

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

フロッピーディスクの使用方法

表 13-4 作業マップ: 一般的なフロッピーディスクの使用方法

| 作業 | 説明 | 手順の説明 |
|----------------------|---|---|
| フロッピーディスクのロード | フロッピーディスクをドライブに挿入して、 <code>volcheck</code> コマンドを実行する。 | 216ページの「フロッピーディスクをロードする方法」 |
| フロッピーディスクの内容の確認 | 省略可能。フロッピーディスクの内容を調べるには、 <code>/diskette</code> の下にある適切なディレクトリを調べる。 | 218ページの「フロッピーディスクの内容を調べる方法」 |
| ファイルの交換 | 省略可能。フロッピーディスクとファイルシステムの間でファイルまたはディレクトリをコピーする。 | 219ページの「フロッピーディスクから情報をコピーまたは移動する方法」 220ページの「フロッピーディスクへ情報をコピーまたは移動する方法」 |
| フロッピーディスクが使用中かどうかの確認 | 省略可能。フロッピーディスクを取り出す前に、フロッピーディスクが使用中かどうかを調べる。 | 221ページの「フロッピーディスクが使用中かどうかを調べる方法」 |
| フロッピーディスクの取り出し | 終了したら、フロッピーディスクをドライブから取り出す。 | 222ページの「フロッピーディスクを取り出す方法」 |

▼ フロッピーディスクをロードする方法

1. フロッピーディスクがフォーマットされていることを確認します。

不確かな場合は、フロッピーディスクを挿入して、217ページの手順3で説明されているように、コンソールの状態メッセージをチェックしてください。フロッピーディスクをフォーマットする必要がある場合は、208ページの「UFS フロッピーディスクをフォーマットする方法」、または 213ページの「DOS フロッピーディスクをフォーマットする方法」を参照してください。

2. フロッピーディスクを挿入します。

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

3. ボリューム管理に通知します。

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

次の2つの状態メッセージのどちらかが表示されます。

media was found

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

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

フロッピーディスクがフォーマットされていない場合でも、「media was found」メッセージは表示されるが、次のエラーメッセージがコンソールに表示される。

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

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

```
fd0: bad format
```

フロッピーディスクをフォーマットしてからでないと、ボリューム管理はそれをマウントできない。フォーマットの方法は、208ページの「UFS フロッピーディスクをフォーマットする方法」(UFS 用)と213ページの「DOS フロッピーディスクをフォーマットする方法」(DOS 用)を参照。

no media was found

ボリューム管理は、フロッピーディスクを検出しなかった。フロッピーディスクが正しく挿入されていることを確認して、volcheck をもう一度実行する。うまくいかない場合は、フロッピーディスクをチェックする。損傷の可能性がある。フロッピーディスクを手作業でマウントしてみることもできる。

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

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

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

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

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

このような行が表示されない場合、フロッピーディスクはマウントされていません。コンソールのエラーメッセージを確認してください。

▼ フロッピーディスクの内容を調べる方法

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

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

-L 出力にシンボリックリンクを含める。

-l 詳細表示。アクセス権と所有権を出力に含める。

例 — フロッピーディスクの内容を調べる

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

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

▼ フロッピーディスクから情報をコピーまたは移動する方法

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

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

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

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

/floppy ディレクトリに何も表示されない場合、フロッピーディスクはマウントされていません。216ページの「フロッピーディスクをロードする方法」を参照してください。フロッピーディスクがフォーマットされていない場合もあります。208ページの「UFS フロッピーディスクをフォーマットする方法」、または213ページの「DOS フロッピーディスクをフォーマットする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

| コピーするもの | 使用するコマンド |
|---------|----------|
| ファイル | cp |
| ディレクトリ | cp -r |

3. コピーまたは移動を確認するには、ls コマンドを使用します。

例 — フロッピーディスクから情報をコピーまたは移動する

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

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

▼ フロッピーディスクへ情報をコピーまたは移動する方法

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

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

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

`/floppy` ディレクトリに何も表示されない場合、フロッピーディスクはマウントされていません。216ページの「フロッピーディスクをロードする方法」に示された作業を参照してください。フロッピーディスクがフォーマットされていない場合もあります。208ページの「UFS フロッピーディスクをフォーマットする方法」、または 213ページの「DOS フロッピーディスクをフォーマットする方法」を参照してください。

3. ファイルまたはディレクトリを移動またはコピーします。

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

4. 移動またはコピーを確認するには、ls コマンドを使用します。

例 — 情報をフロッピーディスクにコピーまたは移動する

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

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

▼ フロッピーディスクが使用中かどうかを調べる方法

1. スーパーユーザーになります。
2. fuser コマンドを起動します。

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

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

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

例 — フロッピーディスクが使用中かどうかを調べる

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

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

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

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

▼ フロッピーディスクを取り出す方法

1. フロッピーディスクが使用中でないことを確認します。

フロッピーディスクは、シェルまたはアプリケーションがそのファイルやディレクトリにアクセスしている場合「使用中」であることを忘れないでください。

フロッピーディスクのすべてのユーザーを検出したかどうかが不確かな場合は (デスクトップツールの背後に隠れたシェルでアクセスしている可能性がある場合は)、221ページの「フロッピーディスクが使用中かどうかを調べる方法」に説明されている fuser コマンドを使用してください。

2. フロッピーディスクを取り出します。

```
# eject floppy0
```


SPARC システムでは、フロッピーはドライブから物理的に取り出されますが、x86 システムでは手でフロッピーディスクを取り出す必要があります。ウィンドウを実行している場合は、フロッピーディスクを取り出すように指示する画面メッセージに注意してください。

フロッピーディスクが使用中の場合は、次のメッセージが表示されます。

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

この場合は、222ページの手順 1 に戻って、誰もフロッピーディスクを使用していないことを確認してから、もう一度それを取り出してください。

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

▼ 他のシステム上のフロッピーディスクにアクセスする方法

他のシステム上のフロッピーディスクを各自のファイルシステムに手作業でマウントすることによって、そのフロッピーディスクにアクセスできます。ただしこれは、他のシステムが、225ページの「ローカルフロッピーディスクを他のシステムで使用可能にする方法」の指示に従ってフロッピーディスクドライブをエクスポートしている場合にかぎります。

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

```
$ mkdir directory
```

directory

他のシステムのフロッピーディスクのマウントポイントとして作成するディレクトリの名前

2. マウントしたいフロッピーディスクの名前を検索します。

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

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

3. スーパーユーザーとして、フロッピーディスクをマウントします。

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

| | |
|--------------------------|--------------------------------|
| <i>system-name</i> | マウントするフロッピーディスクが存在するシステムの名前 |
| <i>diskette-name</i> | マウントしたいフロッピーディスクの名前 |
| <i>local-mount-point</i> | リモートフロッピーディスクのマウント先のローカルディレクトリ |

4. スーパーユーザーをログアウトします。
5. フロッピーディスクが実際にマウントされたかどうかを確認するには、`ls` コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /floppy
```

例 — 他のシステム上のフロッピーディスクにアクセスする

次の例では、`myfiles` という名前の フロッピーディスクを、リモートシステム `mars` からローカルシステムの `/floppy` ディレクトリにマウントしています。

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

▼ ローカルフロッピーディスクを他のシステムで使用可能にする方法

システムを設定して、そのフロッピーディスクをエクスポートすることができます。つまり、これらのドライブ上のフロッピーディスクを、他のシステムが使用できるようにになります。フロッピーディスクドライブがエクスポートされると、他のシステムは、223ページの「他のシステム上のフロッピーディスクにアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれるフロッピーディスクにアクセスできます。

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

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

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

3. 以下の表から該当する手順を選択します。

| 条件 | 次の手順 |
|-------------------------------|-------------|
| <code>nfsd</code> が実行されている場合 | 226ページの手順 8 |
| <code>nfsd</code> が実行されていない場合 | 225ページの手順 4 |

4. `nfsd` がエクスポートするダミーディレクトリを作成します。

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

dummy-dir

たとえば、*dummy* などの任意のディレクトリ名にすることができる。このディレクトリには、ファイルは含まれない。これは、NFS デーモンを「呼び起こして」、エクスポートされたフロッピーディスクを認識させることを目的としています。

5. 次のエントリを `/etc/dfs/dfstab` に追加します。

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

NFS デーモンを起動すると、このエントリを参照して、エクスポートされたフロッピーディスクドライブを認識します。コメント (`-d` が前に付く) はオプションです。

6. NFS デーモンを起動します。

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

7. NFS デーモンが実際に実行されていることを確認します。

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

8. 現在ドライブ内にあるフロッピーディスクを取り出します。

```
# eject floppy0
```

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

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

10. 次の行を `/etc/rmmount.conf` に追加します。

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

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

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

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

この手順により、ファイルはそのデフォルトのアクセス権に戻ります。

12. フロッピーディスクをロードします。

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

ここでロードするフロッピーディスクは他のシステムで使用できるようになります。フロッピーディスクにアクセスするために、リモートユーザーは、223ページの「他のシステム上のフロッピーディスクにアクセスする方法」の指示に従って、名前によりそのフロッピーディスクをマウントする必要があります。

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

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

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

例 — ローカルフロッピーディスクを他のシステムで使用可能にする

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

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

コマンド行での PCMCIA メモリーカードの使用方法

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

この章で説明する手順は次のとおりです。

- 232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」
- 234ページの「UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法」
- 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」
- 240ページの「PCMCIA メモリーカードをロードする方法」
- 242ページの「PCMCIA メモリーカードの内容を調べる方法」
- 243ページの「PCMCIA メモリーカードから情報をコピーまたは移動する方法」
- 244ページの「PCMCIA メモリーカードへ情報をコピーまたは移動する方法」
- 245ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」
- 246ページの「PCMCIA メモリーカードを取り出す方法」
- 247ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」
- 249ページの「ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする方法」

PCMCIA メモリーカードのフォーマット

表 14-1 作業マップ: PCMCIA メモリーカードのフォーマット方法

| 作業 | 説明 | 手順の説明 |
|---------------------------------|---|--|
| フォーマットされていない PCMCIA メモリーカードのロード | IPCMCIA メモリーカードをドライブに挿入し、volcheck コマンドを実行する。 | 240ページの「PCMCIA メモリーカードをロードする方法」 |
| PCMCIA メモリーカードのフォーマット | PCMCIA メモリーカードを、UFS 用にフォーマットする。 PCMCIA メモリーカードを、DOS 用にフォーマットする。 | 232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」 |
| UFS ファイルシステムの作成 | UFS の場合のみ。省略可能。PCMCIA メモリーカードをファイルの保存に使用する場合は、UFS ファイルシステムを作成する。文字の保存に使用する場合には、この手順は不要。 | 234ページの「UFS ファイルシステムを PCMCIA メモリーカード上に作成する方法」 |
| PCMCIA メモリーカードの取り出し | PCMCIA メモリーカードのフォーマットを終了したら、すぐに使用する場合でも、必ずそれを取り出す。 | 246ページの「PCMCIA メモリーカードを取り出す方法」 |

PCMCIA メモリーカードの名前について

PCMCIA メモリーカードを操作する場合は、名前、または次の表 14-2 に示す呼び名によってそれらを識別できます。説明を簡単にするために、作業の説明では pcmem0 を使用しますが、PCMCIA メモリーカード名または別の呼び名も使用できます。

表 14-2 PCMCIA メモリーカードを識別する方法

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

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

ハードウェアの考慮点

Solaris システムは、Solaris システム用と DOS システム用に PCMCIA メモリーカードをフォーマットできます。ただし、ハードウェアプラットフォームにいくつかの制限があります。次の表はこの制限を要約したものです。

| プラットフォームの種類 | PCMCIA メモリーカードのフォーマット仕様 |
|------------------|-----------------------------|
| SPARC 上の Solaris | SPARC 上の Solaris (UFS) 用 |
| | MS-DOS または NEC-DOS (PCFS) 用 |
| x86 上の Solaris | x86 上の Solaris (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、それらが形式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。x86 システム上でフォーマットされた PCMCIA メモリーカードの場合も同様です。これは、SPARC と x86 とで UFS フォーマットが異なるためです。

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

ルシステムをサポートする構造からなります。PCMCIA メモリーカードを準備するために必要な手順はファイルシステムによって異なります。したがって、PCMCIA メモリーカードをフォーマットする前には、どの作業が必要かを決めてください。230ページの「PCMCIA メモリーカードのフォーマット」を参照してください。

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

▼ UFS PCMCIA メモリーカードをフォーマットする方法

最初の説明にあるように、SPARC システムでフォーマットされた UFS PCMCIA メモリーカードは、SPARC システムでしか使用できません。また、x86 システムでフォーマットされた UFS PCMCIA メモリーカードは、Solaris を実行する x86 システムでしか使用できません。



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

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていない PCMCIA メモリーカードを挿入すると、フォーマット用ウィンドウを自動的に表示します。ただし、以下で説明するフォーマット方法が、ファイルマネージャによるフォーマットよりも確実です。このウィンドウが表示されないようにするには、ファイルマネージャを終了してください。ファイルマネージャを開いておきたい場合は、フォーマットウィンドウが表示されたときに、それを終了してください。

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

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

3. PCMCIA メモリーカードを挿入します。

PCMCIA メモリーカードが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

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

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

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

convenience-options

-e フォーマットが終了すると、PCMCIA メモリーカードを取り出す。

-f フォーマットの前に確認を要求しない。

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

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

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

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

5. 目的に応じて、以下のいずれかのキーを使用します。

でしか使用できません。また、x86 システムでフォーマットされた UFS PCMCIA メモリーカードは、Solaris を実行する x86 システムでしか使用できません。

1. **UFS** ファイルシステム用に **PCMCIA** メモリーカードをフォーマットします。
232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」で説明した手順に従ってください。

2. `newfs` コマンドと、ボリューム管理ディレクトリへの絶対パス名を使用して、**PCMCIA** メモリーカード上に **UFS** ファイルシステムを作成します。

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

`-v` 状態メッセージを出力する。

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

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

3. ファイルシステムの作成を確認します。

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

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

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

(続く)

続き

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

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

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

```
$ volrmmount -i pcmem0
```

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

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

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

```
$ ls /pcmem
pcmem0
```

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

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

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

$ volrmmount -i pcmem0
```

```
media was found
```

▼ DOS PCMCIA メモリーカードをフォーマットする方法

SPARC および x86 のどちらの Solaris システムでも、DOS PCMCIA メモリーカードをフォーマットできます。両者の手順は同じですが、SPARC の場合は PCMCIA メモリーカード上に SunOS ファイルシステムが作成されるのに対して、x86 の場合はファイルシステム上に DOS ファイルシステム (MS-DOS または NEC-DOS) が作成されます。



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

1. ファイルマネージャを終了します。

ファイルマネージャは、フォーマットされていない PCMCIA メモリーカードを挿入すると、フォーマットウィンドウを自動的に表示します。ただし、以下で説明するフォーマット方法が、ファイルマネージャによるフォーマットよりも確実です。このウィンドウが表示されないようにするには、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、それを終了してください。

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

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

3. PCMCIA メモリーカードを挿入します。

PCMCIA メモリーカードが完全に挿入されたことを確認してください。

4. フォーマットを実行します。

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

| | |
|----------------------------|--|
| -v | PCMCIA メモリーカードが正しくフォーマットされたかどうかを確認する。 |
| -U | PCMCIA メモリーカードがマウントされている場合は、それを解除する。 |
| <i>density-options</i> | ドライブ密度が 1.44M バイトの場合、 <i>density-options</i> は次のとおり。 |
| -d | MS-DOS 用にフォーマットする。 |
| -t nec -M | 1.2M バイトで NEC-DOS 用にフォーマットする。 |
| | 密度オプションの一覧は、 <code>fdformat(1)</code> のマニュアルページを参照。 |
| <i>convenience-options</i> | |
| -e | フォーマットが終了すると、PCMCIA メモリーカードを取り出す。 |
| -f | フォーマットの前に確認を要求しない。 |
| -b <i>label</i> | PCMCIA メモリーカードに名前を付ける。名前は、8 文字以下にする。大文字と小文字は区別されない。 |
| -z | <code>fdformat</code> コマンドのすべてのオプションを一覧表示するが、PCMCIA メモリーカードはフォーマットしない。 |

注 - 720K バイト (DD) のフロッピーディスクを 1.44M バイト用にフォーマットしようとしたときに、`-v` オプションが指定されていない場合は `fdformat` の処理は中止されません。`-v` オプションが指定されている場合には、`fdformat` はフロッピーディスクをフォーマットしますが、検査でエラーが検出されて次のメッセージ出力されます。`fdformat: check diskette density, I/O error`

実行されるフォーマットのタイプと、確認メッセージが表示されます。

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


5. 目的に応じて、以下のいずれかのキーを使用します。

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

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

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

```
$ volrmmount -i pcmem0
```

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

PCMCIA メモリーカードの使用方法

表 14-3 作業マップ: PCMCIA メモリーカードの使用方法

| 作業 | 説明 | 手順の説明 |
|----------------------|--|----------------------------------|
| PCMCIA メモリーカードのロード | PCMCIA メモリーカードをドライブに挿入して、 <code>volcheck</code> コマンドを実行する。 | 240ページの「PCMCIA メモリーカードをロードする方法」 |
| PCMCIA メモリーカードの内容の確認 | 省略可能。PCMCIA メモリーカードの内容を調べるには、 <code>/PCMCIAmemorycard</code> の下にある適切なディレクトリを調べる。 | 242ページの「PCMCIA メモリーカードの内容を調べる方法」 |

表 14-3 作業マップ: PCMCIA メモリーカードの使用法 続く

| 作業 | 説明 | 手順の説明 |
|---------------------------|---|---|
| ファイルの交換 | 省略可能。PCMCIA メモリーカードとファイルシステムの間でファイルまたはディレクトリをコピーする。 | 243ページの「PCMCIA メモリーカードから情報をコピーまたは移動する方法」 244ページの「PCMCIA メモリーカードへ情報をコピーまたは移動する方法」 |
| PCMCIA メモリーカードが使用中かどうかの確認 | 省略可能。PCMCIA メモリーカードを取り出す前に、PCMCIA メモリーカードが使用中であるかどうかを調べる。 | 245ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」 |
| PCMCIA メモリーカードの取り出し | 終了したら、PCMCIA メモリーカードを取り出す。 | 246ページの「PCMCIA メモリーカードを取り出す方法」 |

▼ PCMCIA メモリーカードをロードする方法

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

確認できない場合は、PCMCIA メモリーカードを挿入して、以下の239ページの「PCMCIA メモリーカードの使用法」で説明されているように、コンソールの状態メッセージをチェックしてください。PCMCIA メモリーカードをフォーマットする必要がある場合は、232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

2. PCMCIA メモリーカードを挿入します。

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

3. ボリューム管理に通知します。

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

次の2つの状態メッセージのいずれかが表示されます。

media was found

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

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

PCMCIA メモリーカードがフォーマットされていない場合でも、「media was found」メッセージは表示されるが、次のエラーメッセージがコンソールに表示される。

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

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

```
fd0: bad format
```

PCMCIA メモリーカードをフォーマットしてからでないと、ボリューム管理はそれをマウントできない。フォーマットの方法は、232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」(UFS 用)と 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」(DOS 用)を参照。

no media was found

ボリューム管理は、PCMCIA メモリーカードを検出しなかった。PCMCIA メモリーカードが正しく挿入されていることを確認して、volcheck をもう一度実行する。うまくいかない場合は、PCMCIA メモリーカードをチェックする。損傷の可能性がある。PCMCIA メモリーカードを手作業でマウントしてみることもできる。

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

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

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

`/pcmem` ディレクトリに何も表示されない場合は、PCMCIA メモリーカードがマウントされていないか、または正しくフォーマットされていないかのいずれかです。これを調べるには、`mount` コマンドを実行して、`/pcmem` で始まる行を探してください (通常は、リストの最後にあります)。

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

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

▼ PCMCIA メモリーカードの内容を調べる方法

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

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

`-L` 出力にシンボリックリンクを含む。

`-l` 長形式。アクセス権と所有権を出力に含む。

例 — PCMCIA メモリーカードの内容を調べる

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

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

▼ PCMCIA メモリーカードから情報をコピーまたは移動する方法

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

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

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

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

/pcmem ディレクトリに何も表示されない場合、PCMCIA メモリーカードはマウントされていません。240ページの「PCMCIA メモリーカードをロードする方法」を参照してください。PCMCIA メモリーカードがフォーマットされていないこともあります。232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

2. ファイルまたはディレクトリをコピーします。

| コピーするもの | 使用するコマンド |
|---------|----------|
| ファイル | cp |
| ディレクトリ | cp -r |

3. コピーまたは移動の結果を確認するには、ls コマンドを使用します。

例 — PCMCIA メモリーカードから情報をコピーまたは移動する

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

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

▼ PCMCIA メモリーカードへ情報をコピーまたは移動する方法

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

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

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

`/pcmem` ディレクトリに何も表示されない場合、PCMCIA メモリーカードはマウントされていません。240ページの「PCMCIA メモリーカードをロードする方法」を参照してください。PCMCIA メモリーカードがフォーマットされていないこともあります。232ページの「UFS PCMCIA メモリーカードをフォーマットする方法」、または 237ページの「DOS PCMCIA メモリーカードをフォーマットする方法」を参照してください。

3. ファイルまたはディレクトリを移動またはコピーします。

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

4. 移動またはコピーの結果を確認するには、ls コマンドを使用します。

例 — 情報を PCMCIA メモリーカードにコピーまたは移動する

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

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

▼ PCMCIA メモリーカードが使用中かどうかを調べる方法

1. スーパーユーザーになります。
2. fuser コマンドを起動します。

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

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

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

例 — PCMCIA メモリーカードが使用中かどうかを調べる

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

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

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

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

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

▼ PCMCIA メモリーカードを取り出す方法

1. PCMCIA メモリーカードが使用中でないことを確認します。

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

PCMCIA メモリーカードのすべてのユーザーを検出したかどうか不確かな場合は (デスクトップツールの背後に隠れたシェルがアクセスしている可能性がある場合は)、245ページの「PCMCIA メモリーカードが使用中かどうかを調べる方法」に説明されている fuser コマンドを使用してください。

2. PCMCIA メモリーカードを取り出します。


```
# eject pcmem0
```

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

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

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

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

▼ 他のシステム上の PCMCIA メモリーカードにアクセスする方法

他のシステム上の PCMCIA メモリーカードを各自のファイルシステムに手作業でマウントすることによって、その PCMCIA メモリーカードにアクセスできます。ただしこれは、他のシステムが、249ページの「ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする方法」の指示に従って PCMCIA メモリーカードドライブをエクスポートしている場合にかぎります。

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

```
$ mkdir directory
```

directory

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

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

手作業でリモート PCMCIA メモリーカードをマウントする場合は、ローカル PCMCIA メモリーカードで使用可能な `pcmem0` または `floppy1` 変数を使用できません。正しい PCMCIA メモリーカード名を使用する必要があります。この名前を検索するには、リモートシステムの `/pcmem` ディレクトリで `ls` コマンドを使用してください。オートマウンタが実行されている場合は、マウントしたい PCMCIA メモリーカードのシステムに `cd` コマンドで移動してから、`ls` コマン

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

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

```
# mount -F nfs -o rw system-name:/pcmem/PCMCIA memory-card-name local-mount-point
```

| | |
|--------------------------------|--|
| <i>system-name</i> | マウントする PCMCIA メモリーカードが存在するシス テムの名前 |
| <i>PCMCIA memory-card-name</i> | マウントしたい PCMCIA メモリーカードの名前 |
| <i>local-mount-point</i> | リモート PCMCIA メモリーカードのマウント先の ローカルディレクトリ |

4. スーパーユーザーをログアウトします。
5. **PCMCIA** メモリーカードが実際にマウントされたかどうかを確認するには、`ls`
コマンドを使用して、マウントポイントの内容を表示します。

```
$ ls /pcmem
```

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

以下の例では、`myfiles` という名前の PCMCIA メモリーカードを、リモートシス
テム `mars` からローカルシステムの `/pcmem` ディレクトリにマウントしています。

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

▼ ローカルの PCMCIA メモリーカードを他のシステムで使用可能にする方法

システムを設定して、その PCMCIA メモリーカードをエクスポートすることができます。つまり、これらのドライブ上の PCMCIA メモリーカードを、他のシステムが使用できるようになります。PCMCIA メモリーカードドライブがエクスポートされると、他のシステムは、247ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」に説明されているように、それらをマウントするだけでそこに含まれる PCMCIA メモリーカードにアクセスできます。

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

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

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

3. 以下のうちの該当する手順に進みます。

| 条件 | 次の手順 |
|-------------------------------|-------------|
| <code>nfsd</code> が実行されている場合 | 250ページの手順 8 |
| <code>nfsd</code> が実行されていない場合 | 249ページの手順 4 |

4. `nfsd` がエクスポートするダミーディレクトリを作成します。

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

dummy-dir

たとえば、*dummy* などの任意のディレクトリ名にすることができ。このディレクトリには、ファイルは含まれない。これは、NFS デーモンを「呼び起こして」、エクスポートされた PCMCIA メモリーカードを認識させることを目的としている。

5. 次のエントリを `/etc/dfs/dfstab` に追加します。

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

NFS デーモンを起動すると、このエントリを参照して、エクスポートされた PCMCIA メモリーカードドライブを認識します。コメント (`-d` が前に付く) はオプションです。

6. NFS デーモンを起動します。

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

7. NFS デーモンが実際に実行されていることを確認します。

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

8. 現在ドライブ内にある PCMCIA メモリーカードを取り出します。

```
# eject pcmem0
```

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

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

10. 次の行を `/etc/rmmount.conf` に追加します。

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

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

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

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

この手順により、ファイルはそのデフォルトのアクセス権に戻ります。

12. **PCMCIA** メモリーカードをロードします。

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

ここでロードした PCMCIA メモリーカードは他のシステムで使用できるようになります。PCMCIA メモリーカードにアクセスするために、リモートユーザーは、247ページの「他のシステム上の PCMCIA メモリーカードにアクセスする方法」の指示に従って、名前によりその PCMCIA メモリーカードをマウントする必要があります。

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

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

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

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

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

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

ボリューム管理の動作の概要

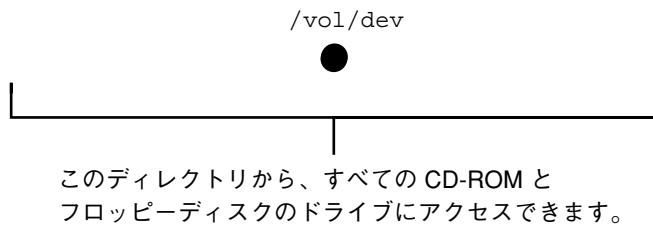
この章では、ボリューム管理が取り外し可能な媒体を格納するために作成するマウントポイントとシンボリックリンクについて説明します。

この章で説明する手順は次のとおりです。

- 253ページの「ボリューム管理が取り外し可能な媒体すべてをマウントする」
- 254ページの「フロッピーディスクへのアクセス」
- 255ページの「CD へのアクセス」
- 256ページの「アクセスが容易なマウントポイント」
- 258ページの「最終的に作成される 2 セットのシンボリックリンク」
- 260ページの「UFS フォーマットによる制限」

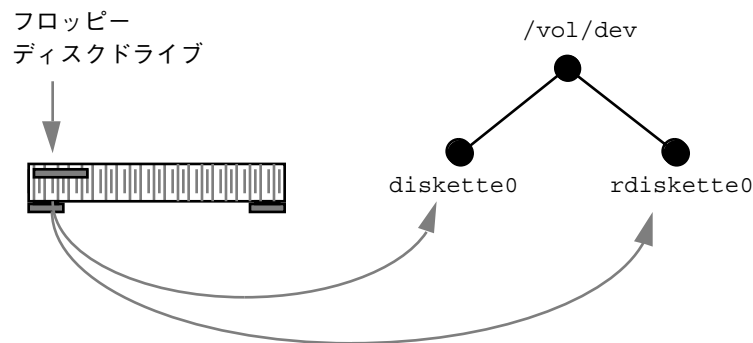
ボリューム管理が取り外し可能な媒体すべてをマウントする

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



フロッピーディスクへのアクセス

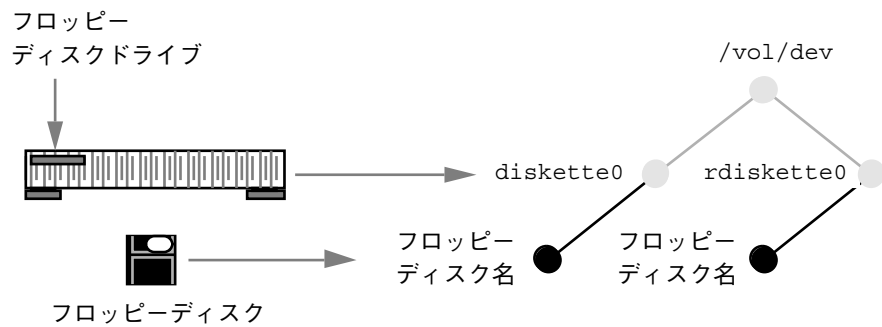
ボリューム管理は、/vol/dev のサブディレクトリ、つまり diskette0 と rdiskette0 から、システムのフロッピーディスクドライブへのアクセスを提供します。



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

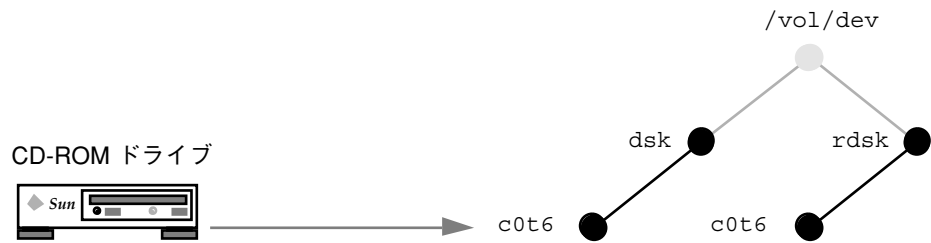
diskette ディレクトリからはファイルシステムに、rdiskette ディレクトリからは raw データに、それぞれアクセスできます。フロッピーディスク自体は、ドライブディレクトリの下にあるサブディレクトリに示されます。¹

1. この章の図では、ノードのいくつかは説明をわかりやすくするためにグレーで示されています。これは構造上の違いを示すものではありません。



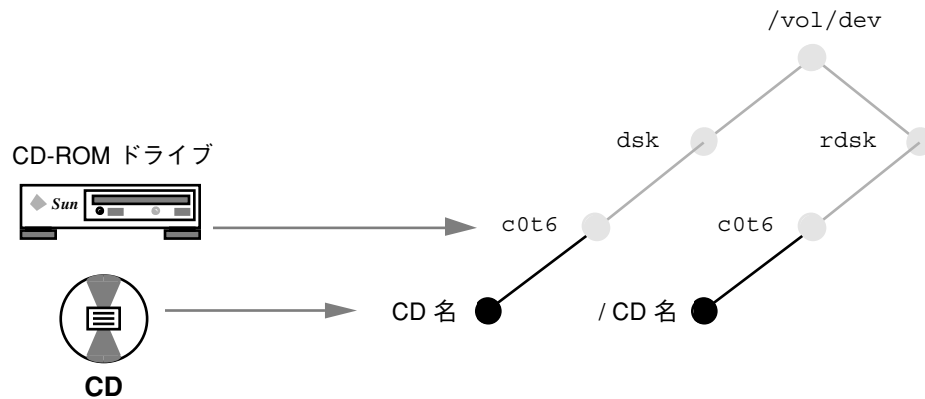
CD へのアクセス

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

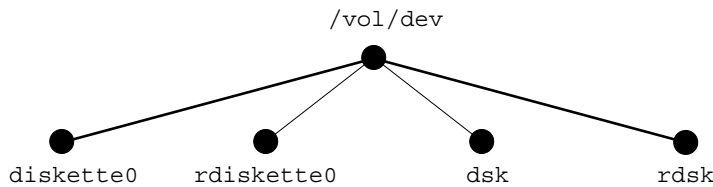


上記の図では、そのディレクトリは c0t6 と命名されています。これは、特定システムのデバイス命名規則を反映しています。このため、使用するシステムによっては、ディレクトリが同じ形式であってもディレクトリ名が異なる場合があります。

ただし、CD 自体は、それらのデバイスに属するディレクトリの下にマウントされるという点で、フロッピーディスクと似た規則に従っています。



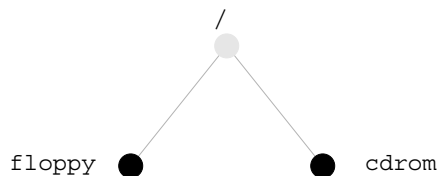
したがって、1つのフロッピーディスクドライブと1つのCD-ROMドライブを備えたシステムの `/vol/dev` ファイルシステムは、次のようになります。



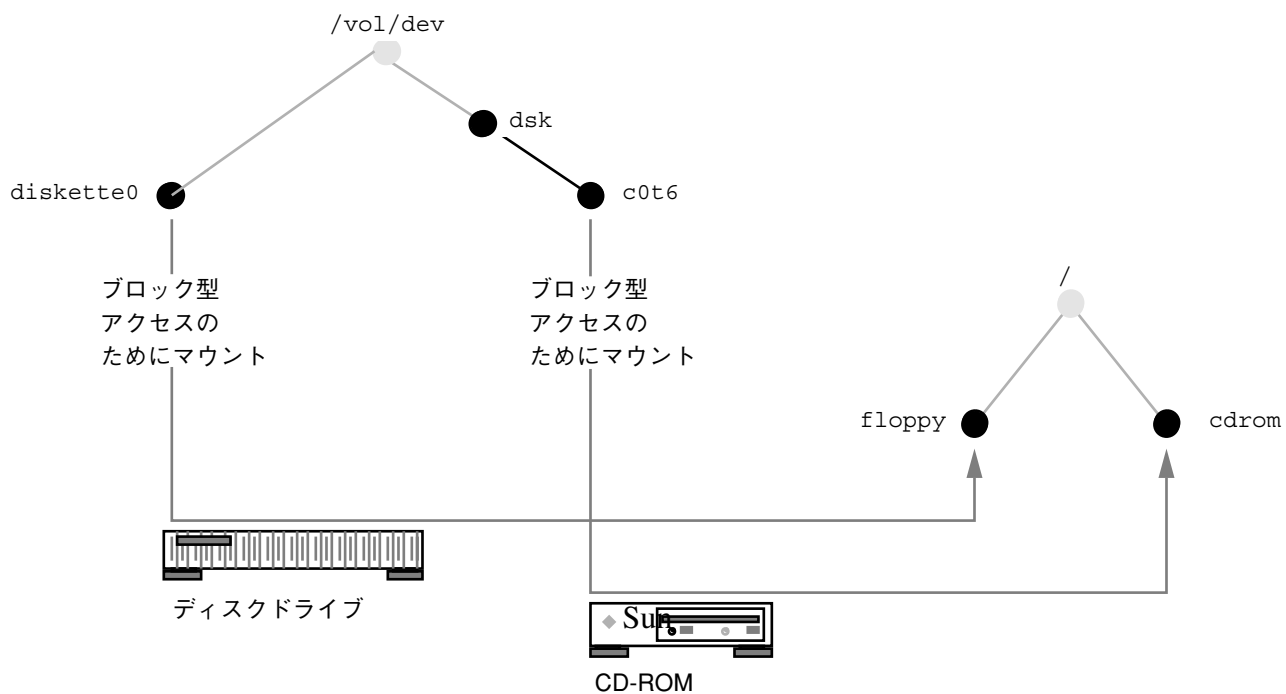
(実際には、`/vol/dev` には、`aliases` という名前のサブディレクトリが含まれますが、これについては後で説明します。)

アクセスが容易なマウントポイント

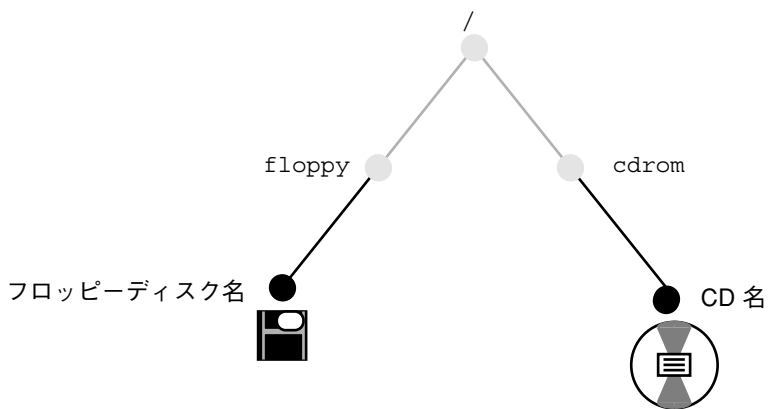
アクセスをより容易にするために、ボリューム管理は、`/floppy` および `/cdrom` という2つの特殊なマウントポイントを使用します。



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



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



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

でしか使用できません。フロッピーディスクは、208ページの「UFS フロッピーディスクをフォーマットする方法」または 213ページの「DOS フロッピーディスクをフォーマットする方法」の説明に従ってフォーマットできます。

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

最終的に作成される 2 セットのシンボリックリンク

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

- ファイルシステムアクセス用
- raw デバイスアクセス用

ファイルシステムアクセス用シンボリックリンク

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

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

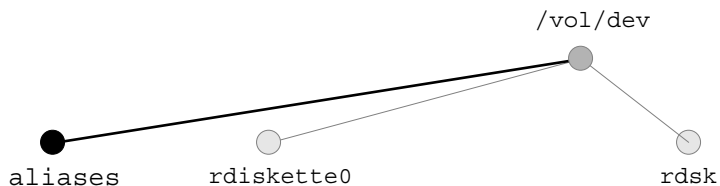
```
/cdrom/cdrom0 --> /cdrom/name --> /vol/dev/dsk/c0t6/CD-name
```

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

後続のドライブに挿入されたフロッピーディスクと CD は、表 11-3 に要約した命名規則に従います。

raw デバイスアクセス用シンボリックリンク

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



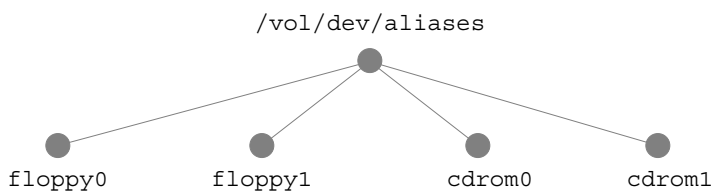
ボリューム管理は、`aliases` ディレクトリの下に、ブロック型アクセス用に使用されるものに似たシンボリックリンクのセットを作成します。つまり、raw データをアクセスする場合、これらのディレクトリは同等にあつかえます。

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

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

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

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



UFS フォーマットによる制限

UFS フォーマットは、アーキテクチャが異なると互換性がないため、それらがフォーマットされたアーキテクチャ上で使用する必要があります。たとえば、SPARC システム用にフォーマットされた UFS CD は、x86 システムでは認識されません。同様に、x86 UFS CD は、ボリューム管理によって SPARC システム上にマウントすることはできません。同じ制限が、フロッピーディスクにもあてはまります。

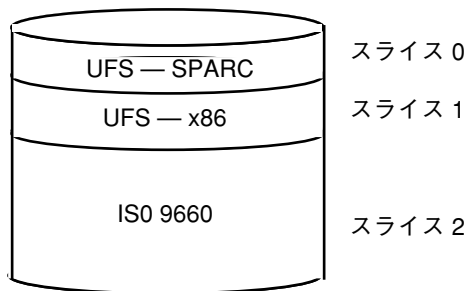
したがって、ボリューム管理は、SPARC システム上の x86 UFS 媒体を認識してマウントすることはできません。また、x86 システム上の SPARC UFS 媒体も認識できません。

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

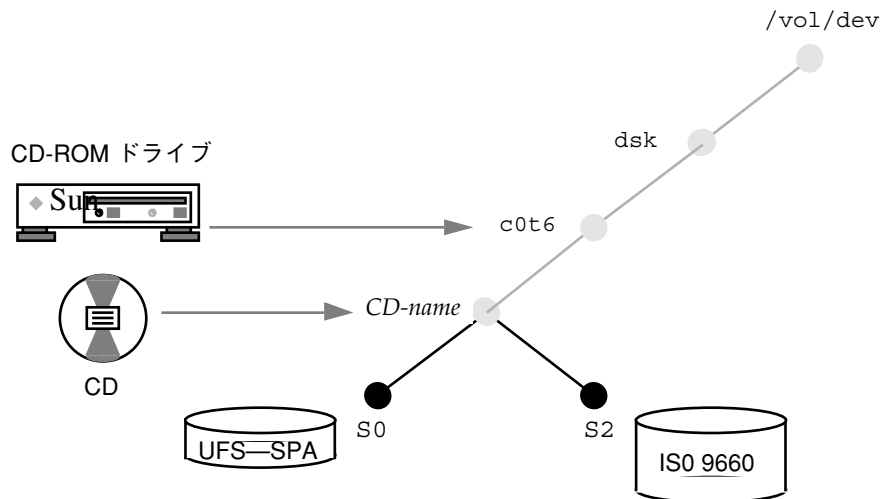
フロッピーディスクの場合、UFS の互換性の問題は、CD の場合よりも頻繁に起こる可能性があります。これは、ユーザーがフォーマットを設定できるためです。あるアーキテクチャで UFS フロッピーディスクをフォーマットした場合、それを別のアーキテクチャでは使用できないことに注意してください。(208ページの「UFS フロッピーディスクをフォーマットする方法」を参照してください。)

混合 (ハイブリッド) フォーマットについて

いくつかの CD、特にインストール用 CD には、混合フォーマットが含まれます。つまり、一部が UFS で、一部 ISO 9660 です。異なるフォーマットを格納するために、CD は、実質的にハードディスクのパーティションに似たスライスに分割されます。9660 部分は移植可能ですが、UFS 部分はアーキテクチャに固有です。さらに、CD をいくつかの異なるアーキテクチャで使用できるようにするために (異なる PROM アーキテクチャがシステムのブートに使用される可能性のあるインストールの場合などのために)、複数の UFS フォーマットが CD にロードされます。



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



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

```

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

```


パート **V** ソフトウェアの管理

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

第 16 章

Solaris 環境におけるソフトウェア製品の追加と削除についての概要を説明します。

第 17 章

クライアントのタイプ別に、ソフトウェアパッケージをインストールしたり削除したりする手順について説明します。

第 18 章

Solaris 環境でのパッチの追加と削除についての概要と手順を説明します。



ソフトウェア管理の概要

ソフトウェアの管理には、スタンドアロンシステム、サーバー、およびそのクライアントへのソフトウェアのインストール、また、ソフトウェアの削除が含まれます。この章では、ソフトウェアのインストールと管理についての概要を示します。ここでは Solaris ソフトウェアのインストールについては説明しません。

この章の内容は次のとおりです。

- 265ページの「ソフトウェア管理についての参照先」
- 266ページの「ソフトウェアパッケージ」
- 266ページの「ソフトウェア管理ツール」
- 267ページの「パッケージの追加または削除時の動作」
- 268ページの「パッケージを追加または削除する前に知っておくこと」
- 269ページの「クライアントソフトウェア管理に関するガイドライン」
- 272ページの「パッケージの削除に関するガイドライン」
- 272ページの「パッケージ追加時のユーザーの対話操作を省略する」

ソフトウェア管理についての参照先

ソフトウェアを管理する手順については、次を参照してください。

- 第 17 章
- 第 18 章

- ソフトウェア管理に関する問題の解決については、『Solaris のシステム管理 (第 2 巻)』の「ソフトウェア管理の問題の解決」を参照してください。

ソフトウェアパッケージ

ここで説明するソフトウェア管理には、ソフトウェア製品のインストールと削除が含まれます。Sun およびそのサードパーティベンダーは、ソフトウェアパッケージと呼ばれる形式で製品を提供しています。(パッケージという用語は一般に、ソフトウェア製品が使用されるシステムに、その製品を配布し、インストールする方式を指します。) パッケージの最も単純な形式としては、定義済みフォーマットのファイルとディレクトリの集合を考えることができます。このフォーマットは、アプリケーションバイナリインタフェース (ABI) に準拠します。この ABI は、System V インタフェース定義を補足するものです。Solaris オペレーティング環境には、このフォーマットを解釈して、パッケージをインストールまたは削除したり、インストールを検査する方法を提供するユーティリティがあります。

ソフトウェア管理ツール

システムにソフトウェアを追加したり、システムから削除するために、次の 2 つのツールがあります。

- pkgadd コマンドと pkgrm コマンド
- Admintool™ (システム管理ツール)

これらのどちらを使用してもかまいませんが、それぞれに長所を持っています。

pkgadd コマンドと pkgrm コマンドには柔軟性があります。たとえば、これらのコマンドをスクリプトに組み込んだり、オプションのファイルを設定してユーザーの対話操作や特別なチェック作業をなくしたり、また、ソフトウェアパッケージをスプールディレクトリにコピーすることができます。すでに pkgadd コマンドと pkgrm コマンドによるパッケージの追加と削除に詳しいユーザーの場合は、これらのコマンドを引き続き使用した方が簡単です。

Admintool を使うとソフトウェアの追加、削除が容易になります。Admintool は pkgadd コマンドと pkgrm コマンドのグラフィカルインタフェースであり、オンラインヘルプが組み込まれています。Admintool のグラフィカルブラウザは、すで

にシステムにインストールされているソフトウェアや、インストール用媒体に入っているソフトウェアを表示するのに特に適しています。ソフトウェアパッケージの命名規則やコマンド行オプションの使用法にまだ慣れていないユーザー、また、一度に1つのシステムでしかソフトウェアの管理を行わないユーザーの場合は、Admintool を使用して、ソフトウェアの追加、削除を行うことをお勧めします。

表 16-1 は、ソフトウェアの管理に、Admintool を使用した場合と、pkgadd コマンドと pkgrm コマンドを使用した場合を比較したものです。

表 16-1 Admintool のソフトウェア管理機能

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

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

パッケージの追加または削除時の動作

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

パッケージを追加する際、pkgadd ソフトウェアは、ファイルを解凍して、インストール用媒体からローカルシステムのディスクにコピーします。パッケージを削除する際、pkgrm コマンドは、そのパッケージに関連するファイルが他のパッケージと共有されている場合を除いて、それらをすべて削除します。

パッケージファイルはパッケージフォーマットで配布され、配布されたままの状態では使用できません。pkgadd コマンドは、ソフトウェアパッケージの制御ファイルを解釈してから、製品ファイルを解凍して、システムのローカルディスクにインストールします。

pkgadd コマンドと pkgrm コマンドは、標準の場所にそのログ出力を記録しませんが、インストールまたは削除された製品について追跡を行います。pkgadd と pkgrm は、インストールまたは削除されたパッケージに関する情報をソフトウェア製品データベースに格納します。

このデータベースを更新することにより、pkgadd および pkgrm の各コマンドは、システムにインストールされたすべてのソフトウェア製品の記録を追跡します。

パッケージを追加または削除する前に知っておくこと

システムでパッケージのインストールまたは削除を行うにあたっては、次のことを知っておく必要があります。

- パッケージの命名規則 – Sun パッケージは、SUNWvolr、SUNWadmap、SUNWab2m などのように、必ず接頭辞 SUNW で始まります。サードパーティのパッケージは、通常、その会社を表す接頭辞で始まります。
- インストール済みのソフトウェア – pkginfo コマンドを使用すると、システムにインストール済みのソフトウェアを表示できます。あるいは、Admintool を使用して、インストール済みのソフトウェアを表示することもできます。
- サーバーとクライアントによるソフトウェアの共有の状態 – クライアントのソフトウェアは、一部がサーバーに、一部がクライアントに置かれる場合があります。このような場合、クライアントにソフトウェアを追加するには、サーバーとクライアントの両方にパッケージを追加する必要があります。(269ページの「クライアントソフトウェア管理に関するガイドライン」で、クライアントソフトウェアの管理方法について詳しく説明します。)

クライアントソフトウェア管理に関するガイドライン

スタンドアロンシステムでのソフトウェアの管理は、パッケージのインストール用ツールと規則を理解すれば、かなり簡単なものです。ソフトウェアパッケージをシステムのローカルディスクにインストールするだけで、そのソフトウェアを使用できるようになります。しかし、クライアントシステム上でソフトウェアを管理する方法は、これよりも難しくなります。ソフトウェアがサーバーとクライアントに分かれて置かれる場合には特に困難です。(たとえば、1つのソフトウェアに、クライアントのルートファイルシステムにインストールされるファイルを持つパッケージと、クライアントが通常サーバーからマウントする `/usr` ファイルシステムにインストールされるファイルを持つパッケージがある場合などです。)

Solaris は、ディスクレスクライアントと Solstice AutoClient システムをサポートしています。ディスクレスクライアントと AutoClient システムでは、すべてのソフトウェアがサーバー上に置かれます。たとえば、ソフトウェアパッケージをディスクレスクライアントに追加する場合、ローカルディスクに記憶デバイスがないため、実際にはそのクライアントにパッケージをインストールしません。代わりに、サーバー、またはクライアントのルートファイルシステム (サーバー上にある)、あるいはこの両方にパッケージを追加します。ディスクレスまたは AutoClient システムのルートファイルシステムは、通常、サーバー上の `/export/root/hostname` です。

AutoClient システムにはディスク領域がありますが、これはキャッシュのためにのみ使用されます。ソフトウェアはサーバー上に置かれます。(詳細は『Solstice AutoClient 2.1 管理者ガイド』を参照してください。)

ディスクレスクライアントと AutoClient システムは、そのルートファイルシステムとサーバーの `/usr` (または他の共有ファイルシステム) に分かれてインストールされる場合があるため、ソフトウェアパッケージをこれらのクライアントに追加するときには、ソフトウェアパッケージがどのファイルシステムにインストールされるかを知る必要があります。

Sun パッケージをサーバーおよびクライアントへインストールする

ディスクレスクライアントと AutoClient システムにパッケージを追加する場合は、これらのパッケージのファイルがどこにインストールされるかを知る必要があります。

す。つまり、クライアントのルートファイルシステムか、サーバーの /usr ファイルシステム (またはクライアントと共有される他のいずれかのファイルシステム) のどちらかです。

Sun ソフトウェアパッケージの多くは、そのインストール場所を示すように命名されています。たとえば、SUNWvolr パッケージはルートファイルシステムにインストールされ、SUNWvolu パッケージは /usr ファイルシステムにインストールされます。「r」という接尾辞はルートを、「u」という接尾辞は /usr を表します。しかし、Sun パッケージのファイルのインストール場所を判断する最も確実な方法は、SUNW_PKGTYPE パラメータを調べることです。これは、パッケージの pkginfo ファイルに設定されています。pkginfo ファイルを調べるためのインタフェースについては、285ページの「パッケージファイルのインストール場所を決定する方法」で説明します。

Sun パッケージの一部には、対応する SUNW_PKGTYPE パラメータがありません。これらのパッケージは、通常、/opt にインストールされるように設定されています。Sun パッケージに SUNW_PKGTYPE パラメータ値がないときは、それをインストールするときに、サードパーティパッケージとして扱ってください。(詳細については、270ページの「サードパーティパッケージをサーバーおよびクライアントへインストールする」を参照してください。)

ディスクレスクライアントまたはデータレスクライアントに Sun パッケージをインストールするときは、表 16-2 の一般的なガイドラインに従ってください。

表 16-2 クライアントへの Sun パッケージのインストール

| パッケージファイルのインストール場所 | パッケージのインストール場所 |
|-----------------------------|-----------------------------|
| ルート (/) ファイルシステム | クライアントまたはクライアントのルートファイルシステム |
| /usr (または他のいずれかの共有ファイルシステム) | サーバー |

サードパーティパッケージをサーバーおよびクライアントへインストールする

サードパーティパッケージは、SUNW_PKGTYPE パラメータを使用しません。このため、パッケージファイルのインストール場所を判断する簡単な方法はありません。

最も確実な方法は、パッケージの `pkgmap` ファイルを調べることです。この情報に基づいて、表 16-2 のガイドラインに従ってインストールを行うことができます。ただし、パッケージの `pkgmap` ファイルを調べる手間を省きたい場合は、次の手順に従ってください。これは、サードパーティパッケージをディスククライアントおよび `AutoClient` システムにインストールするための最も安全な方法です。

1. サーバーにソフトウェアをインストールします。サーバーがクライアントと共有するすべてが更新されます。(これは、サーバーとクライアントが、同じバージョンの `Solaris` ソフトウェアを実行し、同じハードウェアプラットフォームで実行されていることを前提としています。たとえば、両方とも `x86` システムであるか、両方とも `SPARC` システムであるなどです。)
2. ソフトウェアをクライアントにインストールします。使用しているのが `pkgadd` コマンドか `Admintool` のいずれであっても、これはクライアントに適したファイルだけをインストールします。`pkgadd` コマンドまたは `Admintool` は、サーバーからマウントされたファイルシステムで、すでに存在するソフトウェアはインストールしません。これは、そのソフトウェアが、クライアントによりすでに使用可能になっているためです。

異機種環境でのパッケージのインストール

次の場合には、クライアントとサーバーにおけるソフトウェア管理がさらに複雑になります。

- サーバーが、クライアントと異なるリリースの `Solaris` を実行している場合。たとえば、サーバーが `Solaris 7` を実行していて、`Solaris 2.5` のディスククライアントにサービスを提供している場合などです。
- サーバーとクライアントのハードウェアプラットフォームが異なる場合。たとえば、サーバーが `SPARC` システムで、`x86` システムであるディスククライアントにサービスを提供している場合などです。

これらは異機種環境と呼ばれます。異機種環境のソフトウェアを管理する場合は、まずサーバーのクライアントに適した正しい `Solaris` とアーキテクチャサービスを追加する必要があります。これには、ホストマネージャを使用して、モニターに「サービスを追加」します (詳細については、第 4 章を参照してください)。

異機種環境でパッケージを追加する方法の詳細については、284ページの「異機種クライアント/サーバー環境でのパッケージの追加」を参照してください。

パッケージの削除に関するガイドライン

pkgadd コマンドと pkgrm コマンドは、ソフトウェア製品用データベース内の情報を更新するため、パッケージを削除するときは、rm コマンドではなく、pkgrm コマンドを使用する必要があります。たとえば、rm コマンドを使用すると、バイナリ実行可能ファイルを削除することができますが、これは pkgrm を使用してバイナリ実行可能ファイルを含むソフトウェアパッケージを削除する場合とは異なります。rm コマンドを使用してパッケージのファイルを削除すると、ソフトウェア製品用データベースが破壊されます。(1つのファイルだけを削除したい場合には、removef コマンドを使用してください。これは、ソフトウェア製品用データベースを正しく更新します。詳細については、removef(1M) のマニュアルページを参照してください。)

複数のバージョンのパッケージをインストールしておきたい場合は(たとえば、複数バージョンの文書処理アプリケーションなど)、新しいバージョンを、すでにインストールされたパッケージとは異なるディレクトリにインストールしてください。パッケージがインストールされているディレクトリは、ベースディレクトリと呼ばれ、このベースディレクトリは、管理ファイルと呼ばれる特殊ファイルに basedir キーワードを設定することによって操作できます。管理ファイルの使用とベースディレクトリの設定の詳細については、272ページの「パッケージ追加時のユーザーの対話操作を省略する」と admin(4) のマニュアルページを参照してください。

注 - Solaris ソフトウェアをインストールするときにアップグレードオプションを使用すると、Solaris インストール用ソフトウェアは、ソフトウェア製品用データベースを検索して、製品がすでにシステムにインストールされているかどうかを判断します。

パッケージ追加時のユーザーの対話操作を省略する

管理ファイルの使用

pkgadd コマンドを -a オプションを指定して実行するとき、pkgadd コマンドは、どのようにインストールを進めるかについての情報を持つ特殊な管理ファイルを参照します。通常、pkgadd はいくつかのチェックを行い、指定されたパッケージを

実際に追加する前に、プロンプトを出してユーザーに確認します。ただし、管理ファイルを作成すれば、このようなチェックを省略して、ユーザーの確認なしでパッケージをインストールするように `pkgadd` に指示できます。

デフォルトでは、`pkgadd` コマンドは現在の作業用のディレクトリで管理ファイルを探します。現在の作業用ディレクトリの中に管理ファイルを見つけることができなかつた場合、`pkgadd` は `/var/sadm/install/admin` ディレクトリで、指定された管理ファイルを探します。`pkgadd` コマンドには管理ファイルの絶対パスも使用できます。



注意 - 管理ファイルは注意して使用してください。通常は `pkgadd` が提供するチェックとプロンプトを省略するには、管理ファイルを使用する前に、パッケージのファイルがどこにインストールされているのか、および、パッケージのインストールスクリプトをどのように実行するのを知っておく必要があります。

次は、`pkgadd` がパッケージのインストール前にユーザーに確認のプロンプトを出さないようにする管理ファイルの例です。

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

パッケージを追加するときのユーザーの対話操作を省略する以外にも、いろいろな目的で管理ファイルを使用できます。たとえば、管理ファイルを使用すれば、エラーが発生した場合に、(ユーザーの対話操作なしに)パッケージのインストールを終了できます。また、`pkgrm` コマンドでパッケージを削除するときの対話を省略できます。

また、特別なインストールディレクトリをパッケージに割り当てることができます。(これは、1つのシステム上で複数のバージョンのパッケージを管理する場合に役に立ちます。)これを行うには、パッケージがインストールされる場所を指定する、代替ベースディレクトリを管理ファイルに設定します (`-basedir` キーワードを使用する)。詳細については、`admin(4)` のマニュアルページを参照してください。

応答ファイルの使用

応答ファイルには、対話型パッケージで尋ねられる特定の質問に対するユーザーの応答が含まれます。対話型パッケージには、パッケージをインストールする前にユーザーに質問する (たとえば、パッケージのオプションをインストールするかどうかなど) request スクリプトが含まれます。

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

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

ソフトウェア管理の手順

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

この章で説明する手順は次のとおりです。

- 277ページの「スタンドアロンシステムにパッケージを追加する方法」
- 280ページの「スプールディレクトリにパッケージを追加する方法」
- 285ページの「パッケージファイルのインストール場所を決定する方法」
- 286ページの「ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法」
- 289ページの「パッケージをサーバーに追加する方法」
- 294ページの「インストールされたパッケージが完全であるかどうかをチェックする方法」
- 293ページの「インストールされたすべてのパッケージに関する情報を表示する方法」
- 295ページの「パッケージに関する詳細を表示する方法」
- 296ページの「パッケージを削除する方法」
- 297ページの「スプールされたパッケージを削除する方法」
- 297ページの「ディスクレスクライアントまたは AutoClient システムのパッケージを削除する方法」
- 299ページの「Admintool を起動する方法」
- 301ページの「Admintool でパッケージを削除する方法」

ソフトウェアパッケージを処理するコマンド

表 17-1 に、ソフトウェアパッケージの追加、削除、検査、インストールに使用するコマンドを示しています。

表 17-1 パッケージの追加および削除を行うコマンド

| コマンド | 説明 |
|-------------|-------------------------|
| pkgadd(1M) | ソフトウェアパッケージをインストールする |
| pkgrm(1M) | ソフトウェアパッケージを削除する |
| pkgchk(1M) | ソフトウェアパッケージのインストールを検査する |
| pkginfo(1) | ソフトウェアパッケージ情報を表示する |
| pkgparam(1) | ソフトウェアパッケージのパラメータ値を表示する |

パッケージの追加と削除における既知の問題

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

```
NONABI_SCRIPTS=TRUE
```

パッケージの追加

▼ スタンドアロンシステムにパッケージを追加する方法

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

これによってシステムは、追加および削除されたソフトウェアの正しい記録を保持できます。同じアプリケーションの複数のバージョンをシステムで管理したい場合もあります。この方法の概要については 272 ページの「パッケージの削除に関するガイドライン」を、手順については 296 ページの「パッケージを削除する方法」を参照してください。

3. ソフトウェアパッケージをシステムに追加します。

```
# pkgadd -a admin-file -d device-name pkgid...
```

-a admin-file (省略可能) インストール時に `pkgadd` が参照する管理ファイル指定する。(管理ファイルの使用法についての詳細は、272 ページの「管理ファイルの使用」を参照。)

-d device-name ソフトウェアパッケージの絶対パスを指定する。*device-name* は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。パッケージのあるパスを指定しないと、`pkgadd` コマンドはデフォルトのスプールディレクトリ (`/var/spool/pkg`) をチェックする。パッケージがそこがない場合、パッケージのインストールは失敗する。

pkgid (省略可能) インストールされる 1 つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、`pkgadd` コマンドは、使用可能なすべてのパッケージをインストールする。

パッケージのインストール中に `pkgadd` に問題が起こると、その問題に関連するメッセージに続いて、次のプロンプトが表示されます。

```
Do you want to continue with this installation?
```

このプロンプトには、yes、no、または quit のいずれかで応答します。複数のパッケージが指定されている場合は、no と入力して、インストール中のパッケージのインストールを中止してください。pkgadd は、他のパッケージのインストールを続けます。インストールを中止するには、quit と入力してください。

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

```
# pkgchk -v pkgid
```

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

例 — マウントされた CD からソフトウェアをインストールする

注 - 今回の Solaris リリースの名称は「Solaris 7」ですが、コード、パス名、パッケージパス名などには、「Solaris 2.7」または「SunOS 5.7」という名称が使用されていることがあります。コード、パス、パッケージパスなどを実際に入力または使用するときには、必ずマニュアル中に記述されている名称に従ってください。

次の例は、SUNWaudio パッケージをマウントされた Solaris 7 CD からインストールするコマンドを示しています。この例では、pkgchk コマンドを使用して、パッケージファイルが正しくインストールされたかどうかを検査する方法も示しています。

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_2.7/Product SUNWaudio
.
.
.
Installation of <SUNWaudio> complete.
# pkgchk -v SUNWaudio
/usr
/usr/bin
/usr/bin/audioconvert
/usr/bin/audioplay
/usr/bin/audiorecord
```


例 — リモートパッケージサーバーからソフトウェアをインストールする

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

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWaudio
.
.
.
Installation of <SUNWaudio> was successful.
```

オートマウンタがサイトで実行されている場合は、リモートパッケージサーバーを手作業でマウントする必要はありません。代わりに、オートマウンタパス (この場合は、`/net/package-server/latest-packages`) を、`-d` オプションの引数として使用してください。

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

次の例は、前の例に似ていますが、`-a` オプションを使用して `noask-pkgadd` という名前の管理ファイルを指定している点が異なります。このファイルは 272 ページの「パッケージ追加時のユーザーの対話操作を省略する」に示されています。この例では、`noask-pkgadd` 管理ファイルが、デフォルト位置の `/var/sadm/install/admin` にあるものと想定しています。

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

スプールディレクトリの使用方法

使用頻度の高いインストール済みパッケージを、スプールディレクトリにコピーすると便利です。パッケージをデフォルトのスプールディレクトリである `/var/spool/pkg` にコピーする場合、`pkgadd` コマンドを使用するときに、パッケージのソース位置 (`-d device-name` 引数) を指定する必要はありません。`pkgadd` コマンドは、デフォルトにより、`/var/spool/pkg` ディレクトリを探して、コマンド行に指定されたすべてのパッケージを見つけます。パッケージをスプールディレクトリにコピーすることと、パッケージをシステム上にインストールすることとは異なることに注意してください。

▼ スプールディレクトリにパッケージを追加する方法

1. サーバーまたはスタンドアロンシステムのスーパーユーザーとしてログインします。
2. すでにスプールされているパッケージの中で、追加しようとしているものと同じ名前のパッケージをすべて削除します。
スプールされたパッケージを削除する方法については、297ページの「スプールされたパッケージを削除する方法」を参照してください。
3. ソフトウェアパッケージをスプールディレクトリに追加します。

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

| | |
|-----------------------------|--|
| <code>-d device-name</code> | ソフトウェアパッケージの絶対パスを指定する。 <code>device-name</code> は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。 |
| <code>-s spooldir</code> | パッケージがスプールされるスプールディレクトリの名前を指定する。 <code>spooldir</code> を指定する必要がある。 |
| <code>pkgid</code> | (省略可能) インストールされる 1 つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、 <code>pkgadd</code> は、使用可能なすべてのパッケージをコピーする。 |

4. パッケージがスプールディレクトリに正常にコピーされたことを確認するには、`pkginfo` コマンドを使用します。

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

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

例 — マウントされた CD からスプールディレクトリを設定する

次の例は、`SUNWaudio` および `SUNWab2m` の各パッケージを、マウントされた SPARC 版 Solaris 7 CD からデフォルトのスプールディレクトリ (`/var/spool/pkg`) にコピーするコマンドを示しています。

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_2.7/Product -s /var/spool/pkg SUNWaudio SUNWab2m
Transferring <SUNWaudio> package instance
Transferring <SUNWab2m> package instance
```

例 — リモートパッケージサーバーからスプールディレクトリを設定する

インストールしたいパッケージがリモートシステムから利用できるパッケージの場合は、そのパッケージを (パッケージの形で) 含むディレクトリを手作業でマウントして、ローカルスプールディレクトリにそれをコピーすることができます。次の例は、このためのコマンドを示しています。この例では、`package-server` という名前のリモートシステムが、`/latest-packages` ディレクトリにソフトウェアパッケージを持つものと想定しています。`mount` コマンドは、パッケージを `/mnt`

にローカルにマウントし、pkgadd コマンドは、SUNWman パッケージを /mnt からデフォルトのスポールディレクトリ (/var/spool/pkg) にコピーします。

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

オートマウンタがサイトで実行されている場合は、リモートパッケージサーバーを手作業でマウントする必要はありません。代わりに、オートマウンタパス (この場合は、/net/package-server/latest-packages) を、-d オプションの引数として使用してください。

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

例 — デフォルトのスポールディレクトリからパッケージをインストールする

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

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

同機種クライアント/サーバー環境でのパッケージの追加

この説明では、「同機種クライアント/サーバー」とは、同じバージョンの Solaris を実行していて、ハードウェアプラットフォーム (すべて SPARC か、またはすべて x86 プラットフォーム) が同じであるクライアントとサーバーを意味します。

この節では、ファイルをクライアントのルートファイルシステムに置くパッケージのインストール方法を説明します。クライアントにパッケージをインストールする

ときに、そのパッケージがファイルをクライアントのルートファイルシステムに置かない場合、そのパッケージは直接サーバー上にインストールして共有することができます。(これは、パッケージがサーバーの /usr などのファイルシステムにインストールされる場合を想定しています。)

pkgadd コマンドに `-R` オプションを付けて使用して、クライアントインストールのためのクライアントルートファイルシステムの位置を指定してください。(`-R` オプションを使用すると、パッケージインストール用の代替ベースディレクトリを指定できるという一般的な誤解がありますが、これは間違いです。 `-R` オプションは、クライアントのルートファイルシステムを定義するものです。代替ベースディレクトリを指定するには、 `-a` オプションを付けて `pkgadd` を使用し、 `basedir` キーワードが新しいインストールディレクトリに設定されている管理ファイルを提供してください。)

注 - ディスクレスクライアントまたは **AutoClient** システムのサーバーにインストールされたパッケージは、そのクライアントと他のクライアントに読み取り専用で共有されます。

クライアント/サーバー環境でパッケージをインストールして管理する方法はいくつかありますが、この項では、サーバーからのインストール方法について説明します。これはソフトウェアを集中管理する方法です。ただし、クライアントにログインして、直接それらにソフトウェアをインストールできることに注意してください。

クライアントへの **Sun** パッケージの追加

通常、**Sun** パッケージを同機種環境のクライアントにインストールする場合は、表 17-2 に示すガイドラインに従ってください。

表 17-2 同機種環境のクライアントへの Sun パッケージのインストール

| パッケージファイルのインストール場所 | 手順 |
|--------------------|---|
| ルート (/) ファイルシステム | 286ページの「ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法」の手順を使用してパッケージを追加する。 |
| /usr | 277ページの「スタンドアロンシステムにパッケージを追加する方法」の手順を使用してパッケージを追加する。 |

285ページの「パッケージファイルのインストール場所を決定する方法」の手順を使用すると、Sun パッケージファイルのインストール場所を調べることができます。

クライアントへのサードパーティパッケージの追加

サードパーティパッケージをクライアントに追加する場合は、次のガイドラインに従ってください。

1. 277ページの「スタンドアロンシステムにパッケージを追加する方法」の手順に従って、サーバーにパッケージをインストールします。
2. 286ページの「ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法」のいずれか適切な手順に従って、クライアントにパッケージをインストールします。

異機種クライアント/サーバー環境でのパッケージの追加

この説明では、「異機種クライアント/サーバー環境」とは、クライアントとサーバーが異なるバージョンの Solaris を実行しているか、またはハードウェアプラットフォームが異なる (たとえば、Solaris 7 クライアントの Solaris 2.3 サーバーや、SPARC クライアントを持つ x86 サーバーなど) 場合を示します。異機種クライアント/サーバー環境へパッケージを追加することは、特有の困難が伴います。サーバーには、それがサポートする異機種クライアントのための複数の /usr ファイルシ

テムがあります。たとえば、x86 システム用の x86 /usr ファイルシステム、Solaris 2.4 クライアント用の Solaris 2.4 /usr ファイルシステムなどです。通常、異機種クライアント/サーバー環境にパッケージをインストールする場合は、表 17-3 のガイドラインに従ってください。

表 17-3 異機種環境へのパッケージのインストール

| パッケージファイルのインストール場所 | 手順 |
|--------------------|---|
| ルート (/) ファイルシステム | 286ページの「ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法」の手順を使用してパッケージを追加する。 |
| /usr | 289ページの「パッケージをサーバーに追加する方法」の手順を使用してパッケージを追加する。 |

▼ パッケージファイルのインストール場所を決定する方法

この手順は、Sun ソフトウェアパッケージにのみ有効です。サードパーティソフトウェア製品の場合、パッケージファイルのインストール場所を判断する最も確実な方法は、pkgmap ファイルでパッケージのディレクトリを調べる方法です。

1. 任意のシステムにログインします。
パッケージがあるディレクトリにアクセスできなければなりません。
2. Sun パッケージファイルのインストール場所を指定します。

```
$ pkgparam -d device-name pkgid SUNW_PKGTYPE
```

| | |
|-----------------------------|--|
| <code>-d device-name</code> | ソフトウェアパッケージの絶対パスを指定する。 <code>device-name</code> は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。 <code>-d</code> オプションを使用しないと、 <code>pkgparam</code> は、ローカルシステムにインストールされた指定の <code>pkgid</code> のデフォルトのインストール用ディレクトリを返す。 |
| <code>pkgid</code> | ソフトウェアパッケージの名前 |
| <code>SUNW_PKGTYPE</code> | Solaris ソフトウェアパッケージのインストール場所について報告する特殊パラメータを指定する。パッケージに <code>SUNW_PKGTYPE</code> パラメータが設定されていないと、 <code>pkgparam</code> コマンドは、空の文字列を返す。Sun パッケージの場合、これは通常、パッケージが <code>/opt</code> にインストールされることを示す。 |

例 — パッケージファイルのインストール場所を決定する

```

$ pkgparam -d /cdrom/cdrom0/s0/Solaris_2.7 /Product SUNWvolr SUNW_PKGTYPE
root
$ pkgparam -d /cdrom/cdrom0/s0/Solaris_2.7/Product SUNWvolu SUNW_PKGTYPE
usr

```

▼ ディスクレスクライアントまたは AutoClient システムのルート (/) ファイルシステムにパッケージを追加する方法

ディスクレスクライアントまたは AutoClient システムにパッケージを追加する場合、クライアントはローカルにはシステムディスクしか持たないため、実際にはそのクライアントにパッケージをインストールすることにはなりません。代わりに、サーバー上にあるクライアントのルートファイルシステムにパッケージを追加します。ディスクレスクライアントまたは AutoClient システムのルートファイルシステムは通常、サーバーの `/export/root/hostname` にあります。

注 - パッケージファイルが /usr ファイルシステムにインストールされる場合は、パッケージをサーバーにインストールする必要があります。同機種クライアント/サーバー環境で作業している場合は、表 17-2 を参照して、パッケージのインストール方法を判断してください。異機種クライアント/サーバー環境で作業している場合は、表 17-3 を参照して、パッケージのインストール方法を決定してください。

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

これによって、システムは、追加および削除されたソフトウェアの正しい記録を保持できます。同じアプリケーションの複数のバージョンをシステムで管理したい場合もあります。この方法の概要については 272 ページの「パッケージの削除に関するガイドライン」を、手順については 297 ページの「ディスクレスクライアントまたは AutoClient システムのパッケージを削除する方法」を参照してください。

3. ソフトウェアパッケージをクライアントシステムのルート (/) ファイルシステムに追加します。

```
server# pkgadd -R rootpath -d device-name pkgid...
```

-R rootpath クライアントのルートファイルシステムのパス名を指定する。

-d device-name ソフトウェアパッケージの絶対パスを指定する。*device-name* は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。パッケージのあるパスを指定しないと、*pkgadd* コマンドはデフォルトのスプールディレクトリ (*/var/spool/pkg*) をチェックする。パッケージがそこになければ、パッケージのインストールは失敗する。

pkgid (省略可能) インストールされる 1 つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、*pkgadd* は、使用可能なすべてのパッケージを表示する。



注意 - パッケージのインストール中に、次のメッセージが表示されることがあります。WARNING: filename <not present on Read Only file system>

これは、パッケージのファイルの一部がインストールされていないことを示します。クライアントが、ソフトウェアを正しく動作させるために必要なすべてのファイルへのアクセスをもっていない可能性があります。

4. パッケージが正常にインストールされたことを確認するには、サーバーにログインしてスーパーユーザーになり、`pkginfo` コマンドを使用します。

```
server# pkginfo -R rootpath | egrep pkgid
```

`pkginfo` コマンドは、インストールされた `pkgid` に関する 1 行の情報を返します。 `pkgid` がインストールされていない場合、`pkginfo` は何も表示しません。

5. `pkgchk` コマンドを使用して、パッケージが正常にインストールされていることを確認します。

```
server# pkgchk -R rootpath -v pkgid
```

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

例 — マウントされた CD からディスクレスクライアントのルートファイルシステムへパッケージをインストールする

注 - 今回の Solaris リリースの名称は「Solaris 7」ですが、コード、パス名、パッケージパス名などには、「Solaris 2.7」または「SunOS 5.7」という名称が使用されていることがあります。コード、パス、パッケージパスなどを実際に入力または使用する際には、必ずマニュアル中に記述されている名前に従ってください。

次の例は、サーバーからディスクレスクライアントのルートファイルシステムに SUNWadmr (システムおよびネットワーク管理をサポートするソフトウェア) パッケージをインストールするためのコマンドを示しています。この場合、ディスクレスクライアントのルートファイルシステムは、`/export/root/client-1` です。この例では、SUNWadmr パッケージが、マウントされた SPARC 版 Solaris CD

(/cdrom/cdrom0/s0/Solaris_2.7/Product) から使用できるものと想定しています。pkginfo と pkgchk を使用して、パッケージのファイルが正しくインストールされていることを確認する方法も示しています。

```
server# pkgadd -R /export/root/client-1 -d /cdrom/cdrom0/s0/Solaris_2.7/Product SUNWadmr
.
.
Installation of <SUNWadmr> complete.
server# pkginfo -R /export/root/client-1 | egrep SUNWadmr
system SUNWadmr System & Network Administration Root
server# pkgchk -v -R /export/root/client-1 SUNWadmr
/etc
/etc/init.d
/etc/init.d/autoinstall
/etc/init.d/sysid.net
/etc/init.d/sysid.sys
/etc/rc2.d
/etc/rc2.d/S30sysid.net
/etc/rc2.d/S71sysid.sys
/etc/rc2.d/S72autoinstall
/sbin
/sbin/bpgetfile
```

例 — マウントされた CD からディスククライアントのルートファイルシステムへパッケージをインストールする

次の例は、サーバーからディスククライアントのルートファイルシステムへ SUNWcg6 パッケージをインストールする方法を示しています。この場合、ディスククライアントのルートファイルシステムは /export/root/client-2 です。この例では、SUNWcg6 パッケージが、ネットワーク上のパッケージサーバー (/net/package-server/latest-packages) から使用できることを前提としています。

```
server# pkgadd -R /export/root/client-2 -d /net/package-server/latest-packages SUNWcg6
.
.
Installation of <SUNWcg6> complete.
```

▼ パッケージをサーバーに追加する方法

1. サーバーにスーパーユーザーとしてログインします。

2. サーバーに、そのディスクレスクライアントと **AutoClient** システムに必要な **OS** サービスがあることを確認します。
 ホストマネージャを使用して、サーバーで使用可能な OS サービスを確認します。OS サービスを追加したい場合は、ホストマネージャの「サービス追加」機能を使用して追加を行うことができます。詳細については、第 4 章を参照してください。
3. サーバーと、ディスクレスまたは **AutoClient** システムが同じバージョンの **Solaris** であるか、ハードウェアプラットフォームが同じかどうかによって、次の手順を決定します。

| ディスクレスまたは AutoClient システムと、サーバーの状態 | 手順 |
|---|--|
| 同じ Solaris のリリースと同じハードウェアアーキテクチャ | この手順は使用しないこと。代わりに、277ページの「スタンドアロンシステムにパッケージを追加する方法」の手順を使用する。 |
| Solaris のリリースが異なるか、またはハードウェアプラットフォームが異なる (たとえば、 Solaris 7 ディスクレスクライアントの Solaris 2.5 サーバー、または SPARC ディスクレスクライアントの x86 サーバー) | 290ページの手順 4 に進む。 |

4. デフォルト管理ファイルのコピーを作成します。

```
# cp /var/sadm/install/admin/default /var/sadm/install/admin/admin-file
```

5. 新しい管理ファイルを編集して、`basedir` キーワードを設定します。
 テキストエディタを使用して新しい管理ファイルを編集し、`basedir` キーワードを、クライアントをサポートする OS サービスへの正しいパスに設定します。

```
basedir=/export/exec/Solaris_2.7_platform.all/usr
```

| | |
|--------------------------|---|
| <code>Solaris_2.7</code> | Solaris バージョン番号を示す。たとえば、 <code>Solaris_2.7</code> など。 |
| <code>platform</code> | クライアントのハードウェアアーキテクチャーを示す。たとえば、 <code>Solaris_2.7_i386.all</code> や <code>Solaris_2.7_sparc.all</code> などにおける <code>i386</code> や <code>sparc</code> など。 |

6. ソフトウェアパッケージをサーバーに追加します。

管理ファイルは、パッケージを、クライアントに適した `/usr` ファイルシステムにインストールするように指定します。

```
# pkgadd -a admin-file -d device-name pkgid...
```

| | |
|-----------------------------|---|
| <code>-a admin-file</code> | (省略可能) <code>pkgadd</code> がインストール時に参照する管理ファイル指定する。デフォルトにより、 <code>pkgadd</code> は、 <code>/var/sadm/install/admin</code> ディレクトリを調べて、指定の管理管理ファイルを探す。絶対パスを管理ファイルに指定することもできる。 |
| <code>-d device-name</code> | ソフトウェアパッケージの絶対パスを指定する。 <code>device-name</code> は、デバイス、ディレクトリ、またはスプールディレクトリのいずれかへのパスにすることができる。パッケージのあるパスを指定しないと、 <code>pkgadd</code> はデフォルトのスプールディレクトリ (<code>/var/spool/pkg</code>) をチェックする。パッケージがそこがない場合、パッケージのインストールは失敗する。 |
| <code>pkgid</code> | (省略可能) インストールされる 1 つまたは複数のパッケージの名前を空白で区切って示す。この引数を省略すると、 <code>pkgadd</code> は、使用可能なすべてのパッケージを表示する。 |

パッケージのインストール中に `pkgadd` コマンドに問題が起こると、その問題に関連するメッセージに続いて、次のプロンプトが表示されます。

```
Do you want to continue with this installation?
```

このプロンプトには、`yes`、`no`、または `quit` のいずれかで応答します。複数のパッケージが指定されている場合は、`no` と入力して、インストール中のパッケージのインストールを中止してください。`pkgadd` は、他のパッケージのインストールを続けます。インストールを中止するには、`quit` と入力してください。

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

```
# pkginfo pkgid*
```

`pkginfo` コマンドは、インストールされているパッケージのすべてについて表示します。通常、`pkgadd` は、すでにインストールされたパッケージと重複するバージョンを、`pkgid.1`、`pkgid.2` としてインストールします。

8. `pkgchk` コマンドを使用して、パッケージが正常にインストールされていることを確認します。

```
# pkgchk -v pkgid
```

`pkgchk` は、指定されたパッケージにエラーがないと判断した場合は、インストールされたファイルのリストを返します。そうでない場合は、エラーについて報告します。

例 — マウントされた CD からソフトウェアをインストールする

次の例では、`/usr` ファイルシステムにファイルをインストールする架空のパッケージである `SUNWtoolu` をインストールするコマンドを示しています。このパッケージは、デフォルトにより `/cdrom/cdrom0` にマウントされた、マウント済み製品 CD にあるものと想定してください。`pkgadd` コマンドは、`new-basedir` という名前の管理ファイルを使用して、パッケージの新しいインストール用ディレクトリを指定します。この例は、`pkgchk` を使用して、パッケージファイルが正しくインストールされたかどうかを確認する方法も示しています。

```
# pkgadd -a new-basedir /cdrom/cdrom0 SUNWtoolu
.
.
.
Installation of <SUNWtoolu> complete.
# pkgchk -v SUNWtoolu
/usr
/usr/bin
/usr/bin/toolconvert
/usr/bin/toolplay
/usr/bin/toolrecord
```

パッケージのインストールの検査

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

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

▼ インストールされたすべてのパッケージに関する情報を表示する方法

pkginfo コマンドを使用して、インストールされたパッケージに関する情報を表示します。

```
$ pkginfo
```

例 — インストールされたすべてのパッケージを表示する

次の例は、システムがスタンドアロン、サーバー、ディスクレスクライアント、または AutoClient システムのどれであるかに関係なく、ローカルシステムにインストールされたすべてのパッケージを表示する pkginfo コマンドを示しています。出力には、基本カテゴリ、パッケージ名、およびパッケージの説明が示されています。

```
$ pkginfo
system      SUNWab2m      Solaris Documentation Server Lookup
system      SUNWaccr      System Accounting, (Root)
system      SUNWaccu      System Accounting, (Usr)
system      SUNWadmap     System administration applications
system      SUNWadmc      System administration core libraries
```

例 — ディスクレスクライアントまたは AutoClient システムにインストールされたすべてのパッケージを表示する

ディスクレスまたは AutoClient システムクライアント/サーバーの設定では、サーバーからソフトウェアを管理することができます。サーバーはこのための場所であるため、pkginfo コマンドにオプションを付けて使用する必要があります。次の例は、io という名前のディスクレスクライアントにインストールされたすべてのパッケージを表示するための pkginfo -R コマンドを示しています。このコマンドは、ディスクレスクライアントのサーバーから実行します。

```

server$ pkginfo -R /export/root/io
system  SUNWaccr  System Accounting, (Root)
system  SUNWaccu  System Accounting, (Usr)
system  SUNWadmap System & Network Administration Applications
system  SUNWadmfw System & Network Administration Framework
.
.
.

```

▼ インストールされたパッケージが完全であるかどうかをチェックする方法

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

```

# pkgchk -a | -c -v pkgid ...
# pkgchk -d spooldir pkgid ...

```

| | |
|--------------------------|---|
| <code>-a</code> | <code>pkgchk</code> のデフォルトであるファイル属性と内容ではなく、ファイル属性 (つまりアクセス権) だけを検査するように指定する。 |
| <code>-c</code> | <code>pkgchk</code> のデフォルトであるファイル内容と属性ではなく、ファイル内容だけを検査するように指定する。 |
| <code>-v</code> | <code>pkgchk</code> による処理中、ファイル名を表示する詳細モードを指定する。 |
| <code>-d spooldir</code> | スプールディレクトリの絶対パスを指定する。 |
| <code>pkgid</code> | (省略可能) 1 つまたは複数のパッケージを空白で区切って指定する。 <code>pkgid</code> を指定しないと、 <code>pkgchk</code> は、システムにインストールされたすべてのソフトウェアパッケージをチェックする。これを省略すると、 <code>pkgchk</code> は使用可能なすべてのパッケージを表示する。 |

例 — インストールされたパッケージの内容を検査する

次の例は、パッケージの内容をチェックする方法を示しています。

```
# pkgchk -c SUNWadmfw
```

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

例 — インストールされたパッケージのファイル属性を検査する

次の例は、パッケージのファイル属性をチェックする方法を示しています。

```
# pkgchk -a SUNWadmfw
```

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

例 — スプールディレクトリにインストールされたパッケージを検査する

次の例は、スプールディレクトリ (/export/install/packages) にコピーされたソフトウェアパッケージをチェックする方法を示しています。

```
# pkgchk -d /export/install/packages
## checking spooled package <SUNWadmap>
## checking spooled package <SUNWadmfw>
## checking spooled package <SUNWadmc>
## checking spooled package <SUNWsdml>
```

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

▼ パッケージに関する詳細を表示する方法

pkginfo -l コマンドを使用して、インストールされたパッケージに関する情報を表示します。

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

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

pkgid (省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、*pkginfo* は、使用可能なすべてのパッケージに関する情報を表示する。

例 — パッケージに関する詳細を表示する

```
$ pkginfo -l SUNWcar
  PKGINST: SUNWcar
  NAME: Core Architecture, (Root)
  CATEGORY: system
  ARCH: sparc.sun4m
  VERSION: 11.6.0,REV=1998.05.06.20.36
  BASEDIR: /
  VENDOR: Sun Microsystems, Inc.
  DESC: core software for a specific hardware platform group
  PSTAMP: on99819980507193137
  INSTDATE: Jun 02 1998 11:43
  HOTLINE: Please contact your local service provider
  STATUS: completely installed
  FILES: 48 installed pathnames
         5 shared pathnames
         7 directories
         20 executables
         3299 blocks used (approx)
```

サーバーとスタンドアロンシステムからの パッケージの削除



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

▼ パッケージを削除する方法

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

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

pkgid

(省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定する。これを省略すると、使用可能なパッケージすべてが表示される。

▼ スプールされたパッケージを削除する方法

1. スーパーユーザーとしてログインします。
2. インストールされたパッケージを、`pkgrm -s` コマンドによってスプールディレクトリから削除します。

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

`-s spooldir`

パッケージがスプールされたスプールディレクトリの名前を指定する。

pkgid

(省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定する。*pkgid* を指定しないと、`pkgrm` は、ユーザーに対して、スプールディレクトリにある各パッケージを削除するように要求する。これを省略すると、使用可能なパッケージすべてが表示される。

▼ ディスクレスクライアントまたは AutoClient システムのパッケージを削除する方法

1. サーバーにログインして、スーパーユーザーになります。
2. `pkgrm -R` コマンドによって、データレスクライアントの **OS** サーバーからソフトウェアパッケージを削除します。

```
server# pkgrm -R rootpathpkgid...
```

`-R rootpath` クライアントのルートファイルシステムのマウントポイントを指定する。

`pkgid` (省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定する。この引数を省略すると、使用可能なパッケージすべてが表示される。

共有とマークされたクライアントのパッケージデータベース内のファイルは、サーバーから削除されませんが、クライアントのデータベースからは削除されます。すべてのクライアントがパッケージを削除した場合は、`pkgrm` をサーバーで個別に呼び出すことにより、サーバーから共有ファイルを削除することができます。

3. パッケージが正常に削除されたことを確認するには、`pkginfo` コマンドを使用してください。

```
server# pkginfo -R rootpath | egrep pkgid
```

`pkgid` がインストールされている場合、`pkginfo` コマンドは、それに関する情報を 1 行で返します。インストールされていない場合は、システムプロンプトに戻ります。

例 — ディスクレスクライアントのパッケージを削除する

次の例では、クライアントのルートファイルシステムが共有されるものと想定されています。また、これらのコマンドが、クライアントのサーバーで実行されることが前提となっています。

```
server# pkgrm -R /export/root/client-1 SUNWaudio
The following package is currently installed.
SUNWaudio
Do you want to remove this package? y/n/q?

Y
.
.
.
```

Admintool を使用してソフトウェアパッケージを追加または削除する

Solaris 7 には Admintool が組み込まれています。これは、ソフトウェアパッケージの追加と削除を含む、いくつかの管理作業を実行するためのグラフィカルユーザーインターフェースです。Admintool を使用すると、特に次のことを実行できます。

- ソフトウェアパッケージをローカルシステムに追加する
- ソフトウェアパッケージをローカルシステムから削除する
- ローカルシステムにすでにインストールされたソフトウェアを表示する
- インストールされるソフトウェアパッケージをカスタマイズする
- ソフトウェアパッケージの代替インストール用ディレクトリを指定する

▼ Admintool を起動する方法

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

```
$ su
```

UNIX の sysadmin グループ (グループ 14) のメンバーでなければ、システムのスーパーユーザーになって、Admintool によりソフトウェアパッケージの追加または削除を行う必要があります。

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

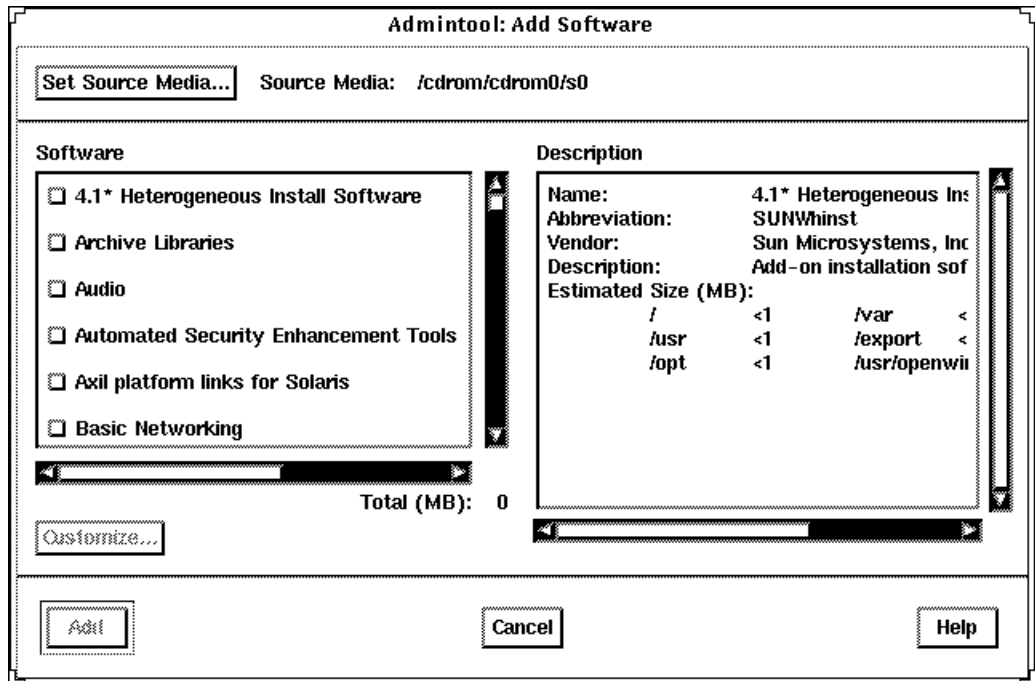
```
# admintool &
```

「ユーザー (Users)」ウィンドウが表示されます。

4. 「ブラウズ (**Browse**)」メニューから「ソフトウェア (**Software**)」を選択します。
「ソフトウェア (Software)」ウィンドウが表示されます。
5. 「編集 (**Edit**)」メニューから「追加 (**Add**)」を選択します。

「ソース媒体の設定 (Set Source Media)」ウィンドウが表示されます。必要であれば、インストール媒体へのパスを指定して、「OK」をクリックします。デフォルトのパスは、マウント済みの SPARC Solaris CD です。

「ソフトウェアを追加 (Add Software)」ウィンドウが表示されます。



- ローカルシステムにインストールしたいソフトウェアを選択します。
ウィンドウの「ソフトウェア (Software)」部分で、インストールしたいソフトウェアに対応するチェックボックスをクリックします。
- 「追加 (Add)」をクリックします。
各パッケージがインストールされるごとに、コマンドツールウィンドウが表示され、インストールの出力が表示されます。
「ソフトウェア (Software)」ウィンドウがリフレッシュされて、追加されたばかりのパッケージが表示されます。

▼ Admintool でパッケージを削除する方法

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

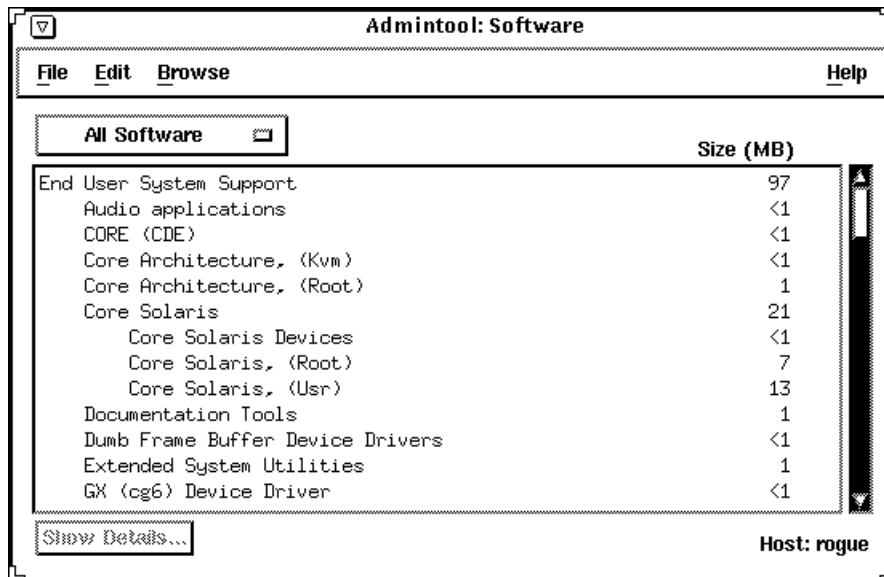
```
$ su
```

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

2. **Admintool** を起動します。

```
# admintool &
```

3. 「ブラウズ (**Browse**)」メニューから「ソフトウェア (**Software**)」を選択します。
「ソフトウェア (Software)」ウィンドウが表示されます。



4. ローカルシステムから削除したいソフトウェアを選択します。
5. 「編集 (**Edit**)」メニューから「削除 (**Delete**)」を選択します。
本当にソフトウェアを削除するのかどうかを確認するための警告ポップアップウィンドウが表示されます。
6. 「削除 (**Delete**)」をクリックして、ソフトウェアを削除したいことを確認します。
削除するパッケージごとにコマンドツールウィンドウが表示され、もう一度ソフトウェアを削除するのかどうか確認されます。y、n、または q を入力します。ソフトウェアを削除することを選択した場合、削除プロセスからの出力が表示されます。

パッチ管理の概要

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

この章の内容は次のとおりです。

- 303ページの「パッチについて」
- 304ページの「パッチ管理用ツール」
- 305ページの「パッチの配布」
- 307ページの「パッチの番号付け」
- 307ページの「パッチの追加に伴う動作」
- 308ページの「パッチの削除に伴う動作」

パッチについて

パッチの最も単純な形式としては、ソフトウェアを実行するのに障害となる既存のファイルとディレクトリを置換または更新する、ファイルとディレクトリの集合と考えることができます。既存のソフトウェアと同様に、パッチはアプリケーションバイナリインタフェースに準拠している指定のパッケージ形式を元に作成されます。(パッケージについての詳細は、第 16 章を参照してください。)

パッチ管理用ツール

パッチ管理用ツールには、次の 2 つがあります。

- `patchadd` – ディレクトリ形式のパッチを Solaris 2.7 システムにインストールするために使用する。
- `patchrm` – Solaris 7 システムにインストールされたパッチを削除するために使用する。このコマンドは、ファイルシステムを、パッチが適用される以前の状態に復元する。

パッチのインストールとバックアウトの詳細は、すべてのパッチに付いている `Install.info` ファイルに示されています。各パッチには、パッチについての特定の情報が入っている `README` ファイルも含まれています。

パッチをインストールするにあたっては、以前にインストールされたパッチの詳細を知る必要がある場合もあります。表 18-1 に、すでにシステムにインストールされているパッチについての情報を表示するためのコマンドを示します。

表 18-1 パッチ管理に役立つコマンド

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

パッチの配布

Sun のユーザーはすべて、セキュリティパッチなどの推奨されるパッチに、World Wide Web または anonymous ftp を介してアクセスできます。サービス契約を購入した Sun ユーザーは、パッチの拡張セットと、パッチ情報に関する完全なデータベースにアクセスできます。この情報は、World-Wide Web、anonymous ftp を介して入手可能であり、通常は CD によって定期的に配布されます (表 18-2 を参照)。

表 18-2 ユーザーパッチアクセス情報

| ユーザーの種類 | アクセスできる情報 |
|----------------------|--|
| Sun Service 契約ユーザー | パッチに関する SunSolve データベースとパッチ情報にアクセスすることができる。これらの情報は、306ページの「World-Wide Web を介したパッチアクセス」と 306ページの「ftp を介したパッチアクセス」に説明されているように、World-Wide Web または anonymous ftp を介して入手できる。 これらのパッチは、夜間に更新される。ユーザーは、パッチ CD を 6~8 週間ごとに受け取る。 |
| Sun Service 契約ユーザー以外 | セキュリティパッチの一般セットなどの推奨されるパッチにアクセスできる。これらの情報は、306ページの「World-Wide Web を介したパッチアクセス」と 306ページの「ftp を介したパッチアクセス」に説明されているように、World-Wide Web または anonymous ftp を介して入手できる。 |

Sun パッチにアクセスするために必要な条件

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

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

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

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

- インターネットに接続されている
- ftp プログラムを実行できる

World-Wide Web を介したパッチアクセス

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

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

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

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

URL を使用して、公開されているパッチにアクセスすることもできます。

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

ftp を介したパッチアクセス

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

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

パッチの番号付け

パッチは、固有の英数字文字列によって識別されます。これは、パッチのベース番号、ハイフン、パッチの改訂バージョン番号、の順で構成されています。たとえばパッチ 101977-02 は、lockd デーモンの問題を修正するための Solaris 2.4 パッチです。

パッチの追加に伴う動作

パッチを追加すると、patchadd コマンドは、パッチディレクトリからローカルシステムのディスクへファイルをコピーします。patchadd は、さらに次の処理も行います。

- 管理ホストとターゲットホストの Solaris バージョン番号を確認する。
- インストール中のパッチによって置換されるパッチ、このパッチに必要な他のパッチ、およびこのパッチと互換性を持たないパッチに関する情報を使用して、パッチの pkginfo ファイルを更新する。

Solaris 2.4 およびその互換バージョンでは、patchadd はパッチのインストール時に、パッチ情報の記録を `/var/sadm/patch/patch-number/log` に保存します。また、Solaris 2.5 およびその互換バージョンでは、インストールエラーが発生した場合にのみ、この場所にログファイルが格納されます。

この patchadd コマンドは、次の条件ではパッチをインストールしません。

- パッケージがホストに完全にインストールされていない。
- パッチのアーキテクチャが、システムのアーキテクチャと異なる。
- パッチのバージョンが、インストールされているパッケージのバージョンと一致しない。
- 同じベースコードと上位のバージョン番号を持つパッチがすでにインストールされている。
- パッチが、すでにインストールされている他のパッチと互換性を持たない。(インストール済みの各パッチは、この情報を pkginfo ファイルに格納している。)
- インストールしようとしているパッチには他のパッチが必要だが、そのパッチがインストールされていない。

パッチの削除に伴う動作

パッチをバックアウトすると、`patchrm` コマンドは、そのパッチによって修正されたすべてのファイルを復元します。ただし、次の場合は除きます。

- パッチが `patchadd -d` を使用してインストールされた (`-d` オプションは、更新中または置換中のファイルのコピーを保管しないように `patchadd` に指示する)。
- パッチが最新のパッチによって置換された。
- パッチが他のパッチによって要求された。

`patchrm` コマンドは、`pkgadd` を呼び出して、最初のパッチインストールで保管されたパッケージを復元します。

パッチのインストール時、`patchrm` は、`/tmp/backoutlog.process_id.` にパッチインストールログを格納します。パッチが正常にバックアウトされた場合、このログファイルは削除されます。

パート VI デバイスの管理

このパートでは、Solaris 7 環境でデバイスを管理する方法を説明します。次の章で構成されています。

第 19 章

デバイス設定の概要とデバイスを設定するための手順を説明します。

第 20 章

デバイス命名規則の概要とデバイスにアクセスするための手順を説明します。



デバイス管理の概要と手順

この章では、Solaris 環境の周辺デバイスを管理する方法の概要を示します。

この章の概要は次のとおりです。

- 313ページの「デバイス管理作業の参照先」
- 313ページの「デバイスドライバについて」
- 314ページの「デバイスの自動設定」
- 315ページの「システムへ周辺デバイスを追加する」
- 319ページの「デバイス設定情報の表示」

この章で説明する手順は次のとおりです。

- 316ページの「周辺デバイスを追加する方法」
- 317ページの「デバイスドライバを追加する方法」
- 320ページの「システム設定情報を表示する方法」
- 322ページの「デバイス情報を表示する方法」

デバイスへのアクセスについては、第 20 章を参照してください。

Solaris 環境のデバイス管理には、通常、システムでの周辺デバイスの追加と削除、デバイスをサポートするためのサードパーティデバイスドライバの追加、システム設定情報の表示が含まれます。

デバイス管理における新機能

この節では、Solaris 7 のデバイス管理における新機能について説明します。

動的再構成

動的再構成は、一部の SPARC サーバー上で使用できる機能です。この機能を使用することにより、サービスプロバイダは、システムを稼動したまま、ホットプラグ対応のシステム入出力ボードの取り外しや交換が行え、システムをリブートする手間と時間が省けます。また、交換すべきボードがすぐに入手できない場合でも、システム管理者は動的再構成を使用して、故障したボードだけをシャットダウンし、システムの稼動を継続できます。

使用しているサーバーが動的再構成をサポートしているかどうかを調べるには、ハードウェアのマニュアルを参照してください。

x86: SCSI ディスクドライバ (sd)

以前の Solaris リリースでは、x86 プラットフォーム上での SCSI ディスクのサポートは `cmdk` ドライバが処理していました。Solaris 7 では、`sd` ドライバが処理します。`sd` ドライバは、Solaris SPARC プラットフォームの SCSI ディスクドライバと似ています (SPARC プラットフォームのドライバも `sd` と呼ばれます)。

これらのデバイスの管理方法に変更はありません。システム管理者は、`format` ユーティリティと同様に、`prtconf`、`sysdef`、および `dmesg` コマンドの出力で、`cmdk` の代わりに `sd` を参照します。

`sd` の機能は `cmdk` が提供する機能のスーパーセットであるため、(`/dev/dsk` で論理ディスク名を使用する) アプリケーションは、ドライバの変更による影響を受けません。IDE デバイスを持つ x86 システムは、引き続き `cmdk` ドライバを使用します。

デバイス管理作業の参照先

表 19-1 に、プリンタやモデムなどのシリアルデバイスと、ディスク、CD-ROM、テープドライブなどの周辺デバイスをシステムに追加するための手順を説明している参照先を示します。

表 19-1 デバイスを追加する場合の参照先

| 作業内容 | 参照先 |
|----------------------|---|
| ディスクの追加 | 第 23 章または第 24 章 |
| CD-ROM またはテープデバイスの追加 | 316ページの「周辺デバイスを追加する方法」 |
| モデムの追加 | 『Solaris のシステム管理 (第 2 巻)』の「端末とモデム管理の概要」 |
| プリンタの追加 | 『Solaris のシステム管理 (第 2 巻)』の「印刷管理の概要」 |

デバイスドライバについて

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

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

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

デバイスの自動設定

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 環境で自動的に設定されます。

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 19-2 で説明します。

表 19-2 カーネルモジュール

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

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

`/etc/system` ファイルを修正することにより、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、

『Solaris のシステム管理 (第 2 巻)』の「カーネルパラメタの調整手順」を参照してください。

機能と利点

自動構成の利点は次のとおりです。

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

自動構成プロセスは、システム管理者が新しいデバイス (およびドライバ) をシステムに追加するときに使用されます。これは、再構成ブートを実行することにより行われるため、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。

詳細については、デバイスのご購入先にお問い合わせください。

システムへ周辺デバイスを追加する

新しい周辺デバイスを追加する場合、通常、次の作業が必要になります。

- システムのシャットダウン
- システムへのデバイスの接続

- システムのリブート

次のデバイスをシステムに追加するには、以下で説明する手順に従ってください。

- CD-ROM

- ディスクドライブ

- テープドライブ

- SBUS カード

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

▼ 周辺デバイスを追加する方法

1. スーパーユーザーになります。
2. デバイスをサポートするためにデバイスドライバを追加する必要がある場合は、317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** に従います。
3. `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

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この `/reconfigure` ファイルによってチェックします。

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

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

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

5. システムがシャットダウンされたら、電源を落とします。

| SPARC システムの場合 | x86 システムの場合 |
|-----------------------------------|---|
| ok または > プロンプトが表示されたら、電源を落としても安全。 | Type any key to continue プロンプトが表示されたら、電源を落としても安全。 |

電源スイッチの位置については、各システムに添付のハードウェアマニュアルを参照してください。

6. すべての外部デバイスの電源を落とします。
周辺デバイスの電源スイッチの位置については、各自の周辺デバイスに添付のハードウェアマニュアルを参照してください。
7. 周辺デバイスをインストールして、追加するデバイスのターゲット番号がシステム上の他のデバイスとは異なることを確認します。
ディスクの裏側に小さいスイッチがあるはずですが、
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 20 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、316ページの「周辺デバイスを追加する方法」を参照してください。

1. スーパーユーザーになります。
2. 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. pkgadd コマンドを使用して、ドライバをインストールします。

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

-d device

デバイスのバス名を指定する。

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. pkgchk コマンドを使用して、パッケージが正しく追加されていることを確認します。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。

```
# pkgchk packagename  
#
```

例 — デバイスドライバを追加する

次の例では、XYZdrv というパッケージをインストールして確認しています。

```
# pkgadd XYZdrv  
(ライセンス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
```


デバイス設定情報の表示

システムとデバイスの設定情報を表示するには、次の 3 つのコマンドを使用します。

| | |
|---------------------------|--|
| <code>prtconf (1M)</code> | メモリーの総量、システムのデバイス階層によって記述されたデバイス設定を含む、システム設定情報を表示します。このコマンドによる出力は、システムのタイプによって異なります。 |
| <code>sysdef (1M)</code> | システムハードウェア、疑似デバイス、ロード可能なモジュール、および指定のカーネルパラメータを含む、デバイス設定情報を表示します。 |
| <code>dmesg (1M)</code> | 最後のリポート以降にシステムに接続されたデバイスのリストと、システム診断情報を表示します。 |

システム上のデバイスを識別するために使用されるデバイス名については、326ページの「デバイス名の命名規則」を参照してください。

driver not attached メッセージ

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

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

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

システムデバイスの識別

`prtconf` コマンドと `sysdef` コマンドを使用すると、デバイスインスタンスの次に示される「driver not attached」メッセージを確認することによって、システムに接続されたディスク、テープ、CD-ROM デバイスを識別できます。これらのデバイスは、何らかのシステムプロセスによって常に監視されているため、

「driver not attached」メッセージは通常、そのデバイスインスタンスにデバイスがないことを示す良い標識になります。

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

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

同じデバイス情報は、sysdef 出力からも得られます。

▼ システム設定情報を表示する方法

システム設定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設定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 — システム設定情報を表示する

次の prtconf 出力は、SPARC システム上で表示されます。

```

# prtconf
System Configuration: Sun Microsystems sun4c
Memory size: 32 Megabytes
System Peripherals (Software Nodes):
SUNW,Sun 4_50
  packages (driver not attached)
    disk-label (driver not attached)
    deblocker (driver not attached)
    obp-tftp (driver not attached)
  options, instance #0
  aliases (driver not attached)
  openprom (driver not attached)
  zs, instance #0
  zs, instance #1
  audio (driver not attached)
  eeprom (driver not attached)
  counter-timer (driver not attached)
  memory-error (driver not attached)
  interrupt-enable (driver not attached)
  auxiliary-io (driver not attached)
  sbus, instance #0
    dma, instance #0
    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
  .
  .
  .

```

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)

```

(続く)

続き

```
i86pc-mmu (driver not attached)
openprom (driver not attached)
options, instance #0
packages (driver not attached)
delayed-writes (driver not attached)
itu-props (driver not attached)
isa, instance #0
motherboard (driver not attached)
pnpADP,1542, instance #0
asy, instance #0
asy, instance #1
lp, instance #0 (driver not attached)
fdc, instance #0
fd, instance #0
fd, instance #1 (driver not attached)
kd (driver not attached)
kdmouse (driver not attached)
.
.
.
```

▼ デバイス情報を表示する方法

デバイス情報は、`dmesg` コマンドを使用して表示してください。

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

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

例 — デバイス情報を表示する

SPARC システムからは、次の `dmesg` 出力が表示されます。

```
# dmesg
SunOS Release 5.7 Version generic [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1998, Sun Microsystems, Inc.
pac: enabled - SuperSPARC
cpu0: TI,TMS390Z50 (mid 8 impl 0x0 ver 0x0 clock 40 MHz)
mem = 32768K (0x2000000)
avail mem = 27803648
Ethernet address = 8:0:20:1f:33:9g
root nexus = SUNW,SPARCstation-10
iommu0 at root: obio 0xe0000000
```

(続く)

続き

```
sbus0 at iommu0: obio 0xe0001000
dma0 at sbus0: SBus slot f 0x400000
dma0 is /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000 (esp0):
  esp-options=0x46
esp0 at dma0: SBus slot f 0x800000 sparc ipl 4
esp0 is /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000
sd1 at esp0: target 1 lun 0
sd1 is /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  sd3 at esp0: target 3 lun 0
sd3 is /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
  root on /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0:a ...
obio0 at root
obio0 at obio0: obio 0x100000 sparc ipl 12
zs0 is /obio/zs@0,100000
obio1 at obio0: obio 0x0 sparc ipl 12
zs1 is /obio/zs@0,0
cpu 0 initialization complete - online
ledma0 at sbus0: SBus slot f 0x400010
le0 at ledma0: SBus slot f 0xc00000 sparc ipl 6
le0 is /iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010/le@f,c00000
dump on /dev/dsk/c0t3d0s1 size 200 MB
#
```

x86 システムからは、次の dmesg 出力が表示されます。

```
# dmesg
SunOS Release 5.7 Version generic [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1998, Sun Microsystems, Inc.
mem = 130684K (0x7f9f000)
avail mem = 114970624
root nexus = i86pc
isa0 at root
ISA-device: asy0
asy0 is /isa/asy@1,3f8
ISA-device: asy1
asy1 is /isa/asy@1,2f8
pci0 at root: space 0 offset 0
  IDE device at targ 0, lun 0 lastlun 0x0
  model WDC AC31000H, stat 50, err 0
  cfg 0x427a, cyl 2100, hd 16, sec/trk 63
  mult1 0x8010, mult2 0x108, dwcap 0x0, cap 0xf00
  piomode 0x180, dmamode 0x200, advpiomode 0x1
  minpio 380, minpioflow 180
  valid 0x3, dwdma 0x3, majver 0x0
ata_set_feature: (0x66,0x0) failed
ISA-device: ata0
Disk0: cmdk0 at ata0 target 0 lun 0
cmdk0 is /isa/ata@1,1f0/cmdk@0,0
root on /isa/ata@1,1f0/cmdk@0,0:a fstype ufs
Number of console virtual screens = 13
```

(続く)

続き

```
cpu 0 initialization complete - online
Ethernet address = 2:7:1:1c:27:e5
.
.
.
```

デバイスへのアクセス

この章では、システム管理者がシステム上のデバイスにアクセスする方法について説明します。

この章の内容は次のとおりです。

- 325ページの「デバイスへのアクセス」
- 327ページの「論理ディスクデバイス名」
- 331ページの「論理テープデバイス名」
- 332ページの「論理 CD-ROM デバイス名」

デバイスの構成についての概要は、第 19 章を参照してください。

デバイスへのアクセス

システム管理者は、コマンドを使用してディスク、ファイルシステムなどのデバイスを管理する場合、デバイス名を指定する方法を知っている必要があります。通常システム管理者は、論理デバイス名を使用してシステムに接続されたデバイスを表現します。論理デバイス名と物理デバイス名は、システム上でそれぞれ論理デバイスファイルと物理デバイスファイルによって表現されます。

デバイス情報が作成される方法

システムがブートされると、デバイス階層が作成されて、システムに接続されたすべてのデバイスが表示されます。カーネルは、このデバイス階層情報を使用して、ドライバを該当するデバイスに対応づけて、特定の操作を実行するドライバへの一連のポインタを与えます。デバイス階層情報の詳細については、『*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。

デバイス名の命名規則

Solaris 環境では、デバイスは次の 3 つの方法で参照されます。

- 物理デバイス名 – デバイス情報階層の完全デバイスパス名を表します。物理デバイス名は、次のコマンドを使用して表示できます。

- `dmesg`
- `format`
- `sysdef`
- `prtconf`

物理デバイスファイルは、`/devices` ディレクトリにあります。

- インスタンス名 – システム上のデバイスすべてのカーネル短縮名を表します。たとえば、`sd0` と `sd1` は、2 つのディスクデバイスのインスタンス名を表します。インスタンス名は、`/etc/path_to_inst` ファイルにマップされて、次のコマンドにより表示できます。

- `dmesg`
- `sysdef`
- `prtconf`

- 論理デバイス名 – システム管理者が、デバイスを参照するために使用します。論理デバイス名を使用するファイルコマンドは、表 20-1 に記載されています。`/dev` ディレクトリの論理デバイスファイルは、`/devices` ディレクトリの物理デバイスファイルにシンボリックリンクされています。

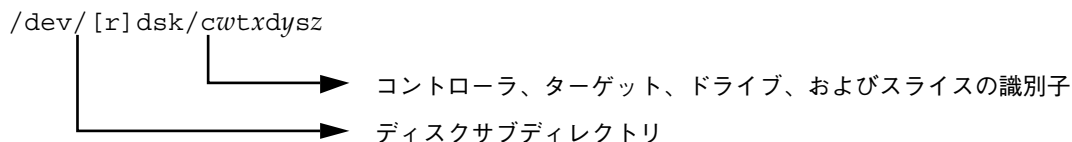
論理ディスクデバイス名

論理デバイス名は、次の場合に、ディスクデバイスにアクセスするために使用されます。

- システムに新しいディスクを追加する。
- あるシステムから別のシステムへディスクを移動する。
- ローカルディスク上にあるファイルシステムにアクセス (またはマウント) する。
- ローカルファイルシステムのバックアップをとる。

管理コマンドの多くは、ディスクスライスまたはファイルシステムを参照する引数を使用します。

シンボリックリンクされるサブディレクトリ (`/dev/dsk` または `/dev/rdisk` のいずれか) に続けて、特定のコントローラ、ディスク、およびスライスを識別する文字列を指定することによって、ディスクデバイスを参照してください。



ディスクサブディレクトリの指定

ディスクとファイルの管理コマンドには、`raw` (または「キャラクタ型」) デバイスインタフェースか、「ブロック」デバイスインタフェースを使用する必要があります。この区別は、データがデバイスから読み取られる方法によって決まります。

`raw` デバイスインタフェースは、一度に少量のデータだけを転送します。ブロックデバイスインタフェースには、大量のデータブロックが一度に読み取られるバッファが含まれます。

コマンドによって、必要なインタフェースは異なります。

- コマンドが `raw` デバイスインタフェースを必要とする場合は、`/dev/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。

表 20-1 は、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ています。

表 20-1 使用頻度の高いコマンドに必要なデバイスインタフェースのタイプ

| コマンド | インタフェースのタイプ | 使用例 |
|---------------------------|-------------|---|
| <code>df (1M)</code> | ブロック | <code>df /dev/dsk/c0t3d0s6</code> |
| <code>fsck (1M)</code> | raw | <code>fsck -p /dev/rdisk/c0t0d0s0</code> |
| <code>mount (1M)</code> | ブロック | <code>mount /dev/dsk/c1t0d0s7 /export/home/ziggy</code> |
| <code>newfs (1M)</code> | raw | <code>newfs /dev/rdisk/c0t0d1s1</code> |
| <code>prtvtoc (1M)</code> | raw | <code>prtvtoc /dev/rdisk/c0t0d0s2</code> |

スライスの指定

特定のスライスを特定のディスクで識別するために使用する文字列は、コントローラのタイプが、直接かバス指向のどちらであるかによって決まります。表 20-2 は、コントローラが直接かバス指向かによって、プラットフォームが何になるかを示しています。

表 20-2 コントローラのタイプ

| 直接コントローラ | バス指向コントローラ |
|------------------|------------------|
| Xylogics (SPARC) | SCSI (SPARC/x86) |
| IDE (x86) | IPI (SPARC) |

両方のタイプのコントローラについて、次の項で説明します。

注・コントローラ番号は、システム初期設定時に自動的に割り当てられます。この番号は、厳密に論理的なものであり、物理コントローラに直接対応するものではありません。

SPARC: 直接コントローラでアクセスされるディスク

SPARC システムにおいて直接コントローラでアクセスされるディスクにスライスを指定するには、図 20-1 に示す命名規則に従ってください。

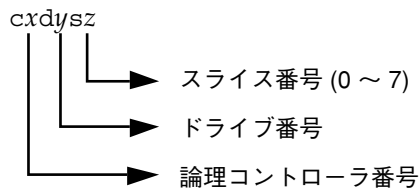


図 20-1 SPARC システムにおいて直接コントローラでアクセスされるディスクの命名規則

ディスク全体を示すには、スライス 2 (`s2`) を指定してください。

システムにコントローラが 1 つしかない場合、`x` は常に 0 になります。

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、図 20-2 に示す命名規則に従ってください。

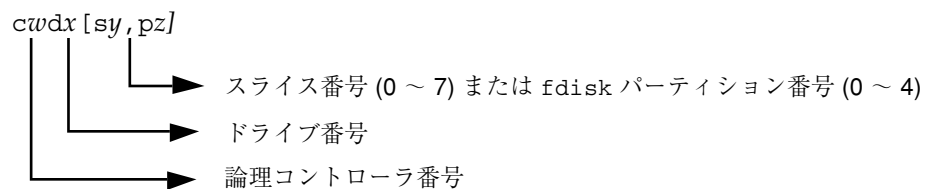


図 20-2 x86 システムにおいて IDE コントローラでアクセスされるディスクの命名規則

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、*w* は常に 0 になります。

SPARC: バス指向コントローラでアクセスされるディスク

SPARC システムにおいてバス指向コントローラ (SCSI など) でアクセスされるディスクにスライスを指定するには、図 20-3 に示す命名規則に従ってください。

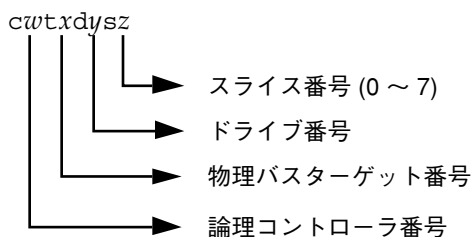


図 20-3 SPARC システムにおいてバス指向コントローラでアクセスされるディスクの命名規則

システムにコントローラが 1 つしかない場合、*w* は常に 0 になります。

SCSI コントローラの場合、*x* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*y* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*y* は通常 0 になります。

ディスク全体を示すには、スライス 2 (s2) を指定してください。

x86: SCSI コントローラでアクセスされるディスク

x86 システムにおいて SCSI コントローラでアクセスされるディスクにスライスを指定するには、図 20-4 に示す命名規則に従ってください。

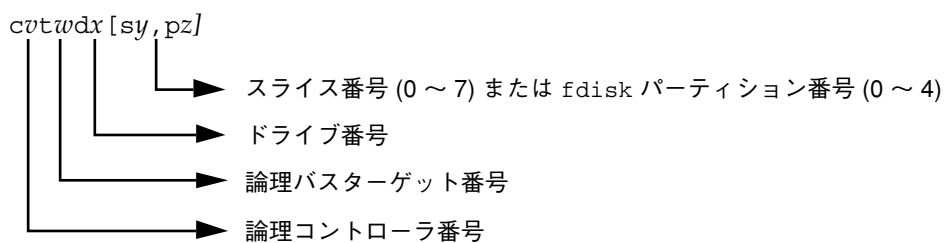


図 20-4 x86 システムにおいて SCSI コントローラでアクセスされるディスクの命名規則

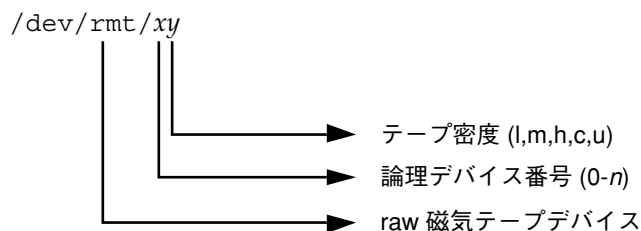
システムにコントローラが 1 つしかない場合、*v* は常に 0 になります。

SCSI コントローラの場合、*w* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*x* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクに組み込みコントローラがある場合、*x* は通常 0 になります。

Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

論理テープデバイス名

論理テープデバイスファイルは、`/devices` ディレクトリからのシンボリックリンクとして、`/dev/rmt/*` ディレクトリにあります。



システムに接続された最初のテープデバイスは 0 (`/dev/rmt/0`) であり、これは、QIC-11、QIC-24、QIC-150、または Exabyte のいずれかのタイプになります。テープ密度の値 (l、m、h、c、u) については、第 38 章を参照してください。

論理 CD-ROM デバイス名

システム上の最初の CD-ROM デバイスを表す論理デバイス名は、`/dev/dsk/c0t6d0s0` です。

CD-ROM はボリューム管理によって管理されるため、論理 CD-ROM デバイス名は、CD を手作業でマウントする必要がある場合以外は使用しません。CD-ROM デバイスへのアクセスについては、第 11 章を参照してください。

パート VII ディスクの管理

このパートでは、Solaris 環境でディスクを管理する手順について説明します。次の章で構成されています。

| | |
|--------|--|
| 第 21 章 | Solaris のディスクスライスの概要と、format ユーティリティを紹介します。 |
| 第 22 章 | ディスクのフォーマット、ディスクラベルの検査、欠陥ディスクセクターの修復を行う手順について説明します。 |
| 第 23 章 | SPARC システムにディスクを追加する手順について説明します。 |
| 第 24 章 | x86 システムにディスクを追加する手順について説明します。 |
| 第 25 章 | format ユーティリティのメニューとコマンドについて説明します。また、format.dat ファイル、format コマンドの入力規則、ヘルプ機能の使い方についても説明します。 |



ディスク管理の概要

この章では、Solaris のディスクスライスの概念と `format` ユーティリティについて説明します。

この章の内容は次のとおりです。

- 336ページの「ディスク関連の用語」
- 336ページの「ディスクスライスについて」
- 337ページの「SPARC: ディスクスライス」
- 339ページの「x86: ディスクスライス」
- 341ページの「使用するスライスの決定」
- 342ページの「`format` ユーティリティ」
- 344ページの「`format` ユーティリティ使用上のガイドライン」
- 346ページの「ディスクのフォーマット」
- 347ページの「ディスクラベルについて」
- 347ページの「パーティションテーブル」

システムにディスクドライブを追加する方法については、第 23 章または第 24 章を参照してください。

ディスク管理作業の参照先

ディスク管理の手順については、次を参照してください。

- 第 23 章
- 第 24 章

概要

一般に、Solaris 環境におけるディスクの管理とは、システムを設定し、Solaris インストールプログラムを実行し、適切なディスクスライスを作成してオペレーティングシステムをインストールすることを意味します。また、format ユーティリティを使用して、新しいディスクドライブを追加したり、欠陥ディスクドライブを交換したりしなければならない場合もあります。

ディスク関連の用語

この節の説明を有効に利用するには、基本的なディスクアーキテクチャを理解しておく必要があります。特に、次の用語を理解しておいてください。

- トラック
- シリンダ
- セクター
- ディスクコントローラ
- ディスクラベル
- デバイスドライバ

上記の用語をよく理解していない場合は、用語集やディスクメーカーの製品情報を参照してください。

ディスクスライスについて

ディスク上に格納されたファイルは、ファイルシステム中で管理されます。ディスク上の各ファイルシステムは「スライス」、つまり、そのファイルシステム用に確保されたシリンダのグループに割り当てられます。オペレーティングシステム (およ

び、システム管理者) からは、各ディスクスライスは別個のディスクドライブであるかのように見えます。

ファイルシステムについての詳細は、パート VIII 「ファイルシステムの管理」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。表 21-1 に、両者の違いを示します。

表 21-1 プラットフォームによるスライスの違い

| SPARC | x86 |
|----------------------------------|--|
| ディスク全体が Solaris 専用になる。 | 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。 |
| ディスクは0から7までの番号が付いた8つのスライスに分割される。 | Solaris の <code>fdisk</code> パーティションは0から9までの番号が付いた10のスライスに分割される。 |

SPARC: ディスクスライス

SPARC システム上では、Solaris は8つのディスクスライスを定義して、それぞれにある程度決まった役割を割り当てます。これらのスライスには、0から7までの番号が付いています。表 21-2 に、SPARC システム上の8つの Solaris スライスの内容を示します。

表 21-2 SPARC : ディスクスライス

| スライス | ファイルシステム | 通常クライアントまたはサーバーのどちらにあるか | 用途 |
|------|---------------------------|-------------------------|--|
| 0 | ルート | 両方 | オペレーティングシステムを構成するファイルとディレクトリを含む。 |
| 1 | スワップ | 両方 | 仮想メモリー、つまり「スワップ空間」を提供する。スワップ空間は、実行中のプログラムが大きすぎてコンピュータのメモリーに入りきらないときに使用される。その場合、Solaris 環境では、プログラムがメモリーからディスクに「スワップ」され、必要に応じて戻される。 |
| 2 | — | 両方 | 慣習により、このスライスはディスク全体を表す。このスライスは、Sun の format と Solaris インストールプログラムによって自動的に定義される。このスライスのサイズは変更しないこと。 |
| 3 | /export | サーバーのみ | オペレーティングシステムの代替バージョンを含む。これらの代替バージョンは、サーバーとはアーキテクチャが異なるクライアントシステムに必要である。アーキテクチャのタイプがサーバーと同じクライアントは、/usr ファイルシステム (通常はスライス 6) にある実行可能プログラムを利用する。 |
| 4 | /export/swap | サーバーのみ | クライアントシステムに仮想メモリー領域を提供する。 |
| 5 | /opt | 両方 | システムに追加されるアプリケーションソフトウェアを含む。インストール時に、このファイルシステムにスライスが割り当てられていなければ、/opt ディレクトリがスライス 0 に入る。 |
| 6 | /usr | 両方 | ユーザーが実行するオペレーティングシステムコマンド (「実行可能」コマンドとも呼ぶ) を含む。また、このスライスには、オンラインマニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。 |
| 7 | /home または /export/home | 両方 | ユーザーによって作成されるファイルを含む。 |

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。

表 21-3 のように、Solaris は x86 システムのディスク上の Solaris fdisk パーティション上に、0 から 9 までの番号が付いた 10 のスライスを配置します。

表 21-3 x86 : ディスクスライス

| スライス | ファイルシステム | 通常クライアントまたはサーバーのどちらにあるか | 用途 |
|------|--------------|-------------------------|---|
| 0 | ルート | 両方 | オペレーティングシステムを構成するファイルとディレクトリを含む。 |
| 1 | スワップ | 両方 | 仮想メモリー、つまり「スワップ空間」を提供する。スワップ空間は、実行中のプログラムが大きすぎてコンピュータのメモリーに入りきらないときに使用される。その場合、Solaris 環境では、プログラムがメモリーからディスクに「スワップ」され、必要に応じて戻される。 |
| 2 | — | 両方 | 慣習により、このスライスは Solaris fdisk パーティション全体を表す。このスライスは、Sun の format ユーティリティと Solaris インストールプログラムによって自動的に定義される。このスライスのサイズは変更しないこと。 |
| 3 | /export | サーバーのみ | オペレーティングシステムの代替バージョンを含む。これらの代替バージョンは、サーバーとはアーキテクチャが異なるクライアントシステムに必要である。 |
| 4 | /export/swap | サーバーのみ | クライアントシステムに仮想メモリーを提供する。 |
| 5 | /opt | 両方 | システムに追加されるアプリケーションソフトウェアを含む。インストール時に、このファイルシステムにスライスが割り当てられていなければ、/opt ディレクトリがスライス 0 に入る。 |

表 21-3 x86 : ディスクスライス 続く

| スライス | ファイルシステム | 通常クライアントまたはサーバーのどちらにあるか | 用途 |
|------|------------------------|-------------------------|--|
| 6 | /usr | 両方 | ユーザーが実行するオペレーティングシステムコマンド (「実行可能」コマンドとも呼ぶ) を含む。また、このスライスには、マニュアル、システムプログラム (init や syslogd など)、ライブラリルーチンも含まれる。 |
| 7 | /home または /export/home | 両方 | ユーザーによって作成されるファイルを含む。 |
| 8 | — | 両方 | Solaris がハードディスクからブートするために必要な情報を含む。スライス番号は 8 であるが、この情報は、Solaris パーティションの先頭にあり、ブートスライスと呼ばれる。 |
| 9 | — | 両方 | 代替ディスクブロック用に予約された領域であり、代替セクタスライスと呼ばれる。 |

raw データスライスの使用

SunOS オペレーティングシステムは、各ディスクのブロック 0、シリンダ 0 に、ディスクラベルを格納します。これは、raw データスライスを作成する、Sun 以外のデータベースアプリケーションを使用するときは、ブロック 0、シリンダ 0 から開始してはならないことを意味します。この領域に raw データスライスを作成すると、ディスクラベルが上書きされて、ディスク上のデータにアクセスできなくなります。

ディスク上の次の領域は、raw データスライス用に使用しないでください。raw データスライスは Sun 以外のデータベースアプリケーションによって作成されることがあります。

1. ブロック 0、シリンダ 0 (ディスクラベルが格納される領域)
2. シリンダ 0 全体 (性能の向上のため)
3. スライス 2 (ディスク全体を表す)

複数のディスク上のスライス配置

十分な大きさのディスクであれば、1台ですべてのスライスとそれに対応するファイルシステムを確保できますが、通常はシステムのスライスとファイルシステムを確保するために複数のディスクが使用されます。

注・1つのスライスを複数のディスクに分割することはできません。ただし、複数のスワップスライスを別々のディスクに配置することはできます。

たとえば、1台のディスクにルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムを入れ、別のディスクは /export/home ファイルシステムやユーザーデータが入っている他のファイルシステムに使用できます。

複数のディスクを使用する場合、オペレーティングシステムソフトウェアとスワップ領域が入っているディスク (つまり、ルート (/)、/usr ファイルシステム、またはスワップ領域用のスライスが入っているディスク) を、「システムディスク」と呼びます。システムディスク以外のディスクを、「二次ディスク」または「非システムディスク」と呼びます。

システムのファイルシステムを複数のディスクに入れると、システムをシャットダウンしたりオペレーティングシステムソフトウェアをロードし直したりしなくても、二次ディスクのファイルシステムとスライスを変更できます。

また、複数のディスクを使用すると、入出力 (I/O) の性能が改善されます。ディスク負荷を複数のディスクに分散すると、I/O のボトルネックを回避できます。

使用するスライスの決定

ディスクのファイルシステムを設定するときには、各スライスのサイズだけでなく、どのスライスを使用するかも決定します。どのように決定するかは、ディスクを接続するシステムの構成と、ディスクにインストールしたいソフトウェアによって異なります。

次の4つのシステム構成があります。

- サーバー
- ディスクレスクライアント
- スタンドアロンシステム
- AutoClient システム

システム構成ごとに、使用すべきスライスが異なります。表 21-4は、これらの要件を示しています。

表 21-4 システム構成とスライスの要件

| スライス | サーバー | ディスクレス クライアント | スタンドアロン システム | AutoClient システム |
|------|--------------|------------------|-----------------|--------------------|
| 0 | ルート | (サーバー上) | ルート | ルート |
| 1 | スワップ | (サーバー上) | スワップ | スワップ |
| 2 | — | — | — | — |
| 3 | /export | — | — | — |
| 4 | /export/swap | — | — | — |
| 5 | /opt | (サーバー上) | /opt | (サーバー上) |
| 6 | /usr | (サーバー上) | /usr | (サーバー上) |
| 7 | /export/home | (サーバー上) | /home | (サーバー上) |

システム構成についての詳細は、第 4 章を参照してください。

注 - Solaris インストールプログラムは、インストール用に選択したソフトウェアに基づいて推奨スライスサイズを表示します。

format ユーティリティ

format ユーティリティの使用法や参照情報を読む前に概要を知りたい場合は、以下を読んでください。

定義

format ユーティリティは、Solaris システム用にハードディスクドライブを準備するためのシステム管理ツールです。format ユーティリティは、フロッピーディスクドライブ、CD-ROM ドライブ、またはテープドライブには使用できません。

機能と利点

表 21-5 に、format ユーティリティの機能とその利点を示します。

表 21-5 format ユーティリティの機能と利点

| 機能 | 利点 |
|-----------------------------|--|
| システム内で接続されている全ディスクドライブを検索する | 次の情報を表示する。 <ul style="list-style-type: none">■ ターゲットの位置■ ディスクのジオメトリ■ ディスクがフォーマット済みかどうか■ ディスク上にマウントされているパーティションが存在するかどうか |
| ディスクラベルを検索する | 修復処理に使用する。 |
| 欠陥セクターを修復する | 回復可能なエラーが発生したディスクドライブをメーカーに返送しなくても、熟練した管理者なら修復できる。 |
| ディスクをフォーマットして、分析する | ディスク上でセクターを作成し、検査する。 |
| ディスクをパーティションに分割する | 個々のファイルシステムを別々のスライス上で作成できるようにディスクを分割する。 |
| ディスクにラベルを付ける | 後から検索できるように (通常は修復用)、ディスクにディスク名と構成情報を書き込む。 |

format ユーティリティの全オプションについての詳細は、第 25 章を参照してください。

format ユーティリティを使用する場合

Solaris のインストール処理の一部として、Solaris インストールプログラムによってディスクがパーティションに分割され、ラベルが付けられます。次のような場合には、format ユーティリティを使用する必要があります。

- スライス情報を表示する。
- ディスクをスライスに分割する。
- 既存のシステムにディスクを追加する。
- ディスクをフォーマットする。
- ディスクを修復する。

システム管理者が format ユーティリティを使用するのは、主にディスクをディスクスライスに分割するためです。これらの手順については、第 23 章と第 24 章を参照してください。

format ユーティリティの使用上のガイドラインについては、表 21-6 を参照してください。

format ユーティリティ使用上のガイドライン

表 21-6 format ユーティリティのガイドライン

| 用途 | 注意事項 | 参照先 |
|----------------|--|---|
| ディスクをフォーマットする | <ul style="list-style-type: none"> ■ ディスクをフォーマットし直すと、既存のデータが失われる。 ■ ディスクドライブをフォーマットしてパーティションに分割した状態で出荷するメーカーが増えているので、ディスクドライブをフォーマットする必要性は減少している。既存のシステムにディスクドライブを追加する場合は、format ユーティリティを使用しなくてもすむことがある。 ■ ディスクを配置し直したら多数のディスクエラーが表示される場合は、フォーマットし直してみるとよい。不良セクターが自動的にマッピングし直される。 | 358ページの「ディスクをフォーマットする方法」 |
| システムディスクを交換する | <ul style="list-style-type: none"> ■ 損傷したシステムディスクのデータは、バックアップ媒体から復元しなければならない。復元しなければ、インストールプログラムを使用してシステムをもう一度インストールしなければならない。 | 第 23 章または第 24 章、システムをインストールし直さなければならない場合は、『Solaris のインストール (上級編)』 |
| ディスクをスライスに分割する | <ul style="list-style-type: none"> ■ すでにスライスに分割されているディスクをパーティションに分割し直してラベルを付け直すと、既存のデータが失われる。 ■ ディスクにラベルを付け直し、ディスクをパーティションに分割し直す前に、復元するために既存のデータをバックアップ媒体にコピーしなければならない。 | 第 23 章または第 24 章 |

表 21-6 format ユーティリティのガイドライン 続く

| 用途 | 注意事項 | 参照先 |
|-------------------|--|-----------------|
| 既存のシステムにディスクを追加する | <ul style="list-style-type: none"> ■ 二次ディスクをフォーマットし直すか、パーティションに分割し直す場合は、既存のデータをバックアップ媒体から復元しなければならない。 | 第 23 章または第 24 章 |
| ディスクドライブを修復する | <ul style="list-style-type: none"> ■ 顧客のサイトによっては、欠陥ドライブの修復ではなくドライブ自体の交換を希望する場合がある。サイトがディスクドライブのメーカーと保守契約を結んでいる場合は、format ユーティリティを使用してディスクドライブを修復する必要はない。 ■ 通常、ディスクドライブの修復とは、不良セクターを欠陥リストに追加することを意味する。新しいコントローラは不良セクターを自動的にマップし直すので、システムを中断する必要はない。 ■ システムに旧型のコントローラがある場合や、失われたデータを復元する場合は、不良セクターをマップし直す必要がある。 | 第 25 章 |

ディスクのフォーマット

ほとんどの場合、ディスクはメーカーまたは再販業者によってフォーマットされているので、ドライブをインストールするときにフォーマットし直す必要はありません。ディスクがフォーマットされているかどうかを判別するには、format ユーティリティを使用します。詳細は、358ページの「ディスクがフォーマット済みかどうかを調べる方法」を参照してください。

ディスクがフォーマットされていない場合、format ユーティリティを使用してフォーマットします。

ディスクをフォーマットする作業は、次の2つの手順に分けられています。

- 使用するディスク媒体を用意する。
- 表面解析に基づいてディスクの欠陥リストを作成する。



注意 - フォーマットは、ディスク上のデータを上書きします。このため、通常は、メーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、format ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドのみを使用するように注意してください。詳細は、358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。

フォーマットにはディスクの種類とサイズに応じて数分から数時間かかります。

ディスクラベルについて

どのディスクにも、そのディスクのコントローラ、ジオメトリ、スライスに関する情報を格納する特殊な領域が確保されています。そのような情報をディスクの「ラベル」と呼びます。また、ディスクラベルを表すのに VTOC (Volume Table of Contents) という用語を使用することもあります。「ディスクにラベルを付ける」とは、ディスクにスライス情報を書き込むことを意味します。通常は、ディスクのスライスを変更した後にラベルを付けます。

スライスを作成した後でディスクにラベルを付けないと、オペレーティングシステムはスライスを「認識」する方法がないので、そのスライスを利用できなくなります。

パーティションテーブル

ディスクラベルのうち重要な部分は「パーティションテーブル」です。この部分は、ディスクのスライス、スライスの境界(シリンダ単位)、スライスの合計サイズを表します。ディスクのパーティションテーブルは、format ユーティリティを使用して表示できます。表 21-7 にパーティションテーブル関連の用語を示します。

表 21-7 パーティションテーブル関連の用語

| 用語 | 値 | 説明 |
|-----|--|---|
| 番号 | 0-7 | パーティション (またはスライス番号)。有効な番号は 0 から 7 まで。 |
| タグ | 0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME | 一般にこのパーティションにマウントされたファイルシステムを記述する数値。 |
| フラグ | | |
| | wm | パーティションは書き込み可能でマウント可能である。 |
| | wu rm | パーティションは書き込み可能でマウント不可である。これは、スワップ領域専用のパーティションのデフォルト状態である。ただし、mount コマンドでは「マウント不可」のフラグはチェックされない。 |
| | rm | パーティションは読み取り専用でマウント可能である。 |

パーティションのフラグとタグは慣習により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、360ページの「ディスクスライス情報を表示する方法」または 365ページの「ディスクラベルを検査する方法」を参照してください。

例 — パーティションテーブル

次のパーティションテーブルの例は、535M バイトのディスクについて format ユーティリティを使用して表示したものです。

| Part | Tag | Flag | Cylinders | Size | Blocks |
|------|------------|------|-------------|----------|--------------------|
| 0 | root | wm | 0 - 292 | 80.12MB | (293/0/0) 164080 |
| 1 | swap | wu | 293 - 410 | 32.27MB | (118/0/0) 66080 |
| 2 | backup | wm | 0 - 1865 | 510.23MB | (1866/0/0) 1044960 |
| 3 | unassigned | wm | 0 | 0 | (0/0/0) 0 |
| 4 | unassigned | wm | 0 | 0 | (0/0/0) 0 |
| 5 | home | wm | 411 - 1311 | 246.37MB | (901/0/0) 504560 |
| 6 | usr | wm | 1312 - 1718 | 111.29MB | (407/0/0) 227920 |
| 7 | unassigned | wm | 1719 - 1865 | 40.20MB | (147/0/0) 82320 |

このパーティションテーブルには、次の情報が入っています。

| カラム名 | 説明 |
|-----------|--|
| Part | パーティション (またはスライス番号)。このカラムについての説明は、表 21-7を参照。 |
| Tag | パーティションのタグ。このカラムについての説明は、表 21-7を参照。 |
| Flags | パーティションのフラグ。このカラムについての説明は、表 21-7を参照。 |
| Cylinders | スライスの開始シリンダ番号と終了シリンダ番号を示す。 |
| Size | スライスのサイズを M バイト単位で示す。 |
| Blocks | 合計シリンダ数と 1 スライス当たりの合計セクター数 (カラムの右端) を示す。 |

次の例では、`prtvtoc` コマンドを使用してディスクラベルを表示します。

```
# prtvtoc /dev/rdisk/c0t3d0s0
* /dev/rdisk/c0t3d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   80 sectors/track
*   9 tracks/cylinder
*   720 sectors/cylinder
*   2500 cylinders
*   1151 accessible cylinders
*
* Flags:
*   1: unmountable
```

(続く)

```

* 10: read-only
*
*
*           First  Sector  Last
* Partition Tag  Flags Sector  Count  Sector  Mount Directory
  0      2    00     0    66240  66239  /
  1      3    01   66240  131760 197999
  2      5    00     0   828720 828719
  3      0    00  198000   66240 264239  /export
  4      0    00  264240   66240 330479  /export/swap
  5      0    00  330480   72000 402479  /opt
  6      4    00  402480  385200 787679  /usr
  7      8    00  787680   41040 828719  /export/home
#

```

このディスクラベルには、次の情報が入っています。

Dimensions – このセクションには、ディスクドライブの物理的な構成が示されます。

Flags – このセクションには、パーティションテーブルのセクションに記載されたフラグが記述されます。パーティションフラグについての説明は、表 21-7 を参照してください。

パーティション (またはスライス) テーブル – このセクションには次の情報が入っています。

| カラム名 | 説明 |
|--------------|--|
| Partition | パーティション (またはスライス番号)。このカラムについての説明は、表 21-7を参照。 |
| Tag | パーティションのタグ。このカラムについての説明は、表 21-7 を参照。 |
| Flags | パーティションのフラグ。このカラムについての説明は、表 21-7 を参照。 |
| First Sector | スライスの最初のセクターを示す。 |
| Sector Count | スライス内の合計セクター数を示す。 |

| | |
|-----------------|---------------------------------|
| Last Sector | スライス内の最後のセクター番号を示す。 |
| Mount Directory | ファイルシステムの最後のマウントポイントのディレクトリを示す。 |

ディスクをスライスに分割する

`format` ユーティリティは、主にシステム管理者がディスクをスライスに分割する場合に使われます。その場合の手順は次のとおりです。

- どのスライスが必要かを決定する。
- 各スライスのサイズを決定する。
- `format` ユーティリティを使用してディスクをスライスに分割する。
- 新しいスライス情報を使用してディスクにラベルを付ける。
- スライスごとにファイルシステムを作成する。

ディスクをスライスに分割するには、`partition` メニューから `modify` コマンドを使用するのが最も簡単な方法です。`modify` コマンドを使用すると、開始シリンダ境界を追跡しなくても、各スライスのサイズを M バイト単位で指定してスライスを作成できます。また、「free hog」スライス内の残りのディスク領域を追跡します。

free hog スライスの使用方法

`format` ユーティリティを使用してディスクスライスのサイズを変更するときには、サイズ変更操作に対応して拡大縮小する一時スライスを指定します。

このスライスは、スライスを拡大すると領域を「解放 (free)」し、スライスを圧縮すると放棄された領域を「回収 (hog)」します。このため、提供側のスライスを「free hog」と呼びます。

提供側のスライスは、インストール時または `format` ユーティリティの実行時にのみ存在します。通常の日常操作中に提供側スライスが継続して存在することはありません。

free hog スライスの使用方法についての詳細は、385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 405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。

ディスク管理の手順

この章では、ディスク管理の手順について説明します。Solaris システム上でディスクを管理する方法に精通している場合は、この章で説明する多くの内容を読み飛ばすことができます。

この章で説明する手順は次のとおりです。

- 355ページの「システム上のディスクを確認する方法」
- 358ページの「ディスクがフォーマット済みかどうかを調べる方法」
- 358ページの「ディスクをフォーマットする方法」
- 360ページの「ディスクスライス情報を表示する方法」
- 363ページの「ディスクラベルを作成する方法」
- 365ページの「ディスクラベルを検査する方法」
- 366ページの「破損したディスクラベルを復元する方法」
- 371ページの「format.dat のエントリを作成する方法」
- 372ページの「SCSI ドライブを自動構成する方法」
- 375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」
- 376ページの「欠陥セクターを修復する方法」

ディスク管理の概要については、第 21 章を参照してください。

ディスクの管理

表 22-1 作業マップ: ディスクの管理

| 作業 | 説明 | 手順の説明 |
|-------------------------------------|---|--|
| 1. システム上のディスクの確認 | システム上のディスクの種類が分からない場合は、 <code>format</code> ユーティリティを使用して確認する。 | 355ページの「システム上のディスクを確認する方法」 |
| 2. ディスクのフォーマット | <code>format</code> ユーティリティを使用して、ディスクがフォーマット済みかどうかを判断する。 ほとんどの場合、ディスクはフォーマット済みである。フォーマットする必要がある場合は、 <code>format</code> を使用する。 | 358ページの「ディスクがフォーマット済みかどうかを調べる方法」 358ページの「ディスクをフォーマットする方法」 |
| 3. スライス情報の表示 | <code>format</code> ユーティリティを使用してスライス情報を表示する。 | 360ページの「ディスクスライス情報を表示する方法」 |
| 4. ディスクラベルの作成 | <code>format</code> ユーティリティを使用してディスクラベルを作成する。 | 363ページの「ディスクラベルを作成する方法」 |
| 5. ディスクラベルの検査 | <code>prtvtoc</code> コマンドを使用してディスクラベルを検査する。 | 365ページの「ディスクラベルを検査する方法」 |
| 6. <code>format.dat</code> のエントリの作成 | サードパーティのディスクをサポートするために <code>format.dat</code> のエントリを作成する。 | 371ページの「 <code>format.dat</code> のエントリを作成する方法」 |
| 7. 欠陥ディスクセクターの検出 | <code>format</code> ユーティリティを使用して欠陥ディスクセクターを調べる。 | 375ページの「表面解析を使用して欠陥セクターを調べる方法」 |
| 8. 欠陥ディスクセクターの修復 (必要な場合) | <code>format</code> ユーティリティを使用して欠陥ディスクセクターを修復する。 | 376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 25 章を参照してください。

▼ システム上のディスクを確認する方法

1. スーパーユーザーになります。
2. format ユーティリティを使用して、システム上で認識されるディスクを確認します。

```
# format
```

format ユーティリティは、AVAILABLE DISK SELECTIONS という見出しの下に、認識されるディスクのリストを表示します。

例 — システム上のディスクを確認する

次の format 出力は、2 つのディスクを持つシステムのものであります。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number):
```

format の出力は、ディスクの物理デバイス名と論理デバイス名を括弧 <> 内の商品名に対応させています。このため、どの論理名がシステムに接続されたディスクを表しているかを一目で識別できます。論理デバイス名と物理デバイス名については、第 20 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/rdsk/c2t0d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@0,0
 1. /dev/rdsk/c2t1d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@1,0
 2. /dev/rdsk/c2t2d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@2,0
 3. /dev/rdsk/c2t3d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@3,0
 4. /dev/rdsk/c2t5d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@5,0
Specify disk (enter its number):
```

次の例では、SPARC システム上のディスクを表示します。

```
# format
AVAILABLE DISK SELECTIONS:
 0. c0t3d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
    /iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0
Specify disk (enter its number):
```

format の出力は、ディスク 0 (ターゲット 3) が第 1 の SCSI ホストアダプタ (espdma@...) に接続されており、そのホストアダプタは第 1 の SBus デバイス (sbus@0...) に接続されていることを示しています。また、この出力は物理デバイス名と論理デバイス名をディスクの商品名 SUN2.1G に対応しています。

次の例では、x86 システム上のディスクを表示します。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. c0t0d0 <DEFAULT cyl 507 alt 2 hd 64 sec 32>
    /eisa/dpt@5c88,0/cmdk@0,0
 1. c0t3d0 <DEFAULT cyl 1852 alt 2 hd 15 sec 74>
    /eisa/dpt@5c88,0/cmdk@3,0
Specify disk (enter its number):
```

format の出力は、ディスク 0、ターゲット 0 (cmdk@0,0) が最初の DPT ホストアダプタ (dpt@5...) に接続され、このアダプタが EISA デバイス (eisa) に接続されていることを示します。x86 システムに関する format の出力では、ディスクは商品名では表示されません。

次に進む手順

format ユーティリティでディスクが認識されなかった場合は、次の表を参照してください。

| ディスクの状態 | 参照先 |
|----------------------------|---------------------------------------|
| 新しく追加したが、再構成ブートを実行しなかった | 第 23 章または第 24 章 |
| サードパーティのディスク | 370ページの「format.dat のエントリの作成」 |
| 電源障害など、システムの問題によってラベルが破損した | 363ページの「ディスクラベルを作成する方法」 |
| システムに正しく接続されていない | ディスクのハードウェアマニュアルを参照して、ディスクをシステムに接続する。 |

ディスクのフォーマット

ディスクはメーカーまたは再販業者によってフォーマットされているので、通常はフォーマットしなくてもドライブをインストールできます。

次の作業の前にディスクをフォーマットしておかなければなりません。

- データの書き込み。ただし、ほとんどのディスクはフォーマット済みです。
- Solaris インストールプログラムを使用して行うシステムのインストール



注意 - フォーマットはディスク上のデータを上書きします。このため、通常はメーカーや再販業者のみがディスクをフォーマットします。ディスクに欠陥があるために問題が再発していると思われる場合は、format ユーティリティを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。

▼ ディスクがフォーマット済みかどうかを調べる方法

1. スーパーユーザーになります。
2. `format` ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、チェックしたいディスクの番号を入力します。

```
Specify disk (enter its number): 0
```

4. ディスクがフォーマット済みかどうかを調べます。選択したディスクがフォーマット済みであれば、次のメッセージが表示されます。

```
[disk formatted]
```

例 — ディスクがフォーマット済みかどうかを調べる

次の例は、ディスク `c0t3d0` がフォーマット済みであることを示しています。

```
# format
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
```

▼ ディスクをフォーマットする方法

1. スーパーユーザーになります。
2. `format` ユーティリティを起動します。

```
# format
```


- 画面に表示されるリストから、フォーマットしたいディスクの番号を入力します。

```
Specify disk (enter its number): 0
```



警告 - システムディスクを選択しないでください。システムディスクをフォーマットすると、オペレーティングシステムや、このディスクに入っているデータが削除されます。

- ディスクのフォーマットを開始するには、`format>` プロンプトで `format` と入力します。`y` と入力してコマンドを確認します。

```
format> format  
Ready to format. Formatting cannot be interrupted  
and takes 26 minutes (estimated). Continue? y
```

- 次のメッセージによりフォーマットが正常に行われたことを確認します。

```
Formatting ...  
done  
Verifying media ...  
pass 0 - pattern = 0xc6dec6de  
2035/12/18  
pass 1 - pattern = 0x6db6db6d  
2035/12/18  
total of 0 defective blocks repaired.
```

例 — ディスクをフォーマットする

次の例では、ディスク `c0t2d0` をフォーマットします。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number):0
Selecting c0t1d0
[disk unformatted]
format> format
Ready to format. Formatting cannot be interrupted
and takes 26 minutes (estimated). Continue? yes
Beginning format. The current time is Wed Jun 10 10:24:48 1998
Formatting ...
done
Verifying media ...
pass 0 - pattern = 0xc6dec6de
2035/12/18
pass 1 - pattern = 0x6db6db6d
2035/12/18
total of 0 defective blocks repaired.
format>
```

ディスクスライスの表示

format ユーティリティを使用すると、ディスクに適切なディスクスライスがあるかどうかをチェックできます。使用したいスライスがディスクに入っていないことが判明した場合は、format ユーティリティを使用して作成し直し、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、385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 405ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

注 - format ユーティリティでは、スライスではなくパーティションという用語を使用します。

▼ ディスクスライス情報を表示する方法

1. スーパーユーザーになります。
2. format ユーティリティを起動します。

```
# format
```

3. AVAILABLE DISK SELECTIONS という見出しの下に表示されたディスクを選択して、スライス情報を表示したいディスクを指定します。

```
Specify disk (enter its number):1
```

4. format> プロンプトで partition と入力して、パーティションメニューに入ります。

```
format> partition
```

5. partition> プロンプトで print と入力して、現在のディスクドライブのスライス情報を表示します。

```
partition> print
```

6. partition> プロンプトで q と入力し、format プロンプトで q と入力して、format> ユーティリティを終了します。

```
partition> q
format> q
#
```

7. スライス情報が特定のスライスのタグとサイズにより表示されることを確認します。

画面の出力が、スライスサイズが割り当てられていないことを示している場合は、ディスクにスライスがないものと思われます。

例 — ディスクスライス情報を表示する

次の例では、ディスク /dev/rdisk/c0t3d0 のスライス情報を表示します。

```

# format
Searching for disks...done
Specify disk (enter its number):1
Selecting c0t3d0
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 1866 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0         root     wm        0 - 292      80.12MB   (293/0/0)   164080
1         swap     wu        293 - 410    32.27MB   (118/0/0)   66080
2         backup   wm        0 - 1865    510.23MB  (1866/0/0) 1044960
3 unassigned wm        0           0          (0/0/0)     0
4 unassigned wm        0           0          (0/0/0)     0
5         home     wm        411 - 1311   246.37MB  (901/0/0)   504560
6         usr      wm        1312 - 1718  111.29MB  (407/0/0)   227920
7 unassigned wm        1719 - 1865  40.20MB   (147/0/0)   82320
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 21 章を参照してください。

次の例では、ディスク /dev/rdisk/c0t0d0 のスライス情報を表示します。

```

# format
Searching for disks...done
Specify disk (enter its number): 0
selecting c0t0d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 1479 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0         root     wm        1 - 400      108.01MB  (400/0/0)  221200
1         swap     wu        401 - 556    42.12MB   (156/0/0)   86268
2         backup   wu        0 - 1479    399.63MB  (1480/0/0) 818440
3 unassigned wm        0           0          (0/0/0)     0
4 unassigned wm        0           0          (0/0/0)     0
5 unassigned wm        557 - 734    48.06MB   (178/0/0)   98434
6         usr      wm        735 - 1401   180.10MB  (667/0/0)  368851
7         home     wm        1402 - 1476  20.25MB   (75/0/0)   41475
8         boot     wu        0 - 0        0.27MB    (1/0/0)     553
9 unassigned wm        0           0          (0/0/0)     0
partition> q
format> q

```

ディスクラベルの作成と検査

一般に、ディスクにラベルを付ける操作は、システムのインストール時、または新しいディスクスライスを作成するときに行います。電源障害などの自然災害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。

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せん。ラベルが付いていないディスクを自動構成できる場合は、次のようなメッセージが表示されます。

```
clt0d0: configured with capacity of 404.65MB
```

▼ ディスクラベルを作成する方法

1. スーパーユーザーになります。
2. `format` ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、ラベルを作成したいディスクの番号を入力します。

```
Specify disk (enter its number):1
```

4. ディスクラベルの作成方法を決定します。

ディスクにラベルが付いておらず、正常に自動構成された場合

ディスクにラベルが付いており、そのタイプを変更したいか、**format** でディスクを自動構成できなかった場合

ディスクラベルを作成したいかどうかを尋ねるプロンプトが表示される。手順5に進んでラベルを作成する。

ディスクタイプを指定しなければならない。手順6と7に進んでディスクのタイプを設定し、ラベルを付ける。

5. `Label it now?` プロンプトで `y` と入力して、ディスクにラベルを付けます。

```
Disk not labeled. Label it now? y
```

これでディスクラベルが作成されました。手順 9 に進んで format ユーティリティを終了します。

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) [8]: 8
```

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
```

それ以外の場合は、次のメッセージが表示されます。

```
Ready to label disk, continue? y
```

9. format のメインメニューから verify コマンドを使用してディスクラベルを検査します。

```
format> verify
```

10. format> プロンプトで q と入力して format ユーティリティを終了します。

```
partition> q  
format> q  
#
```

例 — ディスクラベルを作成する

次の例では、424M バイトのディスクを自動構成してラベルを付けます。

```
# format
c1t0d0: configured with capacity of 404.65MB

AVAILABLE DISK SELECTIONS:
  0. c0t3d0 <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
     /sbus@1,f8000000/esp@0,800000/sd@3,0
  1. c1t0d0 <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
     /sbus@1,f8000000/QLGC,isp@1,10000/sd@0,0
Specify disk (enter its number): 1
Disk not labeled. Label it now? yes
format> verify
#
```

▼ ディスクラベルを検査する方法

`prtvtoc(1M)` コマンドを使用して、ディスクラベル情報を検査します。ディスクラベルの説明と `prtvtoc` コマンドで表示される情報については、第 21 章を参照してください。

1. スーパーユーザーになります。
2. `prtvtoc` コマンドを使用してディスクラベル情報を表示します。

```
# prtvtoc /dev/rdisk/device-name
```

device-name 検査したい raw ディスクデバイス

例 — ディスクラベルを検査する

次の例は、ディスク `/dev/rdisk/c0t0d0s0` のディスクラベル情報を示しています。

```
# prtvtoc /dev/rdisk/c0t0d0s0 * c0t0d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   36 sectors/track
*   9 tracks/cylinder
*   324 sectors/cylinder
```

```

*      1272 cylinders
*      1254 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
*      First   Sector   Last
* Partition Tag  Flags  Sector  Count  Sector  Mount Directory
*   0         2    00      0      37260  37259  /
*   1         3    01    37260   77760 115019
*   2         5    00      0    406296 406295
*   6         4    00   115020 283824 398843  /usr
*   7         6    00   398844   7452  406295  /export/home
#

```

破損したディスクラベルの復元

電源障害やシステム障害が原因で、ディスクが認識されなくなることがあります。このような場合に、かならずしもスライス情報やディスクのデータを作成し直したり、復元しなければならないとは限りません。

破損したディスクラベルを復元する作業の最初の手順は、正しいジオメトリとディスクタイプ情報を使用してディスクにラベルを付けることです。この操作は、通常のディスクラベル作成方法を使用して実行できます。つまり、自動構成するか、またはディスクタイプを手作業で指定します。

`format` でディスクタイプが認識されたら、次の手順はバックアップラベルを検索してディスクにラベルを付けることです。バックアップラベルを使用してディスクにラベルを付けると、ディスクタイプとジオメトリだけでなく、正しいパーティション情報を使用してディスクラベルが作成されます。

▼ 破損したディスクラベルを復元する方法

1. システムをシングルユーザーモードにします。必要であれば、シングルユーザーモードでローカル **CD-ROM** またはネットワークからシステムをブートして、ディスクにアクセスします。

システムをブートする方法については、第 8 章または第 9 章を参照してください。

2. `format` ユーティリティを使用してディスクのラベルを作成し直します。

```
# format
```

この時点で、`format` はラベルが付いていない SCSI ディスクを自動構成しようとして、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。

```
cwtxy: configured with capacity of xyzMB
```

次に、システム上のディスクのリストが表示されます。

3. 画面に表示されたリストから、復元したいディスクの番号を入力します。

```
Specify disk (enter its number): 1
```

4. ディスクラベルの作成方法を決定します。

ディスクが正常に自動構成された場合

ディスクが正常に自動構成されなかった場合

手順 5 と 6 を実行してから、手順 12 に進む。

手順 7 から 11 までを実行してから手順 12 に進む。

5. `verify` コマンドを使用してバックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk or use the
'backup' command.
Backup label contents:
Volume name = <          >
ascii name  = <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
pcyl        = 2500
ncyl        = 1151
acyl        = 2
nhead       = 9
nsect       = 80
Part        Tag   Flag   Cylinders      Size      Blocks
0           root  wm      0 - 91    32.34MB    (92/0/0)  66240
```

(続く)

続き

| | | | | | | |
|---|------------|----|------------|----------|------------|--------|
| 1 | swap | wu | 92 - 183 | 32.34MB | (92/0/0) | 66240 |
| 2 | backup | wu | 0 - 1150 | 404.65MB | (1151/0/0) | 828720 |
| 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 | 184 - 1150 | 339.96MB | (967/0/0) | 696240 |
| 7 | unassigned | wm | 0 | 0 | (0/0/0) | 0 |

6. `format` でバックアップラベルが見つかり、バックアップラベルの内容が適切に表示された場合は、`backup` コマンドを実行し、バックアップラベルを使用してディスクにラベルをつけます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

ディスクラベルが復元されました。手順 12 へ進みます。

7. `format` でディスクを自動構成できなかった場合は、`type` コマンドを使用してディスクタイプを指定します。

```
format> type
```

Available Devices Type メニューが表示されます。

8. 0 を選択してディスクを自動構成するか、またはディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number) [8]: 8
```

9. ディスクが正常に自動構成された場合は、ディスクラベルを作成するかどうか尋ねるプロンプトが表示されたときに `no` と応答します。

```
Disk not labeled. Label it now? no
```

10. `verify` コマンドを使用してバックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
.
.
.
```

11. `format` でバックアップラベルが見つかり、その内容が適切な場合は、`backup` コマンドを実行し、バックアップラベルを使用してディスクにラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

ディスクラベルが復元されました。

12. `q` と入力して `format` ユーティリティを終了します。

```
format> q
```

13. `fsck` コマンドを使用して、復元されたディスク上のファイルシステムを確認します。

`fsck` コマンドの使用方法については、第 31 章を参照してください。

サードパーティのディスクの追加

Solaris 環境では、サードパーティの多数のディスクがサポートされます。ただし、デバイスドライバ、`format.dat`、またはその両方を用意しなければならない場合があります。

サードパーティのディスクが標準の SunOS オペレーティングシステム互換デバイスドライバで機能するように設計されている場合は、適切な `format.dat` エントリを作成するだけで、ディスクは `format` ユーティリティに認識されるはずですが、それ以外の場合は、そのディスクをサポートするためにサードパーティのデバイスドライバをロードする必要があります。

注 - Sun の `format` ユーティリティがサードパーティのどのディスクドライバでも正常に機能するとは限りません。ディスクドライバに Solaris の `format` ユーティリティとの互換性がない場合は、ディスクドライブのベンダーが独自のフォーマットプログラムを提供しているはずですが。

ここでは、ソフトウェアサポートのいずれかが不足している場合に必要な作業について説明します。一般に、`format` ユーティリティを起動し、ディスクタイプが認識されないなどという場合に、不足しているソフトウェアサポートがあることがわかります。

この節の説明にしたがって不足しているソフトウェアを用意してから、第 23 章または第 24 章で説明されている、システムディスクまたは二次ディスクを構成する手順を参照してください。

format.dat のエントリの作成

認識されないディスクは、そのディスクのジオメトリと運用パラメータに関する正確な情報がなければフォーマットできません。この情報は、`/etc/format.dat` ファイル内で指定します。

注 - SCSI-2 ドライブには `format.dat` のエントリは不要です。Solaris 2.3 からは、再構成ブート時にドライブの電源が入っていれば、`format` ユーティリティは SCSI-2 ドライブを自動的に構成します。SCSI ドライブを自動的に構成する手順については、372ページの「SCSI ドライブを自動構成する方法」を参照してください。

ディスクが認識されない場合は、テキストエディタを使用して `format.dat` にディスクのエントリを作成します。作業を始める前に、ディスクとそのコントローラに関連するすべての技術仕様を収集する必要があります。この情報はディスクといっしょに提供されているはずですが、提供されない場合は、ディスクメーカーまたは購入先に問い合わせてください。`/etc/format.dat` ファイルにエントリを追加する方法については、第 25 章を参照してください。

▼ format.dat のエントリを作成する方法

1. スーパーユーザーになります。
2. /etc/format.dat ファイルのコピーを作成します。

```
# cp /etc/format.dat /etc/format.dat.gen
```

3. 第 25 章で説明する format.dat 情報を使用して、/etc/format.dat ファイルに、サードパーティのディスクのエントリを入力します。
ディスクのハードウェア製品マニュアルを参照して、必要な情報を収集してください。

SCSI ディスクドライブの自動構成

Solaris 2.3 およびその互換バージョンでは、/etc/format.dat ファイルに特定のドライブタイプが含まれていない場合でも、format ユーティリティは SCSI ディスクドライブを自動的に構成します。この機能により、ディスクデバイスモード検知に関する SCSI-2 仕様に準拠しているディスクドライブで、フォーマット、スライスへの分割、ラベルの作成が行えます。

次の手順にしたがって、自動構成により SCSI ドライブを構成します。

- システムをシャットダウンする。
- SCSI ディスクドライブをシステムに接続する。
- ディスクドライブの電源をオンにする。
- 再構成用ブートを実行する。
- format ユーティリティを使用して SCSI ディスクドライブを自動構成する。

再構成ブートを実行した後に、format ユーティリティを呼び出すと、format はディスクを構成しようとします。成功すると、ディスクが構成されたことを示すメッセージを表示します。SCSI ディスクドライブを自動構成する手順については、372ページの「SCSI ドライブを自動構成する方法」を参照してください。

次の表は、format がパーティションテーブルの作成に使用する規則を示しています。

表 22-2 SCSI ディスクスライスの規則

| ディスクサイズ | ルートファイルシステム | スワップスライス |
|-----------------|-------------|----------|
| 0 ~ 180M バイト | 16M バイト | 16M バイト |
| 180 ~ 280M バイト | 16M バイト | 32M バイト |
| 280 ~ 380M バイト | 24M バイト | 32M バイト |
| 380 ~ 600M バイト | 32M バイト | 32M バイト |
| 600M ~ 1.0G バイト | 32M バイト | 64M バイト |
| 1.0 ~ 2.0G バイト | 64M バイト | 128M バイト |
| 2.0G バイト ~ | 128M バイト | 128M バイト |

いずれの場合も、スライス 6 (/usr ファイルシステム) にディスク上の残りの領域が割り当てられます。

1.3G バイトの SCSI ディスクドライブに関して format で生成されるパーティションテーブルを示しています。

| Part | Tag | Flag | Cylinders | Size | Blocks |
|------|--------|------|------------|----------|------------|
| 0 | root | wm | 0 - 96 | 64.41MB | (97/0/0) |
| 1 | swap | wu | 97 - 289 | 128.16MB | (193/0/0) |
| 2 | backup | wu | 0 - 1964 | 1.27GB | (1965/0/0) |
| 6 | usr | wm | 290 - 1964 | 1.09GB | (1675/0/0) |

SCSI 自動構成機能の使用方法についての詳細は、第 25 章を参照してください。

▼ SCSI ドライブを自動構成する方法

1. スーパーユーザーになります。
2. システムのブート時に読み込まれる /reconfigure ファイルを作成します。

```
# touch /reconfigure
```

3. システムをシャットダウンします。

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

-i0 システムを init 状態 0 (電源切断) にする。

-g30 ログインしているユーザーに、30 秒後にシステムのシャットダウンを開始することを通知する。

-y ユーザーの介入なしでコマンドを実行するように指定する。

運用環境がシャットダウンされると、ok または > プロンプトが表示されます。

4. システムとすべての外部周辺デバイスの電源を切ります。
5. 追加しようとするディスクに、システム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、この操作のためにディスクの裏側に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チェックします。
インストールについての詳細は、ディスクのハードウェアインストールガイドを参照してください。
7. すべての外部周辺デバイスの電源を入れます。
8. システムの電源を入れます。
システムがブートされ、ログインプロンプトが表示されます。
9. スーパーユーザーとしてログインし、format ユーティリティを呼び出して、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 404.65MB
AVAILABLE DISK SELECTIONS:
  0. c0t3d0 <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
     /sbus@1,f8000000/esp@0,800000/sd@3,0
  1. c1t0d0 <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
```

(続く)

続き

```
/sbus@1,f8000000/QLGC,isp@1,10000/sd@0,0  
Specify disk (enter its number): 1
```

10. プロンプトで `yes` と入力してディスクにラベルを付けます。

`y` と入力すると、自動構成機能によってディスクラベルが生成され、ディスクに書き込まれます。

```
Disk not labeled. Label it now? y
```

11. `verify` コマンドを使用してディスクラベルを確認します。

```
format> verify
```

12. `format` ユーティリティを終了します。

```
format> q
```

欠陥セクターの修復

システム上のディスクに欠陥セクターが存在する場合は、次の手順にしたがって修復できます。欠陥セクターを発見するのは次のような場合です。

- ディスク上で表面解析を実行した場合

`format` ユーティリティの解析機能については、416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、375ページの「表面解析を使用して欠陥セクターを調べる方法」を参照してください。

- システムの実行中に、ディスクドライバからディスクの特定部分に関して多数のエラーメッセージが表示される場合

ディスクエラーに関連するメッセージは次のよう出力されます。

```
WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):  
Error for command 'read' Error Level: Retryable  
Requested Block 126, Error Block: 179  
Sense Key: Media Error  
Vendor 'SEAGATE':  
ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
```

上記のコンソールメッセージは、ブロック 179 が不良であり、format ユーティリティの repair コマンドを使用して配置し直すか、または修復機能を使用可能にして表面解析を実行する必要があることを示しています。

▼ 表面解析を使用して欠陥セクターを調べる方法

1. スーパーユーザーになります。
2. 欠陥セクターの存在するスライス内のファイルシステムをマウント解除します。
詳細は、mount (1M) のマニュアルページを参照してください。

```
# umount /dev/dsk/device-name
```

3. format と入力して format ユーティリティを起動します。

```
# format
```

4. 調べるディスクを選択します。

```
Specify disk (enter its number):1  
selecting c0t2d0:  
[disk formatted]  
Warning: Current Disk has mounted partitions.
```

5. format> プロンプトで analyze と入力して、**analyze** メニューに入ります。

```
format> analyze
```

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 [no]? n
Enter starting block number [0, 0/0/0]: 12330
Enter ending block number [584159, 1216/9/47]: 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 [31, 0/0/31]: 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
      25/7/24
    pass 1
Block 12354 (18/4/18), Corrected media error (hard data ecc)
      25/7/24
^C
Total of 1 defective blocks repaired.
```

▼ 欠陥セクターを修復する方法

1. スーパーユーザーになります。

- format ユーティリティに入り、欠陥セクターの存在するディスクを選択します。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t2d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@2,0
  1. c0t3d0 <SUN0535 cyl 1866 alt 2 hd 7 sec 80>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 1
selecting c0t3d0
[disk formatted]
format>
```

- format> プロンプトで repair コマンドを入力します。

```
format> repair
```

- 欠陥ブロック番号を入力します。

```
Enter absolute block number of defect: 12354
Ready to repair defect, continue? y
Repairing block 12354 (18/4/18)...ok.
format>
```

format を使った欠陥セクターの識別方法がわからない場合は、375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。

ヒント

次のヒントに従って、ディスクの管理効率を高めることができます。

format セッションのデバッグ

format -M と入力して、SCSI デバイスに対してのみ有効な拡張メッセージと診断メッセージを出力するようにします。

次の例では、Inquiry: の下の一連の数字は、その右側に表示されている inquiry データの 16 進値を表わします。

```
# format -M
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c0t3d0 <SUN0535 cyl 1866 alt 2 hd 7 sec 80>
    /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 27 00 00 12 43 4f 4e 4e 45 52 20 20      ....'...CONNER
43 50 33 30 35 34 30 20 20 53 55 4e 30 35 33 35      CP30540  SUN0535
42 30 42 42 39 33 30 38 46 39 30                      BOBB9308F90
Vendor:      CONNER
Product:     CP30540  SUN0535
Revision:    BOBB
format>
```

prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける

prtvtoc コマンドと fmthard コマンドを使用して、同じディスクジオメトリを持つ複数のディスクにラベルを付けます。

この for ループをスクリプト内で使用して、1 台のディスクからディスクラベルをコピーし、複数のディスク上で複製します。

```
# for i in x y z
> do
> prtvtoc /dev/rdisk/cwtxdysz | fmthard -s - /dev/rdisk/cwt${i}d0s2
> done
```

例 — 複数のディスクにラベルを付ける

この例では、ディスクラベルがディスク c2t0d0s0 から他の 4 台のディスクにコピーされます。

```
# for i in 1 2 3 5
> do
> prtvtoc /dev/rdisk/c2t0d0s0 | fmthard -s - /dev/rdisk/c2t${i}d0s2
> done
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
fmthard: New volume table of contents now in place.
#
```


SPARC: ディスク追加の手順

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。

この章で説明する手順は次のとおりです。

- 383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」
- 384ページの「SPARC: 二次ディスクを接続してブートする方法」
- 385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」
- 390ページの「SPARC: ファイルシステムを作成する方法」
- 391ページの「SPARC: システムディスクにブートブロックをインストールする方法」

ディスク管理の概要については、第 21 章を参照してください。

SPARC: システムディスクと二次ディスクについて

システムディスクには、ルート (/) ファイルシステム、/usr ファイルシステム、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法には次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップ媒体から復元する。

二次ディスクには、ルート (/) ファイルシステムも /usr ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上で復元できます。

SPARC: システムディスクまたは二次ディスクの追加

表 23-1 作業マップ: システムディスクまたは二次ディスクの追加

| 作業 | 説明 | 手順の説明 |
|--------------------|--|---|
| 1. ディスクの接続とブート | <p>システムディスク</p> <p>新しいディスクを接続して、ローカルまたはリモートの Solaris CD からブートする。</p> <p>二次ディスク</p> <p>新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行する。</p> | <p>383ページの「SPARC: システムディスクを接続してブートする方法」</p> <p>384ページの「SPARC: 二次ディスクを接続してブートする方法」</p> |
| 2. スライスとディスクラベルの作成 | <p>まだディスクメーカーによって実行されていない場合は、ディスクスライスを作成してディスクにラベルを付ける。</p> | <p>385ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」</p> |
| 3. ファイルシステムの作成 | <p>newfs コマンドを使用してディスクスライス上に UFS ファイルシステムを作成する。システムディスクの場合はルート (/) と /usr ファイルシステムを作成しなければならない。</p> | <p>390ページの「SPARC: ファイルシステムを作成する方法」</p> |

表 23-1 作業マップ: システムディスクまたは二次ディスクの追加 続く

| 作業 | 説明 | 手順の説明 |
|-------------------|--|---|
| 4. ファイルシステムの復元 | システムディスク上にルート (/) と /usr ファイルシステムを復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。 | 第 35 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。 | 391ページの「SPARC: システムディスクにブートブロックをインストールする方法」 |

▼ SPARC: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

1. 損傷したシステムディスクをシステムから外します。
2. 追加しようとするディスクにシステム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの裏側にそのための小型スイッチが付いています。
3. 交換用のディスクがシステムに正しく接続されているかどうかをチェックします。
インストールについての詳細は、ディスクのハードウェアインストールガイドを参照してください。
4. ローカルとリモートのどちらの **Solaris CD** からブートするかに応じて、次の表の手順で操作します。

| ブート元 | 操作 |
|--------------------------------------|--|
| ローカル CD-ROM ドライブ内の Solaris CD から | 1. CD が CD-ROM ドライブに入っているかどうかを確認する。 2. CD からシングルユーザーモードでブートする。 ok boot cdrom -s |
| ネットワーク上の CD-ROM ドライブ内の Solaris CD から | ネットワークからシングルユーザーモードでブートする。 ok boot net -s |

数分後に root プロンプト (#) が表示されます。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。385ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ SPARC: 二次ディスクを接続してブートする方法

1. スーパーユーザーになります。
2. ディスクタイプが **Solaris** ソフトウェアでサポートされない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。

このディスク用の `format.dat` のエントリの作成方法については、371ページの「`format.dat` のエントリを作成する方法」を参照してください。

3. システムのブート時に読み込まれる `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

`/reconfigure` ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

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

| | |
|-----|--|
| -i0 | システムを init 状態 0 (電源切断) にする。 |
| -gn | ログインしているユーザーに、 <i>n</i> 秒後にシステムのシャットダウンを開始することを通知する。 |
| -y | ユーザーの介入なしでコマンドを実行するように指定する。 |

運用環境がシャットダウンされると、ok または > プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクにシステム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの裏側にそのための小型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チェックします。
インストールについての詳細は、ディスクのハードウェアインストールガイドを参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。
システムがブートし、ログインプロンプトを表示します。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。385ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ SPARC: ディスクスライスとディスクラベルを作成する方法

1. スーパーユーザーになります。

- 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。

- 画面に表示されるリストから、パーティションに分割し直したいディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number パーティションに分割し直したいディスクの番号

- partition メニューにアクセスします (スライスを設定できます)。

```
format> partition
```

- 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
```

- 変更作業を開始します。

```
partition> modify
```

- ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]? 1
```

free hog スライスについての詳細は、351ページの「free hog スライスの使用方法」を参照してください。

- 続行するかどうかを尋ねるプロンプトが表示されたら、y と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition table based on
above table[yes]?y
```

9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。
- システムディスクを追加するときは、次のどちらか、または両方のスライスを設定しなければなりません。
- ルート (スライス 0)、スワップ (スライス 1)
 - /usr (スライス 6)
- スライスを設定すると、新しいパーティションテーブルが表示されます。

10. プロンプトが表示されたら `y` と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? y
```

現在のパーティションテーブルが希望どおりでないために変更したい場合は、`no` と応答して386ページの手順6に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

`partition-name` 新しいパーティションテーブルの名前

12. 新しいディスク上のスライスを割り当て終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. `partition` メニューを終了します。

```
partition> q
```

14. verify コマンドを使用して、ディスクラベルを確認します。

```
format> verify
```

15. format メニューを終了します。

```
format> q
```

SPARC: 例 — システムディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティを使用して 1G バイトのディスクを 3 つのスライスに分割します。各スライスをルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムに割り当てます。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
        /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
    1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
        /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0              0      (0/0/0)      0
  1      swap      wu         0              0      (0/0/0)      0
  2  backup      wu    0 - 2035      1002.09MB    (2036/0/0) 2052288
  3 unassigned      wm         0              0      (0/0/0)      0
  4 unassigned      wm         0              0      (0/0/0)      0
  5 unassigned      wm         0              0      (0/0/0)      0
  6      usr      wm         0              0      (0/0/0)      0
  7 unassigned      wm         0              0      (0/0/0)      0
Do you wish to continue creating a new partition
table based on above table[yes]? yes
Free Hog partition[6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb]: 200mb
Enter size of partition '1' [0b, 0c, 0.00mb]: 200mb
Enter size of partition '3' [0b, 0c, 0.00mb]:
```

(続く)

続き

```
Enter size of partition `4` [0b, 0c, 0.00mb]:
Enter size of partition `6` [0b, 0c, 0.00mb]:
Enter size of partition `7` [0b, 0c, 0.00mb]:

Part      Tag      Flag      Cylinders      Size      Blocks
0        root      wm         0 - 406        200.32MB  (407/0/0)  410256
1         swap      wu        407 - 813        200.32MB  (407/0/0)  410256
2    backup      wu         0 - 2035       1002.09MB (2036/0/0) 2052288
3 unassigned      wm          0                0          (0/0/0)    0
4 unassigned      wm          0                0          (0/0/0)    0
5 unassigned      wm          0                0          (0/0/0)    0
6         usr      wm        814 - 2035       601.45MB (1222/0/0) 1231776
7 unassigned      wm          0                0          (0/0/0)    0

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "disk0"
Ready to label disk, continue? yes
partition> quit
format> verify
format> quit
```

SPARC: 例 — 二次ディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティを使用して1G バイトのディスクを /export/home ファイルシステム用の1つのスライスに割り当てます。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  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
```

(続く)

```

 2      backup    wu      0 - 2035    1002.09MB    (2036/0/0) 2052288
 3 unassigned   wm      0           0             (0/0/0)      0
 4 unassigned   wm      0           0             (0/0/0)      0
 5 unassigned   wm      0           0             (0/0/0)      0
 6      usr      wm      0           0             (0/0/0)      0
 7 unassigned   wm      0           0             (0/0/0)      0
Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 7
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
  Part      Tag      Flag      Cylinders      Size      Blocks
  0         root     wm        0                0      (0/0/0)      0
  1         swap     wu        0                0      (0/0/0)      0
  2      backup    wu      0 - 2035    1002.09MB    (2036/0/0) 2052288
  3 unassigned   wm      0                0      (0/0/0)      0
  4 unassigned   wm      0                0      (0/0/0)      0
  5 unassigned   wm      0                0      (0/0/0)      0
  6      usr      wm      0                0      (0/0/0)      0
  7 unassigned   wm      0 - 2035    1002.09MB    (2036/0/0) 2052288
Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "home"
Ready to label disk, continue? y
partition> q
format> verify
format> q
#

```

次に進む手順

ディスクスライスを作成してディスクのラベル付けが終わったら、ディスク上にファイルシステムを作成します。390ページの「SPARC: ファイルシステムを作成する方法」を参照してください。

▼ SPARC: ファイルシステムを作成する方法

1. スーパーユーザーになります。
2. `newfs` コマンドを使用して、スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```


/dev/rdisk/cwtxdysz

作成するファイルシステムの raw デバイス

newfs コマンドについての詳細は、第 27 章を参照してください。

3. 未使用のマウントポイントにマウントすることによって、新しいファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz/mnt
# ls
lost+found
```

参照先

| 追加するディスク | 参照先 |
|----------|---|
| システムディスク | ディスク上にルート (/) と /usr のファイルシステムを復元する必要がある。第 35 章を参照。 ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールする。391ページの「SPARC: システムディスクにブートブロックをインストールする方法」を参照。 |
| 二次ディスク | 新しいディスク上にファイルシステムを復元しなければならないことがある。第 35 章を参照。 新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が終わったことになる。ファイルシステムをユーザーが利用できるようにする方法については、第 28 章を参照。 |

▼ SPARC: システムディスクにブートブロックをインストールする方法

1. スーパーユーザーになります。
2. installboot コマンドを使用して、システムディスクにブートブロックをインストールします。

```
# /usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk  
/dev/rdisk/cwtxdys0
```

/usr/platform/`uname -i`/lib/fs /ufs/ ブートブロックコード
bootblk

/dev/rdisk/cwtxdys0 ルート (/) ファイルシステムの
raw デバイス

3. システムを実行レベル **3** にして、ブートブロックがインストールされていることを確認します。

```
# init 6
```

SPARC: 例 — システムディスクにブートブロックをインストールする

次の例では、SPARCstation 10 にブートブロックをインストールします。

```
# installboot /usr/platform/sun4m/lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0
```

x86: ディスク追加の手順

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。この章で説明する手順は次のとおりです。

- 395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」
- 396ページの「x86: 二次ディスクを接続してブートする方法」
- 398ページの「x86: Solaris fdisk パーティションを作成する方法」
- 405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」
- 408ページの「x86: ファイルシステムを作成する方法」
- 409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」

ディスク管理の概要については、第 21 章を参照してください。

x86: システムディスクと二次ディスクについて

システムディスクには、ルート (/) ファイルシステム、/usr ファイルシステム、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Solaris 環境全体をインストールし直す。
- システムディスクを交換し、ファイルシステムをバックアップ媒体から復元する。

二次ディスクには、ルート (/) ファイルシステムも /usr ファイルシステムも入っていません。通常はユーザーファイル用の領域が入っています。システムに二次

ディスクを追加してディスク容量を増やしたり、損傷した二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。

x86: システムディスクまたは二次ディスクの追加

表 24-1 作業マップ: システムディスクまたは二次ディスクの追加

| 作業 | 説明 | 手順の説明 |
|--------------------|---|---|
| 1. ディスクの接続とブート | <p>システムディスク</p> <p>新しいディスクを接続して、ローカルまたはリモートの Solaris CD からブートする</p> <p>二次ディスク</p> <p>新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行する。</p> | <p>383ページの「SPARC: システムディスクを接続してブートする方法」</p> <p>384ページの「SPARC: 二次ディスクを接続してブートする方法」</p> |
| 2. スライスとディスクラベルの作成 | <p>まだディスクメーカーによって実行されていない場合は、ディスクスライスとディスクラベルを作成する。</p> | <p>385ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」</p> |
| 3. ファイルシステムの作成 | <p><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。ルート (/) および <code>/usr</code> ファイルシステムは、システムディスク上に作成しなければならない。</p> | <p>390ページの「SPARC: ファイルシステムを作成する方法」</p> |
| 4. ファイルシステムの復元 | <p>システムディスク上にルート (/) および <code>/usr</code> ファイルシステムを復元する。必要に応じて、二次ディスク上にファイルシステムを復元する。</p> | <p>第 35 章</p> |
| 5. ブートブロックのインストール | <p>システムディスクのみ</p> <p>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。</p> | <p>391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」</p> |

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、シリンダ境界から始め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から始めなければなりません。
- Solaris の fdisk パーティションにディスク全体を使用するか、それより小さくして DOS パーティションに使用する余地を残すことができます。また、既存のパーティションに影響を与えずに、ディスク上に新しい fdisk パーティションを作成できます(それを作成する余地がある場合)。

x86 - Solaris スライス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 (Intel 版 Solaris でのみサポートされます) と、スライスやパーティションと呼ばれる Solaris の fdisk パーティションを区別するようにしています。

▼ x86: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

1. 損傷したシステムディスクをシステムから外します。
2. 追加しようとするディスクにシステム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
3. 交換用のディスクをシステムに正しく接続します。

インストールについての詳細は、ディスクのハードウェアインストールガイドを参照してください。また、ディスクに固有のハードウェア構成要件については、『Solaris 7 デバイスの構成 (Intel 版)』を参照してください。

4. ローカルまたはリモートの **Solaris CD** からブートする場合は、手順 **a** から **e** までを実行します。ネットワークからブートする場合は、手順 **a** をスキップします。
 - a. **Solaris** のインストール **CD** を **CD-ROM** ドライブに挿入します。
 - b. **Solaris** ブートフロッピーディスクを主フロッピーディスクドライブ (**DOS** ドライブ **A**) に挿入します。
 - c. Type any key to continue プロンプトが表示されたら、任意のキーを押してシステムをリブートします。また、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 Multiple Device Boot Subsystem メニューが表示されます。
 - d. **Multiple Device Boot Subsystem** メニューから、ブートデバイスとして **CD-ROM drive** または **net(work)** を選択します。
Secondary Boot Subsystem メニューが表示されます。
 - 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 root プロンプト (#) が表示されます。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。405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: 二次ディスクを接続してブートする方法

1. スーパーユーザーになります。

2. ディスクが **Solaris** ソフトウェアでサポートされない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。
3. システムのブート時に読み込まれる `/reconfigure` ファイルを作成します。

```
# touch /reconfigure
```

`/reconfigure` ファイルを作成すると、電源を入れるとき、または後からシステムをブートするときに、SunOS ソフトウェアは新しくインストールされた周辺デバイスの有無をチェックします。

4. システムをシャットダウンします。

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

`-i0` システムを `init` 状態 0 (電源切断) にする。

`-gn` ログインしているユーザーに、`n` 秒後にシステムのシャットダウンが始まることを通知する。

`-y` コマンドがユーザーの介入なしで実行されるように指定する。

Type any key to continue プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。
6. 追加しようとするディスクにシステム上の他のデバイスとは異なるターゲット番号が設定されているかどうかを確認します。
通常は、ディスクの背面にそのための小型スイッチが付いています。
7. ディスクをシステムに正しく接続します。
インストールについての詳細は、ディスクのハードウェアインストールガイドを参照してください。また、ディスクに固有のハードウェア構成要件については、『*Solaris 7 デバイスの構成 (Intel 版)*』を参照してください。
8. すべての外部周辺デバイスの電源を入れます。
9. システムの電源を入れます。

システムがブートされ、ログインプロンプトが表示されます。

次に進む手順

システムをブートしたら、ディスク上にスライスとディスクラベルを作成できます。405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: Solaris fdisk パーティションを作成する方法

1. 394ページの「x86: fdisk パーティションの作成上のガイドライン」を必ず読んでください。
2. スーパーユーザーになります。
3. `format` ユーティリティを起動します。

```
# format
```

4. 画面に表示されたリストから、**Solaris** fdisk パーティションを作成するディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number

Solaris fdisk パーティションを作成するディスクの番号

5. fdisk メニューに移動します。

```
format> fdisk
```

表示される fdisk メニューは、fdisk パーティションがすでにディスク上に存在しているかどうかにより異なります。次の表を使用して、次に行う手順を決定してください。

| 何をしたいのか | 次の手順 | 参照 |
|--|------|--|
| 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 | 402ページの「x86: 例 — ディスク全体を占有する Solaris fdisk パーティションを作成する」 |
| Solaris fdisk パーティションを作成し、既存の Solaris 以外の fdisk パーティションは変更しない。 | 手順 7 | 403ページの「x86: 例 — 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する」 |
| Solaris fdisk パーティションと、Solaris 以外の fdisk パーティションを追加作成する。 | 手順 7 | 404ページの「x86: 例 — Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する」 |

6. ディスク全体にまたがる **Solaris fdisk** パーティションを作成してそれをアクティブにするには、プロンプトで **y** を入力します。次に、401ページの手順 14 に進みます。

```
The recommended default partitioning for your disk is:

a 100% ``SOLARIS System'' partition.

To select this, please type ``y''. To partition your disk
differently, type ``n'' and the ``fdisk'' program will
let you select other partitions. y
```

7. ディスク全体にまたがる **Solaris fdisk** パーティションを作成しない場合は、プロンプトで **n** を入力します。

```

To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
                          Cylinders
Partition  Status    Type      Start    End    Length  %
=====  =====  =====  =====  ==  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE
FOLLOWING:

1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
Enter Selection:

```

8. 「1、Create a partition」を選択し、fdisk パーティションを作成します。

```

      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
                          Cylinders
Partition  Status    Type      Start    End    Length  %
=====  =====  =====  =====  ==  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE
FOLLOWING:

1.  Create a partition
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手順については、400ページの手順 9 から 401ページの手順 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
```

次に進む手順

ディスク上に Solaris fdisk パーティションを作成し終わったら、ディスク上にスライスを作成できます。405ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

x86: 例 — ディスク全体を占有する Solaris fdisk パーティションを作成する

次の例は、format の fdisk オプションを使用して、ディスク全体にまたがる Solaris fdisk パーティションを作成する方法を説明しています。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
   0. c0t0d0 <DEFAULT cyl 1479 alt 2 hd 7 sec 79>
      /eisa/eha@1000,0/cmdk@0,0
   1. c0t2d0 <SUN1.05 cyl 2692 alt 2 hd 9 sec 85>
      /eisa/eha@1000,0/cmdk@2,0
Specify disk (enter its number): 1
selecting c0t2d0
[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
```

(続く)

続き

```
format> label
Ready to label disk, continue? yes
format> quit
```

x86: 例 — 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する

次の例は、DOS-BIG fdisk パーティションがすでに存在しているディスクに、Solaris fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
Total disk size is 2694 cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type  Start  End  Length  %
=====  =====  =====  =====  =====  =====  =====
1          DOS-BIG  1      538    538    20
SELECT ONE OF THE FOLLOWING:
1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders
Cylinder size is 765 (512 byte) blocks
Cylinders
Partition  Status  Type  Start  End  Length  %
=====  =====  =====  =====  =====  =====  =====
1          DOS-BIG  1      538    538    20
2          Active  SOLARIS  539    2693    2155    80
SELECT ONE OF THE FOLLOWING:
1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
Enter Selection: Selection: 4
```

(続く)

続き

```
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> q
```

x86: 例 — Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する

次の例は、Solaris fdisk パーティションと DOSBIG fdisk パーティションを作成
する方法を説明しています。

```
format> fdisk
The recommended default partitioning for your disk is:
  a 100% "SOLARIS System" partition.
To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
          Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Change Active (Boot from) partition
  3. Delete a partition
  4. Exit (Update disk configuration and exit)
  5. Cancel (Exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
(1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
(5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?8
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 20
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". n
      Total disk size is 2694 cylinders
      Cylinder size is 765 (512 byte) blocks
          Cylinders
Partition  Status  Type      Start  End  Length  %
=====  =====  =====  =====  ===  =====  ==
          1          DOS-BIG          1    538    538    20
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Change Active (Boot from) partition
  3. Delete a partition
```

(続く)

```

    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    539   2693   2155     80

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exit without updating disk configuration)
Enter Selection: 4
format> q

```

▼ x86: ディスクスライスとディスクラベルを作成する方法

1. スーパーユーザーになります。
2. format ユーティリティを起動します。

```
# format
```

3. 画面に表示されるリストから、パーティションを変更したいディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number

パーティションを変更したいディスクの番号

4. `partition` メニューに移動します (スライスを設定できます)。

```
format> partition
```

5. 現在のパーティション (スライス) テーブルを表示します。

```
partition> print
```

6. 変更作業を開始します。

```
partition> modify
```

7. ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]? 1
```

free hog スライスについての詳細は、351ページの「**free hog** スライスの使用方法」を参照してください。

8. 続行するかどうかを尋ねるプロンプトが表示されたら、`yes` と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition  
table based on above table[yes]? yes
```

9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0)、スワップ (スライス 1) (必須)

および

- /usr (スライス 6) (省略可能)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。

10. プロンプトが表示されたら `yes` と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? yes
```

表示されたパーティションテーブルが希望どおりでないために変更したい場合は、`no` と応答して 406 ページの手順 6 に戻ります。

11. パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

partition-name 新しいパーティションテーブルの名前

12. 新しいディスク上でスライスを割り当て終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13. `partition` メニューを終了します。

```
partition> quit
```

14. `verify` コマンドで、新しいディスクラベルを確認します。

```
format> verify
```

15. `format` メニューを終了します。

```
format> quit
```

次に進む手順

ディスクスライスとディスクラベルを作成し終わったら、ディスク上にファイルシステムを作成できます。408ページの「x86: ファイルシステムを作成する方法」を参照してください。

▼ x86: ファイルシステムを作成する方法

1. スーパーユーザーになります。
2. `newfs` コマンドを使用して、スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdisk/cwtxdysz
```

`/dev/rdisk/cwtxdysz`

作成するファイルシステムの raw デバイス

`newfs` コマンドについての詳細は、第 27 章を参照してください。

3. 未使用のマウントポイントにマウントすることによって、新しいファイルシステムを確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls /mnt
lost+found
```

次に進む手順

| 追加するディスク | 参照先 |
|----------|---|
| システムディスク | ディスク上にルート (/) と /usr のファイルシステムを復元する必要がある。第 35 章を参照。 ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールする。409ページの「x86: システムディスクにブートブロックをインストールする方法」を参照。 |
| 二次ディスク | 新しいディスク上にファイルシステムを復元しなければならないことがある。第 35 章を参照。 新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が完了することになる。ファイルシステムをユーザーが利用できるようにする方法については、第 28 章を参照。 |

▼ x86: システムディスクにブートブロックをインストールする方法

1. スーパーユーザーになります。
2. 次のコマンドを入力します。

```
# /usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/cwtxdys2
```

| | |
|---|---------------------|
| /usr/platform/`uname -i`/lib/fs/ufs/pboot | パーティションのブートファイル |
| /usr/platform/`uname -i`/lib/fs/ufs/bootblk | ブートブロックコード |
| /dev/rdisk/cwtxdys2 | ディスク全体を表す raw デバイス名 |

3. システムをリブートし、レベル **3** で実行することによって、ブートブロックがインストールされていることを確認します。

```
# init 6
```

x86: 例 — システムディスクにブートブロックをインストールする

```
# /usr/sbin/installboot /usr/platform/i86pc/lib/fs/ufs/pboot  
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c0t6d0s2
```

format ユーティリティの参照情報

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。

- 411ページの「format ユーティリティを使用するための要件または制限」
- 412ページの「format のメニューとコマンド記述について」
- 419ページの「format に使用されるファイル — format.dat」
- 427ページの「関連するマニュアルページ」
- 425ページの「format コマンドへの入力規則」

format ユーティリティの概要については、第 21 章を参照してください。

format ユーティリティを使用するための要件または制限

format ユーティリティを使用するには、スーパーユーザーにならなければなりません。スーパーユーザーでない場合は、format を使用しようとするときのエラーメッセージが表示されます。

```
% format Searching for disk...done
No permission (or no disk found)!
```

format を使用する場合に情報を保存するための推奨事項

- 他の作業を実行する前に、ディスクドライブ上のすべてのファイルのバックアップをとります。
- format の dump コマンドを使用して、欠陥のリストをファイルに保存します。ファイル名には、ドライブタイプ、モデル番号、シリアル番号を含めておくべきです。
- メーカーから出荷時にドライブといっしょに提供された、欠陥のリストを保管します。

format のメニューとコマンド記述について

format のメインメニューは次のようになっています。

```
FORMAT MENU:
  disk      - select a disk
  type      - select (define) a disk type
  partition - select (define) a partition table
  current   - describe the current disk
  format    - format and analyze the disk
  repair    - repair a defective sector
  label     - write label to the disk
  analyze   - surface analysis
  defect    - defect list management
  backup    - search for backup labels
  verify    - read and display labels
  save      - save new disk/partition definitions
  inquiry   - show vendor, product and revision
  volname   - set 8-character volume name
  quit
format>
```

表 25-1 に、format のメインメニュー項目を示します。

表 25-1 format のメインメニュー項目の説明

| 項目 | コマンド/メニュー | 機能 |
|-----------|-----------|---|
| disk | コマンド | 後の操作に使用するディスクを選択する (現在のディスクと呼ぶ)。システムのすべてのドライブが表示される。 |
| type | コマンド | 現在のディスクのメーカーとモデルを選択する。認識されているドライブタイプのリストが表示される。SCSI-2 対応ディスクドライブの場合は Auto configure オプションを選択する。 |
| partition | メニュー | スライスを作成したり変更したりする。詳細は、414 ページの「partition メニュー」を参照。 |
| current | コマンド | 現在のディスクに関して次の情報が表示される。 <ul style="list-style-type: none"> ■ デバイス名とタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ 物理デバイス名 |
| format | コマンド | 次のいずれかの情報源をこの順番に使用して、現在のディスクをフォーマットする。 <ol style="list-style-type: none"> 1. format.dat ファイル内の情報 2. 自動構成プロセスからの情報 3. format.dat エントリが見つからない場合にプロンプトに回答した情報 |
| fdisk | メニュー | fdisk プログラムが実行され、Solaris fdisk パーティションを作成する。 |
| repair | コマンド | ディスク上で特定のブロックを修復する。 |
| label | コマンド | 現在のディスクに新しいラベルを書き込む。 |
| analyze | メニュー | 読み取り、書き込み、比較テストを実行する。詳細は、416 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、418 ページの「defect メニュー」を参照。 |
| backup | コマンド | バックアップラベルを検索する。 |

表 25-1 format のメインメニュー項目の説明 続く

| 項目 | コマンド/メニュー | 機能 |
|---------|-----------|--|
| verify | コマンド | ディスクに関して次の情報を出力する。 <ul style="list-style-type: none"> ■ デバイス名とタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ パーティションテーブル |
| save | コマンド | 新しいディスクとパーティションの情報を保存する。 |
| inquiry | コマンド | 現在のドライブのベンダ、製品名、リビジョンレベルが出力される (SCSI ディスクのみ)。 |
| volname | コマンド | 新しい 8 文字のボリューム名を使用してディスクラベルを作成する。 |
| quit | コマンド | format メニューを終了する。 |

partition メニュー

partition メニューは次のようになっています。

```
format> partition
PARTITION MENU:
  0 - change '0' partition
  1 - change '1' partition
  2 - change '2' partition
  3 - change '3' partition
  4 - change '4' partition
  5 - change '5' partition
  6 - change '6' partition
  7 - change '7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit
partition>
```

表 25-2 に、partition メニューの項目を示します。

表 25-2 partition メニューの項目の説明

| コマンド | 機能 |
|----------------------|---|
| change 'x' partition | 以下の値を設定し、新しいスライスを指定する。 <ul style="list-style-type: none"> ■ 識別タグ ■ アクセス権フラグ ■ 開始シリンダ ■ サイズ |
| select | あらかじめ定義されたスライステーブルを選択する。 |
| modify | スライステーブル内のすべてのスライスを変更する。個々のスライスに対して change 'x' partition コマンドを実行するよりも、このコマンドが使用されることが多い。 |
| name | 現在のスライステーブルの名前を指定する。 |
| print | 現在のスライステーブルが表示される。 |
| label | スライスマップとラベルを現在のディスクに書き込む。 |
| quit | partition メニューを終了する |

x86: fdisk メニュー

x86 システム上でのみ、次のような fdisk メニューが表示されます。

```
format> fdisk
          Total disk size is 1855 cylinders
          Cylinder size is 553 (512 byte) blocks
                Cylinders
Partition  Status  Type      Start  End  Length  %
-----  -
1          Active  DOS-BIG   0     370   371     20
2          Active  SOLARIS  370  1851  1482    80

SELECT ONE OF THE FOLLOWING:

1.  Create a partition
2.  Change Active (Boot from) partition
3.  Delete a partition
4.  Exit (Update disk configuration and exit)
5.  Cancel (Exit without updating disk configuration)
```

(続く)

続き

```
Enter Selection:
```

表 25-3 に、fdisk メニューの項目を示します。

表 25-3 fdisk メニューの項目の説明

| コマンド | 機能 |
|-------------------------|---|
| Create a partition | fdisk パーティションを作成する。Solaris や DOS など、オペレーティングシステムごとに別々のパーティションを作成しなければならない。1 台のディスクの最大パーティション数は 4 である。fdisk のパーティションのサイズをパーセンテージで入力するように促すプロンプトが表示される。 |
| Change Active partition | どのパーティションをブートに使用するかを指定する。これにより、第 1 段階のブートプログラムが実行する第 2 段階のブートプログラムの存在するパーティションを指定する。 |
| Delete a partition | 以前に作成したパーティションを削除する。このコマンドを実行すると、パーティション内のすべてのデータが失われる。 |
| Exit | 新しいパーティションテーブルを書き込んで fdisk メニューを終了する。 |
| Cancel | パーティションテーブルを変更せずに fdisk メニューを終了する。 |

analyze メニュー

analyze メニューは次のようになっています。

```
format> analyze
ANALYZE MENU:
  read   - read only test   (doesn't harm SunOS)
  refresh - read then write  (doesn't harm data)
  test   - pattern testing  (doesn't harm data)
  write  - write then read   (corrupts data)
```

(続く)

```

compare - write, read, compare (corrupts data)
purge   - write, read, write   (corrupts data)
verify  - write entire disk, then verify (corrupts data)
print   - display data buffer
setup   - set analysis parameters
config  - show analysis parameters
quit
analyze>

```

表 25-4 に、analyze メニューの項目を示します。

表 25-4 analyze メニューの項目の説明

| コマンド | 機能 |
|---------|--|
| read | このディスクの各セクターを読み込む。デフォルトで欠陥ブロックを修復する。 |
| refresh | データを損なわずにディスク上で読み込んで書き込む。デフォルトで欠陥ブロックを修復する。 |
| test | データを損なわずに一連のパターンをディスクに書き込む。デフォルトで欠陥ブロックを修復する。 |
| write | 一連のパターンをディスクに書き込んで、そのデータをディスクから読み込む。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。 |
| compare | ディスクに一連のパターンを書き込み、そのデータを読み込み、書き込みバッファ内のデータと比較する。ディスク上の既存のデータは破壊される。デフォルトで欠陥ブロックを修復する。 |
| purge | ディスク上のデータを消去する。いかなる手段でも取り出せないように、ディスクに複数のパターンを書き込むことによって、すべてのデータを削除する。ディスク全体 (またはディスクのセクション) に 3 種類のパターンが書き込まれ、検査に合格すると 16 進のビットパターンが書き込まれて、データが削除される。 デフォルトで欠陥ブロックを修復する。 |
| verify | 1 度目にディスク全体の各ブロックに固有のデータを書き込み、2 度目にそのデータを検査する。ディスク上の既存のデータは失われる。デフォルトで欠陥ブロックを修復する。 |

表 25-4 analyze メニューの項目の説明 続く

| コマンド | 機能 |
|--------|--|
| print | 読み込み/書き込みバッファ内のデータを表示する。 |
| setup | 次の解析パラメータを指定する。 Analyze entire disk? yes Starting block number: ドライブによって異なる Ending block number: ドライブによって異なる Loop continuously? no Number of passes: 2 Repair defective blocks? yes Stop after first error? no Use random bit patterns? no Number of blocks per transfer: 126 (0/n/m) Verify media after formatting? yes Enable extended messages? no Restore defect list? yes Restore disk label? yes 太字はデフォルトを示す。 |
| config | 現在の解析パラメータを表示する。 |
| quit | analyze メニューを終了する。 |

defect メニュー

defect メニューは次のようになっています。

```
format> defect
DEFECTION MENU:
  primary - extract manufacturer's defect list
```

(続く)

```

grown      - extract manufacturer's and repaired defects lists
both       - extract both primary and grown defects lists
print      - display working list
dump       - dump working list to file
quit
defect>

```

表 25-5 に defect メニューの項目を示します。

表 25-5 defect メニューの項目の説明

| コマンド | 機能 |
|---------|---|
| primary | メーカーの欠陥リストをディスクドライブから読み込んで、メモリー内の欠陥リストを更新する。 |
| grown | 増分の欠陥リスト (analyze メニューから検出された欠陥) を読み込んで、メモリー内の欠陥リストを更新する。 |
| both | メーカーの欠陥リストと増分の欠陥リストを読み込んで、メモリー内の欠陥リストを更新する。 |
| print | メモリー内の欠陥リストを表示する。 |
| dump | メモリー内の欠陥リストをファイルに保存する。 |
| quit | defect メニューを終了する。 |

format に使用されるファイル — format.dat

format で使用されるデータ、/etc/format.dat の内容は次のとおりです。

- ディスクタイプ
- デフォルトのスライステーブル

Solaris オペレーティングシステムといっしょに出荷される format.dat ファイルでは、多数の標準的なディスクがサポートされます。使用中のディスクドライブが

format.dat ファイルに含まれていない場合は、そのエントリを追加するか、format で処理を実行中に必要な情報の入力を促すプロンプトを表示させることができます。

ディスクドライブをサイト全体で使用する場合は、format.dat ファイルにエントリを追加すると時間を節約できます。format.dat ファイルを他のシステム上で使用する場合は、format.dat ファイルに追加する特定のディスクドライブを使用するシステムごとに、このファイルをコピーしてください。

次の場合には、システムのデータファイルを変更する必要があります。

- ディスクが Solaris オペレーティング環境でサポートされない場合
- スライステーブルが入っているディスクが、Solaris オペレーティング環境のデフォルト構成とは異なる場合

注 - デフォルトエントリは変更しないでください。デフォルトエントリを変更したい場合は、混乱を避けるために、そのエントリをコピーし、別の名前を付けて変更します。

構造

format.dat データファイルには、format ユーティリティに使用されるディスクドライブ情報が入っています。format.dat ファイル内では、次の3つの項目が定義されています。

- 検索パス
- ディスクタイプ
- スライステーブル

構文

データファイルには、次の構文規則が適用されます。

- シャープ記号 (#) はコメント文字です。シャープ記号に続く1行のテキストは、format で解釈されません。
- format.dat ファイル内の各定義は、1つの論理行で評価されます。定義が長すぎて1行に収まらない場合は、定義の最終行を除くすべての行末にバックスラッシュ (\) 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。
- 空白は、format に無視されます。代入値に空白を含めたい場合は、値全体を二重引用符 (") で囲みます。これにより、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。値はカンマ (,) で区切ります。

キーワード

データファイルには、起動時に format に読み込まれるディスク定義が入っています。各定義は、キーワード `search_path`、`disk_type`、または `partition` で始まります。表 25-6 を参照してください。

表 25-6 format.dat のキーワードの説明

| キーワード | 用途 |
|--------------------------|--|
| <code>search_path</code> | このキーワードは format.dat ファイルでは使用しない。Solaris 2.0 リリース以降は、format ユーティリティは論理デバイス階層 (/dev) を検索するので、このキーワードを設定してシステムのディスクを検索する必要がなくなった。 |
| <code>disk_type</code> | コントローラとディスクのモデルを定義する。各 <code>disk_type</code> 定義には、ディスクの物理ジオメトリに関する情報が入っている。デフォルトのデータファイルには、Solaris オペレーティング環境でサポートされるコントローラとディスクの定義が入っている。サポートされないディスクを使用する場合にのみ、新しい <code>disk_type</code> を追加する必要がある。必要に応じて、 <code>disk_type</code> 定義をデータファイルにいくつ追加してもかまわない。 |
| <code>partition</code> | ディスクタイプのスライステーブルを定義する。スライステーブルには、スライス情報だけでなく、format 内で参照できる名前が入っている。デフォルトのデータファイルには、数種類のディスクドライブに対応するデフォルトのスライス定義が入っている。システムのディスク上にスライスを作成し直した場合は、スライス定義を追加する。必要に応じて、スライス情報をデータファイルにいくつ追加してもかまわない。 |

ディスクタイプ

`disk_type` は、コントローラとディスクのモデルを定義します。各 `disk_type` 定義には、ディスクの物理ジオメトリに関する情報が入っています。デフォルトのデータファイルには、Solaris オペレーティング環境でサポートされるコントローラとディスクの定義が入っています。サポートされないディスクを使用する場合に限り、新しい `disk_type` を追加する必要があります。必要に応じて、`disk_type` 定義をデータファイルにいくつ追加してもかまいません。

キーワード自体が、ディスクタイプ名になります。この名前は、ディスクのラベルの一部になり、`format` の実行時にディスクタイプを識別するために使用されます。空白が含まれている名前は、二重引用符で囲んでください。表 25-7 に、すべての `disk_type` 定義でキーワードの他に割り当てなければならない識別子を示します。

表 25-7 必須の `disk_type` 識別子

| 識別子 | 説明 |
|--------------------|--|
| <code>ctrlr</code> | ディスクタイプに有効なコントローラのタイプ。現在、この代入式に有効な値は SCSI と ISP-80 (IPI コントローラ) である。 |
| <code>ncyl</code> | 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。 |
| <code>acyl</code> | ディスクタイプ内の代替シリンダ数。 <code>format</code> は、これらのシリンダを使用して、ドライブの欠陥リストなどの情報を格納する。代替シリンダとして、常に 2 つはシリンダを残しておく必要がある。 |
| <code>pcyl</code> | ディスクタイプ内の物理シリンダ数。この数は、ディスク媒体の境界を計算するために使用される。通常、この数値は <code>ncyl</code> と <code>acyl</code> の合計に等しくなるが、そうでない場合もある。 |
| <code>nhead</code> | ディスクタイプ内のヘッド数。この数値は、ディスク媒体の境界を計算するために使用される。 |
| <code>nsect</code> | ディスクタイプ内の 1 トラック当たりのデータセクター数。この数値は、ディスク媒体の境界を計算するために使用される。これはデータセクターだけで、スペアは割り当てには含まれない。 |
| <code>rpm</code> | ディスクタイプの 1 分当たりの回転数。この情報はラベルに書き込まれ、後からファイルシステムでファイルデータの最適位置の計算に使用される。 |

コントローラによっては、他の代入式が必要な場合があります。表 25-8に、SCSI コントローラに必要な代入式を示します。

表 25-8 SCSI コントローラの disk_type 識別子

| 識別子 | 説明 |
|-----------|---|
| fmt_time | 所定のドライブのフォーマットに要する時間を示す数値。詳細は、コントローラのマニュアルを参照。 |
| cache | format の処理中にオンボードキャッシュの動作を制御する数値。詳細は、コントローラのマニュアルを参照。 |
| trks_zone | 代替セクターのマッピング内で使用される 1 つの欠陥領域当たりのトラック数を指定した数値。詳細は、コントローラのマニュアルを参照。 |
| asect | このパラメータに代入した数値は、所定の欠陥領域内で代替マッピングに利用できるセクター数を指定する。詳細は、コントローラのマニュアルを参照。 |

次に、disk_type 定義の例を示します。

```
disk_type = "SUN0535" \  
: ctlr = SCSI : fmt_time = 4 \  
: ncy1 = 1866 : acyl = 2 : pcyl = 2500 : nhead = 7 : nsect = 80 \  
: rpm = 5400  
disk_type = "SUN0669" \  
: ctlr = SCSI : fmt_time = 4 \  
: trks_zone = 15 : asect = 5 : atrks = 30 \  
: ncy1 = 1614 : acyl = 2 : pcyl = 1632 : nhead = 15 : nsect = 54 \  
: rpm = 3600 : bpt = 31410  
disk_type = "SUN1.0G" \  
: ctlr = SCSI : fmt_time = 4 \  
: trks_zone = 15 : asect = 5 : atrks = 30 \  
: ncy1 = 1703 : acyl = 2 : pcyl = 1931 : nhead = 15 : nsect = 80 \  
: rpm = 3597 : bpt = 41301
```

スライステーブル

partition 定義のキーワードが、スライステーブル名になります。空白が含まれている名前は、二重引用符で囲んでください。表 25-9 に、すべてのスライステーブル内で値を代入しなければならない識別子を示します。

表 25-9 スライステーブルの必須識別子

| 識別子 | 説明 |
|------|---|
| disk | このスライステーブルが定義されている <code>disk_type</code> の名前。この名前は <code>disk_type</code> 内で使用されるとおりに指定しなければならない。 |
| ctlr | このスライステーブルを接続できるコントローラタイプディスク。現在、この代入式に有効な値は IPI コントローラを表す <code>ISP-80</code> と SCSI コントローラを表す <code>SCSI</code> である。ここで指定したコントローラタイプは、上記で選択した <code>disk_type</code> にも定義しなければならない。 |

スライス定義内の他の代入式では、実際のスライス情報を記述します。識別子は 0 から 7 までの番号です。これらの代入式は省略可能です。明示的に代入されていないスライスは、長さ 0 に設定されます。最初の数値はスライスの開始シリンダで、第 2 はスライス内のセクター数です。次に、スライス定義の例を示します。

```
partition = "SUN0535" \
: disk = "SUN0535" : ctlr = SCSI \
: 0 = 0, 64400 : 1 = 115, 103600 : 2 = 0, 1044960 : 6 = 300, 876960
partition = "SUN0669" \
: disk = "SUN0669" : ctlr = SCSI \
: 0 = 0, 32400 : 1 = 40, 64800 : 2 = 0, 1307340 : 6 = 120, 1210140
partition = "SUN1.0G" \
: disk = "SUN1.0G" : ctlr = SCSI \
: 0 = 0, 32400 : 1 = 27, 64800 : 2 = 0, 2043600 : 6 = 81, 1946400
```

位置

`format` ユーティリティは、次の方法でデータファイルの位置を認識します。

1. `-x` コマンド行オプションでパス名を指定した場合は、そのファイルは常にデータファイルとして使用されます。
2. `-x` オプションを指定しなければ、`format` は現在のディレクトリ内でファイル `format.dat` を検索します。このファイルが見つかると、データファイルとして使用されます。
3. どちらの方法でもデータファイルが見つからなければ、`format` はデータファイルとして `/etc/format.dat` を使用します。このファイルは Solaris オペレーティング環境といっしょに出荷されるので、必ずあるはずで

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、427ページの「ヘルプ」を参照してください。

数値

format 内では、整数を入力しなければならない場所が数カ所あります。データを指定するか、選択肢のリストから選択しなければなりません。どちらの場合も、help 機能を使用すると、format は期待する整数の上限と下限を表示し、目的の数値を入力するだけですみます。数値は、その一部として底を明示的に指定しない限り (16 進数を表す 0x など)、10 進数と見なされます。

次の例は、整数の入力を示しています。

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

ブロック番号

ディスクのブロック番号を入力しなければならない場合は、情報を次の 2 つの方法で入力できます。

- ブロック番号を整数として入力する。
- ブロック番号をシリンダ/ヘッド/セクター書式で入力する。

この情報は、論理ブロック番号を表す整数として指定できます。任意の底の整数を指定できますが、デフォルトは 10 進です。また、ここで最大演算子 (ドル記号 \$) を使用して、format に適切な値を選択させることもできます。論理ブロックの形式は、SunOS のディスクドライバによってエラーメッセージに使用されます。

ブロック番号を指定するには、シリンダ/ヘッド/セクター書式を使用する方法もあります。この形式では、ブロック番号の 3 つの論理構成要素である、シリンダ、ヘッド、セクターの値を明示的に指定しなければなりません。これらの値は論理値ですが、媒体のレイアウトに関連するディスク領域の定義に使用できます。

シリンダ/ヘッド/セクター番号を指定しなければ、該当する値は 0 であると見なされます。また、番号の代わりに最大演算子を使用して、format に適切な値を選択させることもできます。次に、シリンダ、ヘッド、セクターエントリの例を示します。

```
Enter defective block number: 34/2/3
Enter defective block number: 23/1/
Enter defective block number: 457//
Enter defective block number: 12345
Enter defective block number: 0xabcd
Enter defective block number: 334/$/2
Enter defective block number: 892//$
```

format は、ブロック番号を常に上記の両方の書式で出力します。また、help 機能によって、期待されるブロック番号の上限と下限が両方の書式で表示されます。

コマンド名

format でメニュープロンプトが表示される場合は、コマンド名を入力する必要があります。コマンド名は、目的のコマンドとして区別できる長さまで省略できます。

たとえば、p(partition) を使用して format メニューから partition メニューにアクセスできます。次に、p(rint) を使用して現在のスライステーブルを表示できます。

```
format> p
PARTITION MENU:
  0 - change '0' partition
  1 - change '1' partition
  2 - change '2' partition
  3 - change '3' partition
  4 - change '4' partition
  5 - change '5' partition
  6 - change '6' partition
  7 - change '7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit
partition> p
```

その他の名前

format では、名前を指定しなければならない場合があります。このような場合は、名前に使用したい文字列を自由に指定できます。空白を含む名前は、二重引用符 (") で囲まなければならない。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。

ヘルプ

`format` ユーティリティにはヘルプ機能が組み込まれているので、入力が必要なときに使用できます。疑問符 (?) を入力するだけで必要な情報に関するヘルプが表示され、どんなタイプの入力が必要かについて簡潔な説明が表示されます。

メニュープロンプトから ? と入力すると、利用できるコマンドのリストが表示されます。

関連するマニュアルページ

`format` ユーティリティに関連するマニュアルページは、`format(1M)` と `format.dat(4)` です。`format(1M)` には、`format` ユーティリティの基本機能とコマンド行で使用できるすべてのオプションについて説明されています。`format.dat(4)` には、`format` ユーティリティで使用するディスクドライブ構成情報について説明されています。

パート VIII ファイルシステムの管理

このパートでは、Solaris オペレーティング環境でファイルシステムを管理する手順について説明します。次の章で構成されています。

| | |
|--------|---|
| 第 26 章 | ファイルシステムの種類、一般に使用する管理コマンド、ファイルシステムをマウントし、マウント解除する基本的な作業など、ファイルシステム全般について説明します。 |
| 第 27 章 | UFS ファイルシステムの作成、一時ファイルシステム (TMPFS) の作成と管理、ループバックファイルシステム (LOFS) の作成を行う手順について説明します。 |
| 第 28 章 | どのファイルシステムがマウントされているかを判断する手順、 <code>/etc/vfstab</code> に列挙されているファイルをマウントする方法、UFS、NFS、PCFS (DOS) ファイルシステムをマウントする方法について説明します。 |
| 第 29 章 | キャッシュファイルシステム (CacheFS™) の概要と使用手順について説明します。 |
| 第 30 章 | スワップ空間を追加する場合の構成、スワップ資源の監視、スワップファイルの作成と使用可能にする方法、余分なスワップ空間を削除する手順などについて説明します。 |

第 31 章

ファイルシステムの状態を記録する方法、`fsck` プログラムでチェックされる内容、自動ブートチェック機能を変更する方法、`fsck` プログラムの使用方法について説明します。

第 32 章

ルート(/)と `/usr` のファイルシステムのデフォルトディレクトリ、`/kernel` ディレクトリにあり、`mkfs` コマンドと `newfs` コマンドに固有のディレクトリなど、ファイルシステムの参照情報について説明します。

ファイルシステムの概要

この章の内容は次のとおりです

- 431ページの「ファイルシステムにおける新機能」
- 433ページの「ファイルシステムのタイプ」
- 437ページの「ファイルシステム管理コマンド」
- 439ページの「デフォルトの Solaris ファイルシステム」
- 440ページの「スワップ空間」
- 441ページの「UFS ファイルシステム」
- 443ページの「ファイルシステムのマウントとマウント解除」
- 450ページの「ファイルシステムのタイプを調べる」

ファイルシステムにおける新機能

Solaris 7では、ファイルシステムの2つの新機能が提供されます。UFS ロギングと、ファイルのアクセス時間更新を無視する新しいマウントオプションです。

UFS ロギングとは、トランザクション (UFS に対する変更操作のすべて) が UFS ファイルシステムに適用される前に、ログに格納するプロセスのことです。一度格納したトランザクションは、その後ファイルシステムに適用できます。

UFS ロギングには2つの利点があります。まず UFS ロギングは、ファイルシステムの整合性を保持するため、`fsck(1M)` を実行する必要がなくなります。また、`fsck`

を省略できるために、システムがクラッシュした場合、あるいは、不完全に停止後、システムリブートの時間が短くなります。

デフォルトでは、UFS ロギングは無効です。UFS ロギングを有効にするには、ファイルシステムをマウントするときに、`-o logging` オプションを `mount` コマンドに指定しなければなりません。また、`fsdb` コマンドも更新され、UFS ロギング用の新しいデバッグコマンドを持つようになりました。

ファイルのアクセス時間の更新を無視するには、UFS ファイルシステムをマウントするときに、`-o noatime` オプションを指定します。このオプションによって、アクセス時間が重要でないファイルシステム (たとえば、Usenet ニュースプール) でのディスクに対する動作が減ります。詳細は、`mount_ufs(1M)` のマニュアルページを参照してください。

概要

ファイルシステムは、ファイルを編成して格納するためのディレクトリ構造です。「ファイルシステム」という用語には、さまざまな使用方法があります。

- ディスクベース、ネットワークベース、または仮想ファイルシステムなど、特定のタイプのファイルシステムを指す場合
- ルートディレクトリから下位へのファイルツリー全体を指す場合
- ディスクスライスや他の媒体記憶デバイスのデータ構造を指す場合
- ファイル構造のうち、アクセスできるようにメインファイルツリー上のマウントポイントに接続されている部分を指す場合

通常、その意味は状況に応じて判断できます。

Solaris オペレーティング環境は、各種ファイルシステムへの標準インタフェースを提供する「仮想ファイルシステム」(VFS) アーキテクチャを使用します。VFS アーキテクチャによって、ユーザーやプログラムが基礎となるファイルシステムのタイプを知らなくても、カーネルはファイルの読み取り、書き込み、一覧表示などの基本操作を処理できます。

ファイルシステムの管理は、最も重要なシステム管理作業の 1 つです。この章を読んで概念と計画方法について理解してください。次の作業については、Part VIII の他の章と他の Part を参照してください。

- 新しい UFS と仮想ファイルシステムを設定する - 詳細は、第 27 章と第 29 章を参照してください。

- 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 – 詳細は第 28 章を参照してください。
- 新しい記憶デバイスを接続して構成する – 詳細は 第 21 章を参照してください。
- 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 – バックアップの作成、およびファイルとファイルシステムの復元については、第 33 章を参照してください。
- ファイルシステムの損傷をチェックして訂正する – 自動 (ブート時) チェックに失敗した場合の処理については、第 31 章を参照してください。

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
- ネットワークベース
- 仮想

ファイルシステムのタイプを確認するには、450ページの「ファイルシステムのタイプを調べる」を参照してください。

ディスクベースのファイルシステム

ディスクベースのファイルシステムは、ハードディスク、CD-ROM、フロッピーディスクなどの物理媒体に格納されます。ディスクベースのファイルシステムは、さまざまな形式で作成できます。利用できる形式は次のとおりです。

- **UFS – UNIX** ファイルシステム (4.3 Tahoe リリースに組み込まれていた BSD Fast File システム)。UFS は、Solaris オペレーティング環境のデフォルトのディスクベースファイルシステムです。

UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクスライスとは物理的なディスクのサブセットのことで、連続するブロックからなる1つの範囲のことで、スライスはスワップ空間などの raw デバイスとして使用することも、ディスクベースのファイルシステムとして使用することもできます。ディスクのフォー

マットとディスクのスライスへの分割についての詳細は、第21章を参照してください。

- **HSFS** – High Sierra と ISO 9660 のファイルシステム。High Sierra は、初めての CD-ROM ファイルシステムです。ISO 9660 は、High Sierra ファイルシステムの公式の標準バージョンです。HSFS ファイルシステムは CD-ROM 上で使用される読み取り専用ファイルシステムです。Solaris HSFS では、ISO 9660 への Rock Ridge 拡張がサポートされるので、CD-ROM 上でも、すべての UFS ファイルシステムのセマンティクスとファイルタイプを提供します。ただし、書き込み可能ではなく、ハードリンクも提供しません。
- **PCFS** – PC ファイルシステム。DOS ベースのパーソナルコンピュータ用に作成された DOS フォーマットのディスク上のデータとプログラムに読み取り/書き込みのアクセスができます。

ディスクベースの各種ファイルシステムは、次のように特定の媒体のタイプに対応しています。

- UFS とハードディスク
- HSFS と CD-ROM
- PCFS とフロッピーディスク

ただし、上記以外の対応も可能です。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適用できます。

ネットワークベースのファイルシステム

ネットワークベースのファイルシステムは、ネットワーク上でアクセスされるファイルシステムです。一般に、ネットワークベースのファイルシステムは1つのシステム上(通常はサーバー上)にあり、他のシステムからネットワーク経由でアクセスされます。NFS は、ネットワークベースで利用できる唯一のファイルシステムです。

NFS は、Solaris の分散ファイルシステムです。NFS で分散資源(ファイルやディレクトリ)を管理するには、サーバーから分散資源をエクスポートして個々のクライアントシステムでそれらをマウントします。詳細は、447ページの「ネットワークファイルシステム(NFS)」を参照してください。

仮想ファイルシステム

仮想ファイルシステムは、特殊なカーネル情報と機能へのアクセスを提供するメモリーベースのファイルシステムです。ほとんどの仮想ファイルシステムは、ディスク領域を使用しません。ただし、キャッシュファイルシステム (CacheFS) は、ディスク上のファイルシステムを使用してキャッシュを保持します。また、一時ファイルシステム (TMPFS) などの一部の仮想ファイルシステムは、ディスク上のスワップ空間を使用します。

キャッシュファイルシステム

キャッシュファイルシステム (CacheFS) を使用すると、リモートファイルシステムや、CD-ROM ドライブのような低速デバイスの性能を改善できます。ファイルシステムをキャッシュすると、リモートファイルシステムや CD-ROM から読み込まれたデータは、ローカルシステム上のキャッシュに格納されます。CacheFS ファイルシステムの設定と管理については、第 29 章を参照してください。

一時ファイルシステム

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べてアクセス速度が高速です。TMPFS 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を軽減でき、システムパフォーマンスを改善できます。たとえば、プログラムをコンパイルすると一時ファイルが作成されます。オペレーティングシステムは、これらのファイルを処理する間に大量のディスク処理やネットワーク処理を行います。TMPFS を使用してこれらの一時ファイルを格納すると、その作成、処理、または削除が大幅に高速になります。

ファイルシステムのマウントが解除されるときと、システムがシャットダウンまたはリブートされるときに、一時ファイルシステム上のファイルは削除されます。

TMPFS は、Solaris オペレーティング環境内の /tmp ディレクトリのデフォルトのファイルシステムです。UFS /tmp ファイルシステムの場合と同様に、/tmp ディレクトリとの間でファイルをコピーまたは移動できます。

TMPFS ファイルシステムは、一時的な退避場所としてスワップ空間を使用します。TMPFS ファイルシステムがマウントされたシステムのスワップ空間が足りないと、次の 2 つの問題が発生する可能性があります。

- TMPFS ファイルシステムは、通常のファイルシステムがいっぱいになった場合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)、一部のプログラムはスワップ空間不足のために実行できなくなる。

TMPFS ファイルシステムの作成方法については、第 27 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 30 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、/ 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、/ 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 27 章を参照してください。

プロセスファイルシステム

プロセスファイルシステム (PROCFS) はメモリー内にあります。PROCFS の /proc ディレクトリには、有効なプロセスのプロセス番号別リストが入っています。/proc ディレクトリ内の内容は、ps などのコマンドに使用されます。デバッガや他の開発ツールも、ファイルシステムコールを使用して、プロセスのアドレス空間にアクセスできます。



注意 - /proc ディレクトリ内のファイルは削除しないでください。/proc ディレクトリからプロセスを削除するのは、最善の方法ではありません。/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ないので注意してください。

/proc ディレクトリには、システム管理は不要です。

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。

- **FIFOFS** (先入れ先出し) – プロセスにデータへの共通アクセス権を与える名前付きパイプのファイル
- **FDFS** (ファイル記述子) – 開いているファイルに、記述子を使用して名前を明示的に与える
- **NAMEFS** – ほとんどの場合、ファイル記述子をファイルの先頭に動的にマウントするために STREAMS に使用される
- **SPECFS** (特殊) – キャラクタ型特殊デバイスとブロック型特殊デバイスへのアクセスを提供する
- **SWAPFS** – カーネルがスワッピングに使用するファイルシステム

ファイルシステム管理コマンド

ほとんどのファイルシステム管理コマンドには、汎用コマンドとファイルシステム専用の2種類があります。可能な場合には、常に汎用コマンドを使用してください。汎用コマンドは、ファイルシステム固有のコマンドを呼び出します。表 26-1 に、汎用ファイルシステム管理コマンドを示します。これらのコマンドは、`/usr/sbin` ディレクトリに入っています。

表 26-1 汎用ファイルシステム管理コマンド

| コマンド | 機能 |
|---------------------------|---|
| <code>clri</code> (1M) | i ノードをクリアする。 |
| <code>df</code> (1M) | 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> (1M) | 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> (1M) | ファイルシステムの完全性をチェックし、検出された損傷を修復する。 |
| <code>fsdb</code> (1M) | ファイルシステムをデバッグする。 |
| <code>fstyp</code> (1M) | ファイルシステムのタイプを調べる。 |
| <code>labelit</code> (1M) | テープにコピーするときに、ファイルシステムのラベルを表示または作成する (<code>volcopy</code> コマンド専用)。 |

表 26-1 汎用ファイルシステム管理コマンド 続く

| コマンド | 機能 |
|----------------|---|
| mkfs (1M) | 新しいファイルシステムを作成する。 |
| mount (1M) | ローカルおよびリモートのファイルシステムをマウントする。 |
| mountall (1M) | 仮想ファイルシステムテーブルに指定されているすべてのファイルシステムをマウントする。(/etc/vfstab) |
| ncheck (1M) | パス名とその i 番号のリストを生成する。 |
| umount (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 (1M) | 仮想ファイルシステムテーブルに指定されているすべてのファイルシステムをマウント解除する。(/etc/vfstab) |
| volcopy (1M) | ファイルシステムのイメージコピーを作成する。 |

ファイルシステムコマンドによるファイルシステムタイプの判断

汎用ファイルシステムコマンドは、次の順序でファイルシステムのタイプを判断します。

1. -F オプションで指定されているファイルシステムのタイプ
2. 特殊デバイスを /etc/vfstab 内のエントリと突き合わせて判断します (special が指定されている場合)。たとえば fsck は、まず fsck device フィールドと突き合わせて一致するエントリを検索します。一致するエントリが見つからなければ、特殊デバイスフィールドと突き合わせてチェックします。
3. ローカルファイルシステムの場合は /etc/default/fs 内で指定されたデフォルトを使用し、リモートファイルシステムの場合は /etc/dfs/fstypes 内で指定されたデフォルトを使用して判断します。

汎用コマンドと専用コマンドのマニュアルページ

汎用コマンドと専用コマンドについては、『*man Pages(1M): System Administration Commands*』を参照してください。専用コマンドのマニュアルページは、汎用コマンドのマニュアルページに続いて入っています。特定のマニュアルページを見つける

には、汎用コマンド名の末尾にアンダースコアとファイルシステムタイプの略称を追加してください。たとえば、UFS ファイルシステムのマウントに関して専用コマンドのマニュアルページを調べるには、`man mount_ufs(1M)` と入力します。

デフォルトの Solaris ファイルシステム

Solaris ファイルシステムは階層構造になっており、ルートディレクトリ (/) から始まり、下位に多数のディレクトリが形成されています。Solaris のインストールプロセスは、デフォルトのディレクトリセットをインストールし、一連の規則を適用して類似するタイプのファイルをグループ化します。表 26-2 に、デフォルトの Solaris ファイルシステムの概要と、各ファイルシステムのタイプを示します。

システムを動作させるには、ルート (/) と /usr のファイルシステムが必要です。/usr ファイルシステムに置かれている最も基本的なコマンドの一部 (mount など) は、システムのブート時や、システムがシングルユーザーモードで実行しており、/usr ファイルシステムがマウントされていない場合でも使用できるように、ルート (/) ファイルシステムにも置かれています。デフォルトディレクトリのリストについては、第 32 章を参照してください。

表 26-2 デフォルトの Solaris ファイルシステム

| ファイルシステムまたはディレクトリ | ファイルシステムのタイプ | 説明 |
|-------------------|--------------|--|
| ルート (/) | UFS | 階層ファイルツリーの最上位。ルートディレクトリには、カーネル、デバイスドライバ、システムのブートに使用されるプログラムなど、システム処理に欠かせないディレクトリとファイルが入っている。また、ローカルとリモートのファイルシステムをファイルツリーに接続できるマウントポイントディレクトリも入っている。 |
| /usr | UFS | 他のユーザーと共有できるシステムファイルとディレクトリ。特定のタイプのシステム上でのみ実行できるファイルは、/usr ディレクトリに入っている (SPARC 実行可能ファイルなど)。どのタイプのシステム上でも使用できるファイル (マニュアルページなど) は、/usr/share に入っている。 |

表 26-2 デフォルトの Solaris ファイルシステム 続く

| ファイルシステムまたはディレクトリ | ファイルシステムのタイプ | 説明 |
|---------------------------|--------------|--|
| /export/home または /home | NFS、UFS | ユーザーのホームディレクトリのマウントポイント。ホームディレクトリには、そのユーザーの作業ファイルが格納される。デフォルトでは、/home は自動マウントされるファイルシステムである。スタンドアロンシステム上では、/home はローカルディスクスライス上の UFS ファイルシステムの場合がある。 |
| /var | UFS | ローカルシステムの使用中に変化または拡大する可能性のあるシステムファイルとディレクトリ。これには、システムログ、vi と ex のバックアップファイル、および uucp ファイルが含まれる。 |
| /opt | NFS、UFS | オプションの Sun 以外のソフトウェア製品のマウントポイント。システムによっては、/opt がローカルディスクスライス上の UFS ファイルシステムの場合がある。 |
| /tmp | TMPFS | システムがブートされるたびにまたは /tmp ファイルシステムがマウント解除されるたびに消去される一時ファイル |
| /proc | PROCFS | アクティブなプロセスの番号別リスト |

スワップ空間

Solaris オペレーティング環境は、一部のディスクスライスをファイルシステムではなく一時記憶域として使用します。これらのスライスを「スワップスライス」または「スワップ空間」と呼びます。スワップスライスは、現在のプロセスを処理するだけの十分な物理メモリーがシステムにない場合に、仮想メモリー記憶域として使用されます。

多くのアプリケーションは十分なスワップ空間が使用できることを前提に作成されているため、スワップ空間を割り当て、その使われ方を監視して、必要に応じてスワップ空間を追加する方法を知っておくことは大切です。スワップ空間の概要とスワップ空間を追加する手順については、第 30 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。

- 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り、必要のないチェックをファイルシステム上で行わなくて済みます。ファイルシステムが「クリーン」状態、「安定」状態、または「ロギング処理」状態になっていると、ファイルシステムのチェックは実行されません。
- 拡張基礎タイプ (EFT) – 32 ビットのユーザー ID (UID)、グループ ID (GID)、およびデバイス番号
- 大規模ファイルシステム – UFS ファイルシステムの最大サイズは 1T バイト (テラバイト) です。Solaris オペレーティング環境では、論理スライスの大きさを 1T バイトのファイルシステムに対応させるストライプ機能はサポートされていませんが、Sun 提供の Solstice DiskSuite™ ソフトウェアには、この機能が含まれています。
- 大規模ファイル – デフォルトでは、UFS ファイルシステムは 2G バイト (ギガバイト) を超える通常ファイルを持つことができます。2G バイトの最大ファイルサイズ制限を有効にするには、`nolargefiles` マウントオプションを明示的に使用しなければなりません。

UFS ファイルシステムの詳細は、第 32 章を参照してください。

UFS ファイルシステムの構成

UFS ファイルシステムを作成すると、ディスクスライスは、1 つまたは複数の連続するディスクシリンダから構成されるシリンダグループに分割されます。シリンダグループはさらに、位置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。

- ブートブロック – システムのブート時に使用される情報が格納される。

- 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が格納される。
- i ノード - ファイル名を除く、ファイルに関するすべての情報が格納される。
- 記憶域またはデータブロック - 各ファイルのデータが格納される。

各ブロックタイプの詳細については、565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。

`newfs` コマンドまたは `mkfs` コマンドの引数を使用してファイルシステムをカスタマイズするには、これらのパラメータの変更方法について第 32 章を参照してください。

UFS ロギング

UFS ロギングは、トランザクション (完全な UFS 操作を構成する変更) をログに保存してから、そのトランザクションを UFS ファイルシステムに適用するプロセスです。保存されたトランザクションは、後でファイルシステムに適用できます。

システムはリブート時に、不完全なトランザクションを廃棄しますが、完結している操作のトランザクションは適用します。完結しているトランザクションだけが適用されるために、ファイルシステムの整合性が保たれます。通常であればシステムコールの実行が中断され、UFS ファイルシステムの整合性が確保できないシステムクラッシュ時にも、ファイルシステムの整合性が保たれます。

UFS ロギングには 2 つの長所があります。まず、ファイルシステムの整合性が保持されるため、`fsck (1M)` を実行する必要がなくなります。また、`fsck` を省略できるため、システムがクラッシュしたり、異常停止 (クリーンではない停止) した場合でもシステムをリブートするのに要する時間を短縮できます。クリーンではない停止についての詳細は、536ページの「`fsck` でチェックして修復される内容」を参照してください。UFS ロギングを使用すれば、特に、通常であれば `fsck` による読み込みと確認に時間がかかる大規模ファイルシステムを持つシステムのブート時間を短縮できます。

UFS ロギングが作成するログは、いっぱいになるとフラッシュされます。また、ファイルシステムがマウント解除されたとき、あるいは `lockfs -f` コマンドを実行したときにも、ログはすべてフラッシュされます。

デフォルトでは、UFS ロギングは無効です。UFS ロギングを有効にするには、ファイルシステムをマウントするときに、`-o logging` オプションを指定して `mount (1M)` コマンドを実行しなければなりません。ログはファイルシステムの空きブロックから割り当てられ、1G バイトのファイルシステムごとに約 1M バイトの

サイズ (合計で 64M バイトまで) が割り当てられます。ログは、ルート (/) ファイルシステムを含む、任意の UFS で有効にできます。また、`fsdb` コマンドには、UFS ロギングをサポートするための新しいデバッグオプションが追加されています。

UFS ファイルシステムの計画

ファイルシステムの配置を決めるときには、要求が競合する可能性があることを考えなければなりません。推奨事項は次の通りです。

- 作業負荷を異なる入出力システムやディスクドライブ間でできるだけ均等に分散します。`/export/home` を分散させたり、ディレクトリを入れ替えたりして、異なるディスクに負荷を均等に割り当てます。
- プロジェクトの個々の部分やグループのメンバーを同じファイルシステム内に入れます。
- 1 ディスク当たりのファイルシステム数をできるだけ少なくします。通常、システム (またはブート) ディスク上には、`/`、`/usr`、スワップ空間の 3 つのファイルシステムがあります。他のディスク上では、1 つまたは通常は 2 つのスライスを作成します。多数の小型ファイルシステムに分割しすぎるよりもファイルシステム数を少なくして余地を設ける方が、ファイルがフラグメントに分割される可能性が小さくなります。容量の大きいテープドライブを使用し、`ufsdump` で複数のボリュームを処理できるようにしておけば、大型ファイルシステムでも簡単にバックアップをとることができます。
- 絶えずきわめて小さいファイルを作成するユーザーがいる場合は、iノード数を増やして別のファイルシステムを作成することを検討します。ただし、ほとんどのサイトでは、類似するタイプのファイルを同じファイルシステム内で保管するようにする必要はありません。

デフォルトのファイルシステムパラメータや、新しいファイルシステムを作成するときの前提条件と手順については、第 27 章を参照してください。

ファイルシステムのマウントとマウント解除

ファイルシステム上のファイルにアクセスするには、ファイルシステムをマウントする必要があります。ファイルシステムのマウントとは、ファイルシステムをディレクトリ (マウントポイント) に接続し、システムで使用可能にすることです。ルー

ト (/) ファイルシステムは、常にマウントされています。他のファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりすることができます。

ファイルシステムをマウントすると、そのファイルシステムがマウントされている間は、マウントポイントのディレクトリ内に実際に存在しているファイルやディレクトリは使用できなくなります。これらのファイルはマウント処理の影響を永続的に受けるわけではなく、ファイルシステムをマウント解除すると再び使用できるようになります。通常は存在はするがアクセスできないファイルは混乱の原因になるので、マウントディレクトリを空にしておきます。

次の例では、ルート (/) ファイルシステムの下に、サブディレクトリ `sbin`、`etc`、および `opt` を持つローカルのファイルシステムを示します。

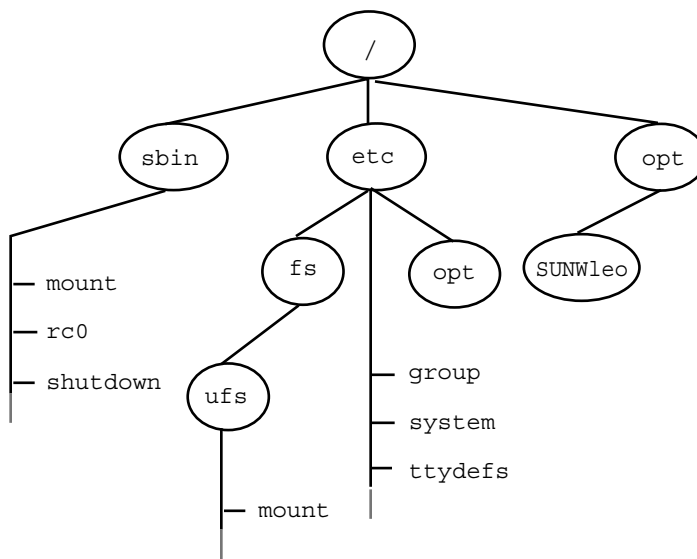


図 26-1 ルート (/) ファイルシステムの例

ここでは、標準添付されていない製品セットが含まれている `/opt` ファイルシステムからローカルファイルシステムにアクセスするものとします。

まず、ファイルシステムをマウントするためのマウントポイントとして使用するディレクトリを作成しなければなりません (たとえば、`/opt/unbundled`)。マウントポイントを作成すると、`mount` コマンドでファイルシステムをマウントでき、`/opt/unbundled` 内のすべてのファイルとディレクトリにアクセスできるようになります (図 26-2 を参照)。これらの作業を実行するための詳細な手順については、第 28 章を参照してください。

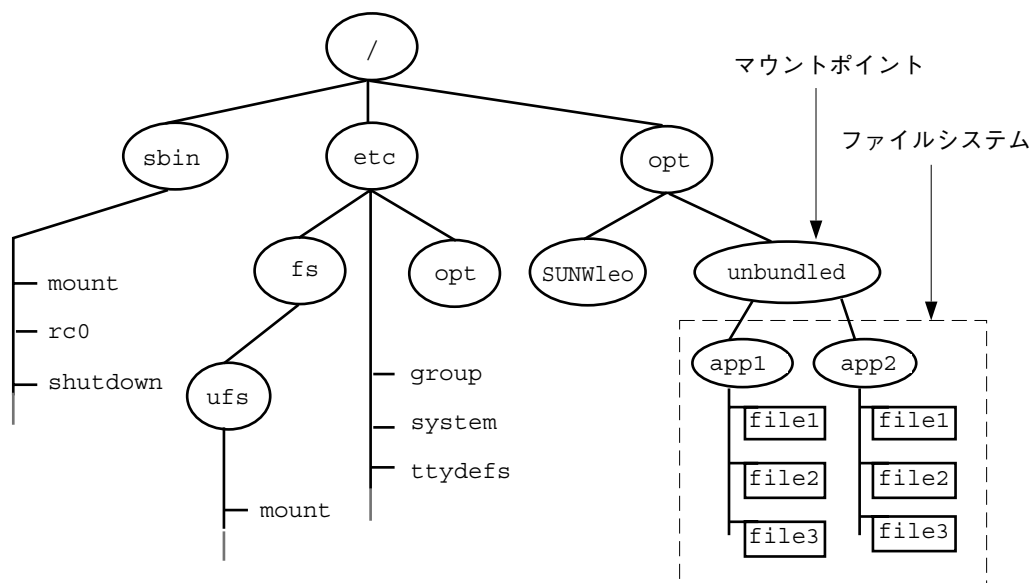


図 26-2 ファイルシステムのマウント

ファイルシステムをマウント解除すると、そのファイルシステムはマウントポイントから削除されます。一部のファイルシステム管理作業は、マウントされたファイルシステム上では実行できません。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。

- 不要になった場合や、最新ソフトウェアが入っているファイルシステムに交換した場合。
- `fsck` コマンドを使用して検査し修復する場合。`fsck` コマンドの詳細は、第 31 章を参照。

ファイルシステムの完全バックアップを実行する前に、そのファイルシステムを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方法については、第 33 章を参照してください。

注・ファイルシステムは、システムのシャットダウン手続きの一部として自動的にマウント解除されます。

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。マウントテーブルの内容は `cat` または `more` コマンドを使用して表示できますが、`/etc/vfstab` ファイルのように編集することはできません。次にマウントテーブルファイル `/etc/mnttab` の例を示します。

```
$ more /etc/mnttab
/dev/dsk/c0t3d0s0 /      ufs      rw,suid,dev=800018,largefiles 863804345
/dev/dsk/c0t3d0s6 /usr  ufs      rw,suid,dev=80001e,largefiles 863804345
/proc /proc  proc    rw,suid,dev=2900000      863804345
fd    /dev/fd fd      rw,suid,dev=29c0000     863804345
/dev/dsk/c0t3d0s3 /export ufs      suid,rw,largefiles,dev=80001b 863804347
/dev/dsk/c0t3d0s7 /export/home ufs suid,rw,largefiles,dev=80001f 863804348
/dev/dsk/c0t3d0s4 /export/swap ufs suid,rw,largefiles,dev=80001c 863804348
/dev/dsk/c0t3d0s5 /opt  ufs      suid,rw,largefiles,dev=80001d 863804347
swap  /tmp   tmpfs    dev=2a80000      863804347
$
```

仮想ファイルシステム

アクセスするたびにファイルシステムを手動でマウントするのは、時間のかかる作業です。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) が作成されました。このファイルに、マウントするファイルシステムとそのマウント方法が指定されます。`/etc/vfstab` ファイルは、2つの重要な機能を持っています。1つは、システムブート時に自動的にマウントするファイルシステムを指定できることです。もう1つは、マウントポイント名だけでファイルシステムをマウントできることです。これは、`/etc/vfstab` ファイルにマウントポイントと実際のデバイススライス名とのマッピングを指定することより可能になります。

デフォルトの `/etc/vfstab` ファイルは、システムをインストールするときに作成され、その内容はシステムソフトウェアをインストールするときに行なった選択によって異なります。ただし、システムの `/etc/vfstab` ファイルはいつでも編集できます。エントリを追加するときに指定する必要がある主な情報は、ファイルシステムが置かれているデバイス、マウントポイントの名前、ファイルシステムのタイプ、システムブート時に自動的にブートするかどうか (`mountall(1M)` コマンドを使用する)、およびマウントオプションです。

次の例は、`/etc/vfstab` ファイルの内容を示しています。コメント行は `#` で始まります。この例は、2つのディスク (`c0t0d0` と `c0t3d0`) を持つシステムの `/etc/vfstab` ファイルです。

コード例 26-1 /etc/vfstab ファイルの例

```

$ 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 /      ufs     1      no     -
/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    -
$

```

コード例 26-1 の最後のエントリーは、システムブート時に /dev/dsk/c0t2d0s7 スライス上の UFS ファイルシステムを自動的にマウントポイント /test にマウントすることを指定しています。ルート (/) と /usr では、「mount at boot」フィールドの値が no に指定されていることに注意してください。これはこの2つのファイルシステムが、mountall コマンドを実行する前にブートシーケンスの一部としてカーネルによってマウントされるからです。

/etc/vfstab の各フィールド、このファイルの編集方法、使用方法については、第 28 章を参照してください。

ネットワークファイルシステム (NFS)

NFS は、1 つのシステム (通常はサーバー) の資源 (ファイルやディレクトリ) をネットワーク経由で他のシステムと共有するための分散型ファイルシステムです。たとえば、Sun 以外のアプリケーションやソースファイルを他のシステム上のユーザーと共有できます。

NFS は、資源の実際の物理的な位置をユーザーが意識しなくてすむようにします。共通に使用されるファイルのコピーをシステムごとに配置しなくても、あるシステムのディスク上にコピーを 1 つ配置することによって NFS は、他のすべてのシステムがそのコピーにネットワーク経由でアクセスできるようにします。NFS の環境では、リモートファイルシステムは、実際にはローカルシステムと区別がつかいません。

システムは、ネットワークに対して共有またはエクスポートするファイルシステムを持っているときに、NFS サーバーになります。サーバーは、現在エクスポートされている資源とアクセス制限 (読み取り/書き込み、読み取り専用など) のリストを管理します。

資源を共有する場合は、リモートシステムにマウントできるように、その資源を使用可能な状態にします。

資源を共有するには、次の方法があります。

- `share(1M)` コマンドまたは `shareall(1M)` コマンドを使用する
- `/etc/dfs/dfstab` (分散ファイルシステムテーブル) ファイルにエントリを追加する

資源を共有する方法については、第 28 章を参照してください。NFS の詳細な説明については、『NFS の管理』を参照してください。

自動マウント (AutoFS)

NFS ファイルシステム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 ファイルシステム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、ファイルシステムはマウントされたままになります。ファイルシステムが一定の時間アクセスされなかった場合、ファイルシステム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 ファイルシステム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 ファイルシステムをマウントまたはマウント解除するために、`root` のパスワードを知っている必要はありません。
- NFS ファイルシステムは使用されるときにだけマウントされるために、ネットワークトラフィックが軽減されます。

AutoFS サービスは `automount(1M)` によって初期化され、`automount` はシステムのブート時に自動的に実行されます。自動マウントデーモン `automountd(1M)` は永続的に動作し、必要に応じて NFS ファイルシステムをマウントまたはマウント解除します。デフォルトでは、Solaris オペレーティング環境は `/home` を自動マウントします。

AutoFS は、ネームサービスに指定されたファイルシステムに対して動作します。この情報は、NIS、NIS+、またはローカルの `/etc` 内のファイルで管理されます。AutoFS を使用すると、同じファイルシステムを提供するサーバーを複数指定できます。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。各サーバーに優先度を示す数

値を割り当てることによって、マップ内の各資源ごとにサーバーの優先順位を指定できます。

AutoFS の設定と管理方法については、『NFS の管理』を参照してください。

キャッシュファイルシステム (CacheFS)

NFS マウントされたファイルシステムの性能とスケーラビリティを向上させるには、キャッシュファイルシステム (CacheFS) を使用します。CacheFS は、サーバーとネットワークの負荷を軽減することによって、NFS サーバーの性能とスケーラビリティを向上させる汎用ファイルシステムキャッシュ機構です。

CacheFS は、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS はサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。また、CacheFS と AutoFS サービスを併用すると、さらにパフォーマンスとスケーラビリティを向上することができます。

CacheFS についての詳細は、第 29 章を参照してください。

マウント方法の決定

表 26-3 に、ファイルシステムの用途に応じてファイルシステムをマウントするためのガイドラインを示します。

表 26-3 ファイルシステムのマウント方法の決定

| マウントするファイルシステムとマウントの頻度 | 方法 |
|---------------------------------------|--|
| ローカルまたはリモートのファイルシステムをときどきマウントする | コマンド行から手動で <code>mount</code> コマンドを入力する。 |
| ローカルのファイルシステムを頻繁にマウントする | <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに自動的にファイルシステムをマウントする。 |
| リモートのファイルシステムを頻繁にマウントする (ホームディレクトリなど) | <ul style="list-style-type: none"> ■ <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに自動的にファイルシステムをマウントする。 ■ AutoFS により、ユーザーがディレクトリに移動したとき (マウント)、あるいはディレクトリから移動したとき (マウント解除) に、自動的にファイルシステムをマウントまたはマウント解除する。 <p>パフォーマンスを向上させるには、CacheFS を使用してリモートのファイルシステムをキャッシュすることもできる。</p> |

ファイルシステムを含む CD-ROM は、単にドライブに挿入するだけでマウントできます (ボリューム管理が自動的にマウントします)。ファイルシステムを含むフロッピーディスクは、ドライブに挿入して `volcheck` コマンドを実行することによりマウントできます。詳細は、パート IV 「取り外し可能な媒体の管理」を参照してください。

ファイルシステムのタイプを調べる

ファイルシステムのタイプは、次のいずれかを使用して判断できます。

- 仮想ファイルシステムテーブル (`/etc/vfstab` ファイル) 内の FS type フィールド
- ローカルファイルシステムの `/etc/default/fs` ファイル
- NFS ファイルシステムの `/etc/dfs/fstypes` ファイル

▼ ファイルシステムのタイプを調べる方法

この手順は、ファイルシステムがマウントされているかどうかに関わらず使用できます。

grep コマンドを使用して、ファイルシステムのタイプを調べます。

```
$ grep mount-point fs-table
```

mount-point

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。

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指定する。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 デバイス名を知っている場合、fstyp(1M) コマンドで、ファイルシステムのタイプを調べることができます (そのディスクスライスにファイルシステムが含まれている場合)。

例 — ファイルシステムのタイプを調べる

次の例は、/etc/vfstab を使用して、/export ファイルシステムのタイプを調べます。

```
$ grep /export /etc/vfstab
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /export ufs 2 yes -
$
```

次の例は、/etc/mnttab ファイルを使用して、現在マウントしている (ボリューム管理でマウントした) フロッピーディスクのファイルシステムのタイプを調べます。

```
$ grep /floppy /etc/mnttab
/vol/dev/diskette0/unnamed_floppy /floppy/
unnamed_floppy pcfs rw,nohidden,
nofoldcase,dev=16c0009      89103376
$
```

ファイルシステムの作成手順

この章では、UFS、TMPFS、および LOFS ファイルシステムを作成する方法について説明します。UFS ファイルシステムの場合、この章では、`newfs(1M)` コマンドを使用してファイルシステムをハードディスク上に作成する方法を示します。TMPFS と LOFS は仮想ファイルシステムであるため、これらのファイルシステムを「作成」するには、ファイルシステムをマウントします。

この章で説明する手順は、次のとおりです。

- 455ページの「UFS ファイルシステムを作成する方法」
- 457ページの「一時ファイルシステムを作成する方法」
- 459ページの「LOFS ファイルシステムを作成する方法」

注 - UFS と DOS のファイルシステムを取り外し可能な媒体上に作成する手順については、第 11 章を参照してください。

UFS ファイルシステムの作成

UFS ファイルシステムをディスクに作成するためには、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクスライスとは物理的なディスクのサブセットで、連続するブロックからなる 1 つの範囲のことです。スライスはスワップ空間などの `raw` デバイスとして使用することも、ディスクベースのファイルシステムとして使用することもできます。ディスクのフォーマットとディスクのスライスへの分割についての詳細は、第 21 章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。SunOS 4.0 およびその互換バージョンでは、スライスは「パーティション」と呼ばれていました。

UFS ファイルシステムは、インストール手順の一部として Solaris オペレーティング環境によって自動的に作成されるので、UFS ファイルシステムを作成しなければならないことはほとんどありません。次の場合には、UFS ファイルシステムを作成する (または作成し直す) 必要があります。

- ディスクを追加または交換する場合
- 既存のパーティション構造を変更する場合
- ファイルシステム全体を復元する場合

`newfs (1M)` コマンドを使用するのが、UFS ファイルシステムを作成する標準的な方法です。`newfs` コマンドは `mkfs (1M)` の使いやすいフロントエンドで、新しいファイルシステムを作成します。Solaris システムの場合、デフォルトでは 1 シリンダ当たりのトラック数や 1 トラック当たりのセクター数のような `newfs (1M)` のパラメータは、新しいファイルシステムを作成するディスクのラベルから読み込まれ、ユーザーが選択したオプションは、`mkfs` コマンドに渡されファイルシステムが作成されます。

ファイルシステムパラメータ

ディスクスライス上に新しいファイルシステムを作成するには、ほとんどの場合に `newfs` コマンドを使用します。表 27-1 に、`newfs` コマンドで使用するデフォルトのパラメータを示します。

表 27-1 `newfs` コマンドで使用するデフォルトのパラメータ

| パラメータ | デフォルト値 |
|-----------|--|
| ブロックサイズ | 8K バイト |
| フラグメントサイズ | 1K バイト |
| 最小空き領域 | ((64M バイト/パーティションサイズ) * 100) で算出した値を最も近い整数に切り捨てる。値は、パーティションサイズの 1% から 10% の範囲に制限される。 |

表 27-1 newfs コマンドで使用するデフォルトのパラメータ 続く

| パラメータ | デフォルト値 |
|---------|----------------------|
| 回転遅延 | デバイスに依存 |
| 最適化のタイプ | 容量 |
| i ノード数 | 2K バイトのディスク領域ごとに 1 個 |

▼ UFS ファイルシステムを作成する方法

- 次の前提条件を満たしているかどうかを確認します。
 - UFS ファイルシステムを作成する前に、ディスクをフォーマットしてスライスに分割しておかなければならない。ディスクのフォーマットとスライスへの分割についての詳細は、第 21 章を参照してください。
 - ファイルシステムを格納するスライスのデバイス名を知っていなければならない。ディスク番号とディスクスライス番号を調べる方法については、第 22 章を参照してください。
 - 既存の UFS ファイルシステムを作成し直す場合は、そのマウントを解除する。
 - スーパーユーザーでなければならない。
- ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdisk/device-name
```

| | |
|-------------|---|
| -N | newfs が mkfs 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です。 |
| -b size | ファイルシステムのブロックサイズを指定する。デフォルトは 8192 バイトです。 |
| -i bytes | i ノード当たりのバイト数を指定する。デフォルトは 2048 バイトです。 |
| 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。 |



注意 - 次の手順を実行する前に、スライスに合ったデバイス名を指定したか確認してください。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。

システムから、確認を促すプロンプトが表示されます。

- UFS** ファイルシステムが作成されていることを確認するには、`fsck(1M)` コマンドを使用して新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

device-name 新しいファイルシステムを格納するディスクデバイス名を指定する。

`fsck` コマンドは、新しいファイルシステムの整合性をチェックして、問題が検出された場合には、問題を修復する前にプロンプトを表示します。`fsck` についての詳細は、第 31 章を参照してください。

例 — UFS ファイルシステムを作成する

次の例では、`/dev/rdisk/c0t3d0s7` 上に UFS ファイルシステムを作成します。

```
# newfs /dev/rdisk/c0t3d0s7
newfs: construct a new file system /dev/rdisk/c0t3d0s7 (y/n)? y
```

(続く)

```
/dev/rdisk/c0t3d0s7:    163944 sectors in 506 cylinders of 9 tracks,  
    36 sectors 83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)  
super-block backups (for fsck -b #) at:  
    32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,  
    47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,  
    93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064,  
    135296, 140528, 145760, 150992, 156224, 161456,
```

次に進む手順

ファイルシステムをマウントし、使用可能にするには、第 28 章に進みます。

一時ファイルシステム (TMPFS) の作成

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルのメモリーを使用します。そのため、一時ファイルシステムは、UFS ファイルシステムに比べはるかに高速です。TMPFS ファイルシステムを使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドが軽減されるのでシステムのパフォーマンスを向上できます。TMPFS ファイルシステム内のファイルは、リブートまたはマウント解除すると削除されます。

複数の TMPFS ファイルシステムを作成した場合は、すべてのファイルシステムが同じシステム資源を使用するという点に注意してください。mount コマンドの `-o size` オプションを使用して TMPFS のサイズを制限しなければ、ある TMPFS ファイルシステムで作成されたファイルが、他の TMPFS のための領域を使い切ってしまう可能性があります。

詳細は、`tmpfs(7FS)` のマニュアルページを参照してください。

▼ 一時ファイルシステムを作成する方法

1. スーパーユーザーになります。
2. 必要に応じて、**TMPFS** ファイルシステムをマウントしたいディレクトリを作成し、アクセス権と所有権を設定します。

3. TMPFSファイルシステムを作成します。

ブート時に一時ファイルシステムを自動的に作成するようにシステムを設定するには、458ページの「例 — ブート時に TMPFS ファイルシステムを作成する」を参照してください。

```
# mount -F tmpfs -o size=number swap mount-point
```

size=number TMPFS ファイルシステムのサイズを M バイト単位で指定する。

mount-point TMPFS ファイルシステムとしてマウントするディレクトリを指定する。

4. mount コマンドからの出力を調べて、TMPFS ファイルシステムが作成されていることを確認します。

```
# mount -v
```

例 — 一時ファイルシステムを作成する

次の例は、新しいディレクトリ /export/reports を作成し、そのマウントポイントに TMPFS ファイルシステムをマウントして、50M バイトに制限します。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50 swap /export/reports
```

例 — ブート時に TMPFS ファイルシステムを作成する

ブート時にシステムが自動的に TMPFS ファイルシステムを作成するようにするには、/etc/vfstab ファイルにエントリを追加します。次の例は、システムのブート時に TMPFS ファイルシステムを /export/test に作成する /etc/vfstab ファイルのエントリを示します。size=number オプションを指定していないため、/export/test の TMPFS ファイルシステムのサイズは利用できるシステム資源によって制限されます。

```
swap - /export/test tmpfs - yes -
```

/etc/vfstab ファイルについての詳細は、466ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。

ループバックファイルシステム (LOFS) の作成

LOFS ファイルシステムは、既存のファイルシステムへの代替パスを提供する仮想ファイルシステムです。他のファイルシステムを LOFS ループバックファイルシステムにマウントしても、元のファイルシステムは変化しません。

詳細は、`lofs(7FS)` のマニュアルページを参照してください。



注意 - LOFS ファイルシステムは慎重に作成してください。LOFS は仮想ファイルシステムなので、ユーザーやアプリケーションを混乱させる可能性があります。

▼ LOFS ファイルシステムを作成する方法

1. スーパーユーザーになります。
2. **LOFS** ファイルシステムをマウントしたいディレクトリを作成し、適切なアクセス権と所有権を設定します。
3. **LOFS** ファイルシステムを作成します。
ブート時にループバックファイルシステムを自動的に作成するようにシステムを設定するには、460ページの「例 — ブート時に LOFS ファイルシステムを作成する」を参照してください。

```
# mount -F lofs loopback-directory mount-point
```

loopback-directory

ループバックマウントポイントにマウントするファイルシステムを指定する。

mount-point

LOFS ファイルシステムをマウントするディレクトリを指定する。

4. `mount` コマンドからの出力を調べて、**LOFS** ファイルシステムが作成されていることを確認します。

```
# mount -v
```

例 — LOFS ファイルシステムを作成する

次の例は、`chroot (1M)` コマンドと新しい LOFS ファイルシステムを使用して、プロセスやプロセスファミリが完全な仮想ファイルシステムを認識できるようにする。

```
# mkdir /tmp/newroot  
# mount -F lofs / /tmp/newroot  
# chroot /tmp/newroot command
```

例 — ブート時に LOFS ファイルシステムを作成する

ブート時にシステムが自動的に LOFS ファイルシステムを作成できるようにするには、`/etc/vfstab` ファイルにエントリを追加します。次の例は、ルート (`/`) ファイルシステムの LOFS ファイルシステムを `/tmp/newroot` に作成する `/etc/vfstab` ファイルのエントリを示しています。

```
/ - /tmp/newroot lofs - yes -
```



注意 - ループバックファイルシステムのエントリは、`/etc/vfstab` ファイル内の最後のエントリでなければなりません。ループバックファイルシステムのエントリが、そこに組み込まれるファイルシステムよりも前にあると、ループバックファイルシステムを作成できません。

`/etc/vfstab` ファイルの詳細は、466ページの「`/etc/vfstab` ファイルのフィールドの説明」を参照してください。

ファイルシステムのマウントとマウント解除の手順

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。

- 465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」
- 467ページの「/etc/vfstab ファイルにエントリを追加する方法」
- 469ページの「/etc/vfstab ファイル内に列挙されている複数のファイルシステムをマウントする方法」
- 471ページの「/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する方法」
- 472ページの「UFS ファイルシステムをマウントする方法」
- 474ページの「NFS ファイルシステムをマウントする方法」
- 475ページの「x86: System V (S5FS) ファイルシステムをマウントする方法」
- 476ページの「ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法」
- 479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」
- 480ページの「ファイルシステムをマウント解除する方法」
- 481ページの「/etc/vfstab ファイルに列挙されている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

ファイルシステムを作成したら、そのファイルシステムをシステムで使用できるようにする必要があります。ファイルシステムを使用できるようにするには、マウントします。マウントしたファイルシステムは、システムのディレクトリツリー内の指定したマウントポイントに接続されます。ルート (/) ファイルシステムは常にマウントされています。他のファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりできます。

表 28-1 に、ファイルシステムの用途に応じてファイルシステムをマウントするためのガイドラインを示します。

表 28-1 ファイルシステムのマウント方法の決定

| マウントするファイルシステムとマウントの頻度 | 方法 |
|---------------------------------------|---|
| ローカルまたはリモートのファイルシステムをときどきマウントする | コマンド行から手動で <code>mount</code> コマンドを入力する。 |
| ローカルのファイルシステムを頻繁にマウントする | <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに、自動的にファイルシステムをマウントする。 |
| リモートのファイルシステムを頻繁にマウントする (ホームディレクトリなど) | <ul style="list-style-type: none">■ <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに、自動的にファイルシステムをマウントする。■ AutoFS により、ユーザーがディレクトリに移動したとき (マウント)、あるいはディレクトリから移動したとき (マウント解除) に、自動的にファイルシステムをマウントまたはマウント解除する。 パフォーマンスを向上させるには、CacheFS を使用してリモートのファイルシステムをキャッシュすることもできる。 |

ファイルシステムを含む CD-ROM は、単にドライブに挿入するだけでマウントできます (ボリューム管理が自動的にマウントします)。ファイルシステムを含むフロッピーディスクは、ドライブに挿入して `volcheck(1)` コマンドを実行することによりマウントできます。詳細は、第 11 章を参照してください

ファイルシステムのマウントとマウント解除に使用するコマンド

表 28-2 に、ファイルシステムのマウントとマウント解除に使用する `/usr/sbin` ディレクトリ内のコマンドを示します。

表 28-2 ファイルシステムのマウントとマウント解除に使用するコマンド

| コマンド | 説明 |
|-----------------------------|---|
| <code>mount (1M)</code> | ファイルシステムとリモート資源をマウントする。 |
| <code>mountall (1M)</code> | <code>/etc/vfstab</code> ファイル内で指定されたすべてのファイルシステムをマウントする。マルチユーザー実行状態になる時に、 <code>mountall</code> コマンドが自動的に実行される。 |
| <code>umount (1M)</code> | ファイルシステムとリモート資源をマウント解除する。 |
| <code>umountall (1M)</code> | <code>/etc/vfstab</code> ファイル内で指定されたすべてのファイルシステムをマウント解除する。 |

`mount` コマンドは、整合性のない読み取り/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 31 章を参照してください。

`umount` コマンドは、使用中のファイルシステムはマウント解除しません。ユーザーがファイルシステム内のディレクトリをカレントディレクトリにしているか、あるいは、プログラムがそのファイルシステム上にあるファイルをオープンしている場合、そのファイルシステムは、使用中であると認識されます。

汎用マウントオプション

表 28-3 に、`mount` コマンドの `-o` オプションで指定できる汎用マウントオプションを示します。複数のオプションを指定する場合は、`-o ro,nosuid` のようにコマンド (空白を入れない) で区切ります。

各ファイルシステムタイプで指定可能なマウントオプションのリストについては、各マウントコマンドのマニュアルページ (たとえば、`mount_ufs (1M)`) を参照してください。

表 28-3 -o で指定する汎用マウントオプション

| オプション | ファイルシステム | 説明 |
|---------------------------|-------------------------------|--|
| bg fg | NFS | 最初の試行に失敗すると、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。デフォルトは fg。 |
| f | UFS | /etc/mnttab 内にエントリを作成するが、実際にはファイルシステムをマウントしない。 |
| 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 は、エラーが返されることを示す。hard 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。 |
| 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ファイルシステムを作成できるようにする。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)。デフォルトは largefiles。 |
| logging nologging | UFS | <p>ファイルシステムのロギングを有効にする。UFS ロギングとは、トランザクション (完全な UFS 操作を構成する変更) をログに保存してから、そのトランザクションを UFS ファイルシステムに適用するプロセスである。ファイルシステムの整合性が保たれるために、fsck(1M) を省略できる。したがって、システムがクラッシュしたとき、あるいは、不完全な停止後にシステムをリブートする際の時間を短縮できる。</p> <p>ログはファイルシステムの空きブロックから、1G バイトのファイルシステムごとに約 1M バイトのサイズ (合計で 64M バイトまで) が割り当てられる。デフォルトは nologging。</p> |
| m | CacheFS, NFS, PCFS, S5FS, UFS | /etc/mnttab にエントリを作成せずに、ファイルシステムをマウントする。 |

表 28-3 -o で指定する汎用マウントオプション 続く

| オプション | ファイルシステム | 説明 |
|-----------------|-------------------------------|---|
| noatime | UFS | ファイルのアクセス時間更新を無視する。ただし、ctime または mtime の更新が同時に行れると場合を除く。stat(2) を参照。このオプションによって、アクセス時間が重要でないファイルシステム (たとえば、Usenet ニュースプール) でのディスクに対する動作が減る。デフォルトでは、通常アクセス時間 (atime) が記録される。 |
| remount | NFS, S5FS, UFS | 読み取り専用のファイルシステムを読み書き可能として (rw オプションで) 再マウントする。このオプションは、f、logging nologging、および m オプションと組み合わせる場合にだけ指定できる。このオプションは現在マウントされている読み取り専用のファイルシステムに対してのみ動作する。 |
| retry= <i>n</i> | NFS | マウント操作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, S5FS | 読み取り/書き込みまたは読み取り専用と指定する。このオプションを指定しない場合のデフォルトは読み取り/書き込み。 |
| suid nosuid | CacheFS, HSFS, NFS, S5FS, 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。 |

▼ どのファイルシステムがマウントされているかを調べる方法

どのファイルシステムがマウントされているかを調べるには、mount (1M) コマンドを使用します。

```
$ mount -v
```

-v

マウントされているファイルシステムのリストを冗長モードで表示する

例 — どのファイルシステムがマウントされているかを調べる

```
$ mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid/largefiles on ...
/usr on /dev/dsk/c0t3d0s6 read/write/setuid/largefiles on ...
/proc on /proc read/write/setuid on Mon Jun 8 10:28:31 1998
/dev/fd on fd read/write/setuid on Mon Jun 8 10:28:31 1998
/export on /dev/dsk/c0t3d0s3 setuid/read/write/largefiles on ...
/export/home on /dev/dsk/c0t3d0s7 setuid/read/write/largefiles on ...
/export/swap on /dev/dsk/c0t3d0s4 setuid/read/write/largefiles on ...
/opt on /dev/dsk/c0t3d0s5 setuid/read/write/largefiles on ...
/tmp on swap read/write on Mon Jun 8 10:28:31 1998
$
```

/etc/vfstab ファイルでファイルシステムをマウントする

/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、表 28-4 に示すように7つのフィールドがあります。

表 28-4 /etc/vfstab ファイルのフィールドの説明

| フィールド名 | 説明 |
|-----------------|---|
| device to mount | <ul style="list-style-type: none">■ ローカル UFS ファイルシステム用のブロックデバイス名 (/dev/dsk/c0t0d0s0 など)■ リモートファイルシステム用の資源名 (myserver:/export/home など)。NFS の詳細は、『NFS の管理』を参照してください。■ スワップ空間用のスライスのブロックデバイス名 (/dev/dsk/c0t3d0s1 など)■ /proc ディレクトリと proc ファイルシステムのタイプ |
| device to fsck | 「device to mount」フィールドで指定した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)。これにより、fsck が使用するインタフェースが決まる。読み取り専用ファイルシステムやリモートファイルシステムなど、適用できるデバイスがない場合は、ダッシュ (-) を使用する。 |

表 28-4 /etc/vfstab ファイルのフィールドの説明 続く

| フィールド名 | 説明 |
|---------------|---|
| mount 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 | 「device to mount」フィールドで指定したファイルシステムのタイプを指定する。 |
| fsck pass | <p>fsck がファイルシステムをチェックするか決めるために使用するパス番号。このフィールドでダッシュ (-) を指定すると、ファイルシステムはチェックされない。</p> <p>このフィールドに 0 が指定されている場合、UFS ファイルシステムはチェックされないが、フィールドに 0 より大きい値が指定されている場合に UFS 以外のファイルシステムはチェックされる。</p> <p>このフィールドに 1 が指定されている場合、ファイルシステムは順番にチェックされる。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 は複数のディスク上のファイルシステムを並行して自動的にチェックする。それ以外の場合、このフィールドの値は意味を持たない。fsck pass フィールドでは、ファイルシステムをチェックする順序を明示的には指定できない。</p> |
| mount at boot | システムのブート時にファイルシステムが mountall によって自動的にマウントされるかどうかを表す。yes または no に設定。このフィールドは AutoFS とは連動していないので注意してください。 |
| mount options | ファイルシステムのマウントに使用されるオプションを (空白を空けずに) コンマで区切ったリスト。オプションなしを示すにはダッシュ (-) を使用する。汎用オプションについては、表 28-3 を参照。 |

注 - /etc/vfstab ファイル内のフィールドごとに 1 つずつエントリがなければなりません。フィールドで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。

▼ /etc/vfstab ファイルにエントリを追加する方法

1. スーパーユーザーになります。

また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

2. /etc/vfstab ファイルを編集してエントリを追加します。

注 - ルート (/) ファイルシステムは、ブートプロセスの過程でカーネルによって読み取り専用としてマウントされます。そのため、remount オプション (および、remount と一緒に使用できるオプション) だけが /etc/vfstab ファイルのルート (/) エントリでは有効です。

/etc/vfstab のフィールドエントリの詳細は、表 28-4 を参照してください。次のことを確認します。

- 各フィールドを空白 (空白文字またはタブ) で区切る。
- フィールドで値を指定しない場合はダッシュ (-) を入力する。

3. 変更結果を保存します。

例 — /etc/vfstab ファイルにエントリを追加する

次の例では、デフォルトのマウントオプション (読み取り/書き込み) を使用して、ディスクスライス /dev/dsk/c0t3d0s7 を UFS ファイルとして、マウントポイントディレクトリ /files1 にマウントします。また、「device to fsck」として raw キャラクタ型デバイス /dev/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 | - |
| / | - | /tmp/ | | | | |
| newroot | lofs | - | yes | - | | |

次の例では、システム pluto 上のディレクトリ /export/man を、NFS ファイルシステムとしてマウントポイント /usr/man にマウントします。ファイルシステムが NFS であるため、「device to fsck」や「fsck pass」は指定されません。この例では、「mount options」は ro (読み取り専用) と soft になっています。

信頼性を高めるために、読み取り/書き込みの NFS ファイルシステムでは `hard` マウントオプションを指定します。

```
#device      device      mount  FS      fsck  mount  mount
#to mount    to fsck     point  type    pass  at boot options
pluto:/export/man -          /usr/man nfs    -      yes    ro,soft
```

次の例では、ルート (/) ファイルシステムをループバックマウントポイント `/tmp/newroot` にマウントします。「`mount at boot`」には `yes` を指定し、「`device to fsck`」と「`fsck pass`」の番号は指定しません。LOFS ファイルシステムをマウントするときは、LOFS を構成するのに使用されるファイルシステムを先にマウントし、その後で LOFS をマウントします。

```
#device      device      mount  FS      fsck  mount  mount
#to mount    to fsck     point  type    pass  at boot options
#
/
newroot lofs -          yes    -
```

▼ /etc/vfstab ファイル内に列挙されている複数のファイルシステムをマウントする方法

1. スーパーユーザーになります。

また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

2. /etc/vfstab ファイル内に列挙されたファイルシステムをマウントします。

```
# mountall [-l | -r] [-F fstype]
```

オプションを指定しなければ、`/etc/vfstab` ファイル内で「`mount at boot`」フィールドに `yes` を指定したすべてのファイルシステムがマウントされます。

- l /etc/vfstab ファイル内で「mount at boot」フィールドに **yes** を指定したすべてのローカルファイルシステムがマウントされる
- r /etc/vfstab ファイル内で「mount at boot」フィールドに **yes** を指定したすべてのリモートファイルシステムがマウントされる
- F *fstype* /etc/vfstab ファイル内で「mount at boot」フィールドに **yes** を指定した *fstype* タイプのすべてのファイルシステムがマウントされる

マウントが実行される前に、「device to fsck」エントリがあるすべてのファイルシステムがチェックされ、必要であれば修正されます。

例 — /etc/vfstab ファイル内に列挙されている複数のファイルシステムをマウントする

次の例は、`mountall` コマンドを実行したときに、すでにファイルシステムがマウントされている場合に表示されるメッセージを示します。

```
# mountall
mount: /tmp already mounted
nfs mount: mount: /usr/openwin: Device busy
nfs mount: mount: /usr/man: Device busy
```

次の例では、`/etc/vfstab` ファイル内に列挙されているすべてのローカルシステムをマウントします。

```
# mountall -l
# mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid/largefiles on ...
/usr on /dev/dsk/c0t3d0s6 read/write/setuid/largefiles on ...
/proc on /proc read/write/setuid on Wed May 27 09:11:50 1998
/dev/fd on fd read/write/setuid on Wed May 27 09:11:50 1998
/tmp on swap read/write on Wed May 27 09:11:50 1998
```

次の例では、`/etc/vfstab` ファイル内に列挙されているすべてのリモートファイルシステムをマウントします。


```
# mountall -r
# mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid/largefiles on ...
/usr on /dev/dsk/c0t3d0s6 read/write/setuid/largefiles on ...
/proc on /proc read/write/setuid on Mon Jun  8 10:28:31 1998
/dev/fd on fd read/write/setuid on Mon Jun  8 10:28:31 1998
/tmp on swap read/write on Mon Jun  8 10:28:31 1998
/nfs/mars.dist on mars:/usr/dist intr/remote on ...
/nfs/mars.mail on mars:/var/mail intr/noac/remote on ...
```

▼ /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する方法

1. スーパーユーザーになります。
また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。
2. /etc/vfstab ファイル内に列挙されているファイルシステムをマウントします。

```
# mount mount-point
```

mount-point /etc/vfstab ファイル内の「mount point」または「device to mount」フィールド内のエントリ。通常は、マウントポイントを指定する方が簡単です。

例 — /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する

次の例では、/etc/vfstab ファイル内に列挙されているファイルシステム /usr/openwin をマウントします。

```
# mount /usr/openwin
```

mount コマンドでファイルシステムをマウントする

▼ UFS ファイルシステムをマウントする方法

1. スーパーユーザーになります。

また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

2. `mount (1M)` コマンドにより、**UFS** ファイルシステムをマウントします。

```
# mount [-o mount-options] /dev/dsk/device-name mount-point
```

| | |
|-----------------------------------|--|
| <code>-o mount-options</code> | UFS ファイルシステムのマウントに使用できるマウントオプションを指定する。汎用マウントオプションについては、表 28-3を参照。すべてのオプションについては、 <code>mount_ufs (1M)</code> のマニュアルページを参照。 |
| <code>/dev/dsk/device-name</code> | ファイルシステムを保持するディスクスライス用のディスクデバイス名 (<code>/dev/dsk/c0t3d0s7</code> など)。ディスクデバイス名を調べる方法については、360ページの「ディスクスライス情報を表示する方法」を参照。 |
| <code>mount-point</code> | ファイルシステムをマウントするディレクトリ |

例 — UFS ファイルシステムをマウントする

次の例では、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントします。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

例 —ログを有効にして、UFS ファイルシステムをマウントする

UFS ロギングによってファイルシステムの整合性が保たれるために、システムの起動時間が大幅に短縮されます。次の例は、ログを有効にして、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントします。

```
# mount -o logging /dev/dsk/c0t3d0s7 /files1
```

▼ 大規模ファイルを持たない UFS ファイルシステムを再マウントする方法

ファイルシステムをマウントするときには、`largefiles` オプションがデフォルトで選択されるため、2G バイトを超えるファイルを作成できます。大規模ファイルを作成した後で、`nolargefiles` オプションを指定してファイルシステムを再マウントするには、大規模ファイルをすべて削除して、`fsck` を実行して状態を「`nolargefiles`」にリセットしなければなりません。

以下の手順では、ファイルシステム用のエントリが `/etc/vfstab` ファイルにあるものとします。

1. スーパーユーザーになります。
2. ファイルシステム内に大規模ファイルが存在しないことを確認します。

```
# cd mount-point  
# find . -xdev -size +2000000 -exec ls -l {} \;
```

mount-point

大規模ファイルがあるかどうかをチェックするファイルシステムのマウントポイントを指定する。

大規模ファイルが当該ファイルシステム内に存在する場合は、そのファイルを削除するか、他のファイルシステムに移動します。

3. ファイルシステムのマウントを解除します。

```
# umount mount-point
```

4. ファイルシステムの状態をリセットします。

```
# fsck mount-point
```

5. `nolargefiles` オプションを指定してファイルシステムを再マウントします。

```
# mount -o nolargefiles mount-point
```

例 — 大規模ファイルを持たないファイルシステムをマウントする

次の例は、`/datab` ファイルシステムをチェックし、`nolargefiles` オプションを指定して再マウントしています。

```
# cd /datab
# find . -xdev -size +2000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

▼ NFS ファイルシステムをマウントする方法

1. スーパーユーザーになります。

また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

2. 資源 (ファイルまたはディレクトリ) がサーバーから使用可能かどうかを確認します。

NFS ファイルシステムをマウントするには、`share (1M)` コマンドを使用し、サーバー上の資源を使用可能にしておかなければなりません。資源を共有する方法については、『*NFS の管理*』を参照してください。

3. `mount (1M)` コマンドにより、**NFS** ファイルシステムをマウントします。

```
# mount -F nfs [-o mount-options] server:/directory mount-point
```

| | |
|--------------------------------|---|
| <code>-o mount-options</code> | NFS ファイルシステムのマウントに使用できるマウントオプションを指定する。汎用マウントオプションのリストについては、表 28-3を参照。オプションについては、 <code>mount_nfs(1M)</code> のマニュアルページを参照。 |
| <code>server:/directory</code> | 共有する資源を持つサーバーのホスト名と、マウントするファイルまたはディレクトリへのパスを指定する。 |
| <code>mount-point</code> | ファイルシステムをマウントするディレクトリを指定する。 |

例 — NFS ファイルシステムをマウントする

次の例は、サーバー `pluto` の `/export/packages` ディレクトリを `/mnt` にマウントしています。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: System V (S5FS) ファイルシステムをマウントする方法

1. スーパーユーザーになります。
また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。
2. `mount(1M)` コマンドにより、**S5FS** ファイルシステムをマウントします。

```
# mount -F s5fs [-o mount-options] /dev/dsk/device_name mount-point
```

| | |
|-----------------------------------|--|
| <code>-o mount-options</code> | S5FS ファイルシステムのマウントに使用できるマウントオプションを指定する。汎用マウントオプションについては、表 28-3 を参照。オプションについては、 <code>mount_s5fs(1M)</code> のマニュアルページを参照。 |
| <code>/dev/dsk/device-name</code> | ファイルシステムが存在するディスクスライスのデバイス名 (<code>/dev/dsk/c0t3d0s7</code> など)。ディスクデバイス名を調べる方法については、360ページの「ディスクスライス情報を表示する方法」を参照。 |
| <code>mount-point</code> | ファイルシステムをマウントするディレクトリを指定する。 |

例 — S5FS ファイルシステムをマウントする

次の例では、`/dev/dsk/c0t3d0s7` を `/files1` ディレクトリにマウントします。

```
# mount -F s5fs /dev/dsk/c0t3d0s7 /files1
```

▼ ハードディスク上の PCFS (DOS) ファイルシステムをマウントする方法

次の手順で、PCFS (DOS) ファイルシステムをハードディスクからマウントします。

1. スーパーユーザーになります。

また、ファイルシステムをマウントするには、ローカルのシステムにマウントポイントが必要です。マウントポイントとは、マウントされるファイルシステムが接続されるディレクトリのことです。

2. `mount(1M)` コマンドにより、PCFS ファイルシステムをマウントします。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive mount-point
```

| | |
|-----------------------------------|--|
| <code>-o rw ro</code> | PCFS ファイルシステムを読み取り/書き込みまたは読み取り専用でマウントできるように指定する。このオプションを指定しなければ、デフォルトは読み取り/書き込みになる |
| <code>/dev/dsk/device-name</code> | ディスク全体のデバイス名 (<code>/dev/dsk/c0t0d0p0</code> など) |
| <code>logical-drive</code> | DOS の論理ドライブ名 (c から z まで) または 1 から 24 までのドライブ番号を指定する。ドライブ C はドライブ 1 に相当し、ディスク上の基本 DOS スライスを表す。他のすべてのドライブ名やドライブ番号は、拡張 DOS スライス内の DOS 論理ドライブを表す |
| <code>mount-point</code> | ファイルシステムをマウントするディレクトリ |

「device-name」と「logical-drive」は、コロンで区切らなければならないので注意してください。

例 — PCFS (DOS) ファイルシステムをハードディスクからマウントする

次の例では、基本 DOS スライス内のドライブが `/pcfs/c` ディレクトリにマウントされます。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例では、ディスク上の拡張 DOS スライス内の最初の論理ドライブが、`/pcfs/d` に読み取り専用としてマウントされます。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /pcfs/d
```

ファイルシステムのマウント解除

ファイルシステムをマウント解除すると、ファイルシステムのマウントポイントから削除され、そのエントリが `/etc/mnttab` ファイルから削除されます。マウントされているファイルシステム上では、一部のファイルシステム管理作業を実行できません。次の場合には、ファイルシステムをマウント解除する必要があります。

- ファイルシステムが不要になった場合、またはより新しいソフトウェアが入ったファイルシステムに交換された場合
- `fsck` コマンドを使用してファイルシステムを検査し、修復する場合 (`fsck` コマンドについての詳細は、第 31 章を参照)

ファイルシステムの完全バックアップを実行する前に、マウント解除しておく
とよいでしょう。バックアップの実行についての詳細は、第 34 章を参照してく
ださい。

注 - 各ファイルシステムは、ファイルシステムのシャットダウン手続きの一部とし
て自動的にマウント解除されます。

前提条件

ファイルシステムをマウント解除する場合の前提条件は次のとおりです。

- スーパーユーザーでなければならない
- ファイルシステムがマウント解除が可能な状態でなければならない。使用中の
ファイルシステムはマウント解除できません。ファイルシステムは、ユーザーが
そのディレクトリ内に入っているときや、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には使用中と見なされます。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にアクセス中のすべての
プロセスを表示し、必要に応じて終了させる。詳細は、479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。

マウント解除したファイルシステムを検査する

ファイルシステムをマウント解除したことを確認するには、`mount` コマンドからの
出力を調べます。この手順については、465ページの「どのファイルシステムがマウ
ントされているかを調べる方法」を参照してください。

▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法

1. スーパーユーザーになります。
2. どのプロセスを終了させるのかがわかるように、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ます |
| mount-point | プロセスを終了させたいファイルシステム名 |

3. ファイルシステムのすべてのプロセスを終了させます。

注 - ユーザーのプロセスを終了させるときには、必ず事前に警告してください。

```
# fuser -c -k mount-point
```

ファイルシステムを使用中のプロセスに SIGKILL が送信されます。

4. ファイルシステムを使用中のプロセスがないことを確認します。

```
# fuser -c mount-point
```

例 — あるファイルシステムを使用中のすべてのプロセスを終了させる

次の例では、/export/home ファイルシステムを使用中のプロセス 4006 を終了させます。

```
# fuser -c /export/home
/export/home: 4006c
# fuser -c -k /export/home
/export/home: 4006c
# fuser -c /export/home
/export/home:
```

▼ ファイルシステムをマウント解除する方法

次の手順で、ファイルシステム (/ または /usr を除く) をマウント解除します。

注 - ルート (/) と /usr の UFS ファイルシステムは特殊な場合です。システムが機能するにはルートが必要なので、ルート (/) のファイルシステムはシャットダウン中でなければマウント解除できません。

1. 478ページの「前提条件」の前提条件を満たしているかどうかを確認します。
2. ファイルシステムをマウント解除します。

```
# umount mount-point
```

mount-point

マウント解除したいファイルシステム名。ここでは、ファイルシステムがマウントされているディレクトリ名、ファイルシステムのデバイス名パス、NFS ファイルシステムの資源、LOFS ファイルシステムのループバックディレクトリのいずれかを指定できる

例 — ファイルシステムをマウント解除する

次の例では、ローカルのホームディレクトリからファイルシステムがマウント解除されます。

```
# umount /export/home
```

次の例ではローカルディスクの7番目のスライス上のファイルシステムのマウントが解除されます。

```
# umount /dev/dsk/c0t0d0s7
```

▼ /etc/vfstab ファイルに列挙されているファイルシステムをマウント解除する方法

次の手順で、/、/proc、/var、/usr のファイルシステムを除き、/etc/vfstab ファイルに列挙されているすべてのファイルシステムをマウント解除します。

1. 478ページの「前提条件」の前提条件を満たしているかどうかを確認します。
2. /etc/vfstab ファイルに列挙されているすべてのファイルシステムをマウント解除します。

```
# umountall
```

可能なファイルシステムがすべてマウント解除されます。使用中のファイルシステムはマウント解除されません。

3. 使用中だったためにマウント解除されなかったファイルシステムの場合は、479ページの「ファイルシステムを使用中のすべてのプロセスを終了させる方法」で説明した手順に従って、マウント解除できるようにします。
4. 必要に応じて、すべてのファイルシステムがマウント解除されるまで手順 2 を繰り返します。

キャッシュファイルシステムの手順

キャッシュファイルシステム (CacheFS) は、サーバーとネットワークの負荷を軽減して NFS サーバーの性能とスケーラビリティを改善する汎用ファイルシステムキャッシュ機構です。CacheFS は階層化ファイルシステムとして設計されていて、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS はサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) のような低速リンク上でクライアントの性能を改善します。

この章の内容は次のとおりです。

- 484ページの「CacheFS の機能」
- 485ページの「キャッシュファイルシステムの設定」
- 486ページの「キャッシュを作成する方法」
- 487ページの「mount を使用してキャッシュにマウントするファイルシステムを指定する方法」
- 490ページの「/etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法」
- 491ページの「AutoFS を使用してファイルシステムをキャッシュにマウントする方法」
- 493ページの「キャッシュ内でファイルシステムを変更する方法」
- 494ページの「キャッシュされたファイルシステムに関する情報を表示する方法」
- 495ページの「必要に応じて整合性チェックを指定する方法」
- 496ページの「キャッシュされたファイルシステムを削除する方法」

- 497ページの「キャッシュされたファイルシステムの完全性をチェックする方法」
- 512ページの「CacheFS の統計情報」
- 513ページの「CacheFS 統計情報を設定して表示するための前提条件」
- 513ページの「CacheFS 統計情報の設定」
- 514ページの「ロギングプロセスを設定する方法」
- 516ページの「キャッシュサイズの表示」
- 516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」
- 517ページの「統計情報の表示」
- 518ページの「キャッシュ統計情報を表示する方法」
- 519ページの「キャッシュの構造と動作」
- 520ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」
- 520ページの「必要に応じて行う整合性チェック」

CacheFS の機能

キャッシュにマウントするように指定したファイルシステムに、ユーザーがネットワークを経由しなくてもローカルにアクセスできるように、`cfsadmin(1M)` コマンドを使用してキャッシュをクライアント上で作成します。図 29-1 は、CacheFS の使用に関連する構成要素の関係を示しています。

バックファイルシステムとは、キャッシュにマウントするように指定するファイルシステムです。通常、これは NFS または HSFS (High Sierra File System) ファイルシステムです。ユーザーがバックファイルシステムの一部であるファイルにアクセスしようとする、そのファイルはキャッシュに書き込まれます。したがって、ユーザーがファイルにアクセスするまで、キャッシュは空になっています。ユーザーにとって、ファイルに初めてアクセスするときは低速に思われますが、同じファイルを使用すると高速になります。

注 - 共有可能なファイルシステムしかマウントできません。`share(1M)` コマンドを参照してください。

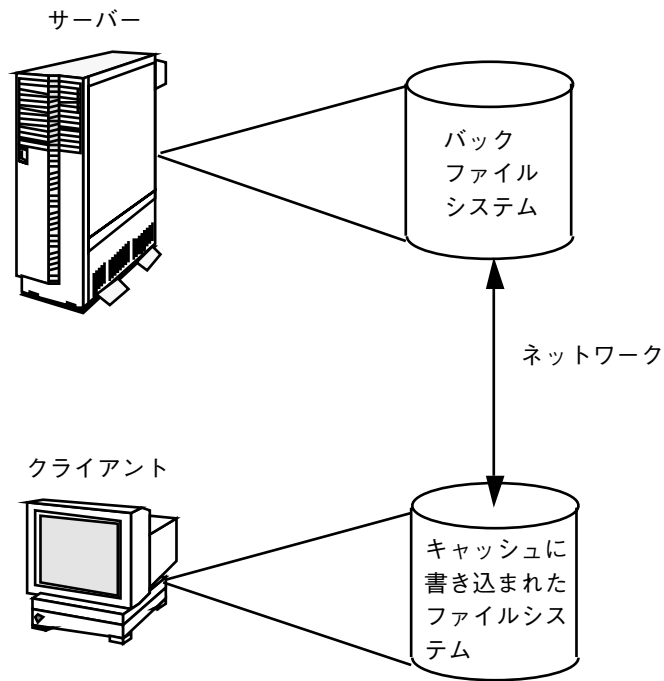


図 29-1 CacheFS の機能

キャッシュファイルシステムの設定

表 29-1 作業マップ：キャッシュファイルシステムの設定

| 作業 | 説明 | 手順の説明 |
|--------------------------|---------------------------------------|---|
| 1. キャッシュの作成 | cfsadmin コマンドを使用してキャッシュを作成する。 | 486ページの「キャッシュを作成する方法」 |
| 2. ファイルシステムをキャッシュにマウントする | mount コマンドを使用してファイルシステムをキャッシュにマウントする。 | 487ページの「mount を使用してキャッシュにマウントするファイルシステムを指定する方法」 |

表 29-1 作業マップ: キャッシュファイルシステムの設定 続く

| 作業 | 説明 | 手順の説明 |
|----|--|--|
| | <p>/etc/vfstab ファイルを編集してファイルシステムをキャッシュに書き込む。</p> <p>AutoFS を使用してファイルシステムをキャッシュに書き込む。</p> | <p>490ページの「/etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法」</p> <p>491ページの「AutoFS を使用してファイルシステムをキャッシュにマウントする方法」</p> |

▼ キャッシュを作成する方法

1. スーパーユーザーになります。
2. `cfsadmin(1M)` コマンドの `-c` オプションを使用してキャッシュを作成します。

```
# cfsadmin -c cache-directory
```

cache-directory

キャッシュが作成されるディレクトリ名。詳細は、`cfsadmin(1M)` のマニュアルページを参照

注 - キャッシュを作成し終わったら、キャッシュディレクトリ内では操作しないでください。何か実行すると、CacheFS ソフトウェア内で矛盾が生じます。

例 — キャッシュを作成する

次の例では、デフォルトのキャッシュパラメータ値を使用して、`/local/mycache` ディレクトリ内にキャッシュを作成します。

```
# mkdir /local
# cfsadmin -c /local/mycache
```


キャッシュにマウントするファイルシステムの指定

指定したファイルシステム内のファイルにユーザーがローカルにアクセスできるように、キャッシュにマウントされるファイルシステムを指定します。実際には、各ファイルは、ユーザーがアクセスするまではキャッシュに書き込まれません。

キャッシュにマウントされるファイルシステムを指定するには、次の3つの方法があります。

- `mount (1M)` 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必要があります。
- `/etc/vfstab` ファイルを編集する方法 – 一度編集するだけですむ。`/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 – 一度実行するだけですむ。`AutoFS` のマップは、システムをリブートした後も変更されずに残る。

いずれかの方法を選択してファイルシステムをマウントしてください。

注 - `CacheFS` では、ルート (`/`) と `/usr` のファイルシステムをキャッシュに書き込むことはできません。ルート (`/`) と `/usr` のファイルシステムをキャッシュに書き込むには、`Solstice AutoClient` 製品を購入しなければなりません。`AutoClient` 製品の詳細は、『*Solstice AutoClient 2.1 管理者ガイド*』を参照してください。

▼ `mount` を使用してキャッシュにマウントするファイルシステムを指定する方法

1. スーパーユーザーになります。
2. マウントポイントを作成します。

マウントポイントを作成すると、ユーザーはそのマウントポイントで指定されたファイルシステムにアクセスできます。マウントポイントはどこからでも作成できます。次の手順のように、マウントコマンドで使用する `CacheFS` オプションによって、作成するマウントポイントを、指定したキャッシュディレクトリ内のキャッシュに書き込むように決めます。
3. `mount` コマンドを使用して、ファイルシステムをキャッシュにマウントします。

```
# mount -F cachefs -o backfstype=fstype,cachedir=cache-directory[,options]
back-filesystem mount-point
```

| | |
|------------------------|---|
| <i>fstype</i> | nfs など、バックファイルシステムのタイプ |
| <i>cache-directory</i> | キャッシュがあるディレクトリ名。これは、486ページの「キャッシュを作成する方法」でキャッシュを作成するときの指定と同じ |
| <i>options</i> | ファイルシステムをキャッシュにマウントするときに追加できる他のマウントオプションを指定する。CacheFS マウントオプションの詳細は、mount_cachefs(1M) のマニュアルページを参照 |
| <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れば、merlin: /usr/openwin など、ファイルシステムのマウント元となるサーバーのホスト名と、キャッシュにマウントするファイルシステム名(コロンで区切る)を指定しなければならない |
| <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するには、次のように `cachefsstat(1M)` コマンドを使用します。

```
# cachefsstat mount-point
```

たとえば、次のようになります。

```
# cachefsstat /docs
/docs
      cache hit rate:   100% (0 hits, 0 misses)
      consistency checks: 1 (1 pass, 0 fail)
      modifies:        0
      garbage collection: 0
```

マウントポイントは、作成したキャッシュファイルシステムです。cachefsstat コマンドの詳細は、512ページの「CacheFS の統計情報」を参照してください。ファイルシステムがキャッシュにマウントされなかった場合は、次のようなエラーメッセージが表示されます。

```
# cachefsstat mount-point
cachefsstat: mount-point: not a cachefs mountpoint
```

例 — mount を使用してキャッシュにマウントするファイルシステムを指定する

次の例では、マウントポイント /docs を作成し、NFS ファイルシステム merlin:/docs を、キャッシュされたファイルシステム /docs としてキャッシュ /local/mycache にマウントします。

```
# mkdir /docs
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

次の例では、CD-ROM (HSFS ファイルシステム) をキャッシュされたファイルシステム /docs として使用できるようにします。CD-ROM には書き込めないで、引数 ro を指定してキャッシュされたファイルシステムを読み込み専用にします。CD-ROM は挿入時にボリューム管理によって自動的にマウントされるので、引数 backpath を指定しなければなりません。マウントポイントは /cdrom ディレクトリにあり、CD-ROM 名によって決まります。マウントする特殊デバイスは、backpath オプションに指定した値と同じです。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/local/mycache,ro backpath=/cdrom/cdrom_name /cdrom/cdrom_name /docs
```

次の例では、demandconst オプションを使用して、NFS キャッシュファイルシステム /docs の整合性チェックを指定します。/docs のバックファイルシステムは merlin:/docs です。詳細は、520ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」を参照してください。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ /etc/vfstab ファイルを編集してファイルシステムをキャッシュにマウントする方法

1. スーパーユーザーになります。
2. エディタを使用して、マウントするファイルシステムを /etc/vfstab ファイル内で指定します。

```
#device      device          mount FS    fsck mount  mount
#to mount    to fsck          point type pass  at boot options
#
/dev/dsk/devicename /dev/rdisk/devicename /mount-point cachefs 2    yes  -
```

最終行は新しいエントリを表します。

3. 次のように mount コマンドを使用してキャッシュされたファイルシステムをマウントするか、リブートします。

```
# mount /mount-point
```

例 — /etc/vfstab ファイルを編集してキャッシュにファイルシステムをマウントする

次の例は、キャッシュファイルシステム用の /etc/vfstab エントリを示します。

```
#device      device          mount FS    fsck mount  mount
#to mount    to fsck          point type pass  at boot options
#
/dev/dsk/c0t1d0s0 /dev/rdisk/c0t1d0s0 /usr/local cachefs 2    yes  -
```

次のように入力して、/usr/local ディレクトリをキャッシュディレクトリにマウントします。

```
# mount /usr/local
```

▼ AutoFS を使用してファイルシステムをキャッシュにマウントする方法

自動マウントマップ内で `-fstype=cachefs` マウントオプションを指定して、AutoFS によりキャッシュにファイルシステムをマウントします。CacheFS のマウントオプション (`backfstype` や `cachedir` など) も、自動マウントマップ内で指定します。自動マウントマップの詳細は、`automount(1M)` のマニュアルページや、『NFS の管理』も参照してください。

1. スーパーユーザーになります。

2. エディタを使用して、次の 1 行を `auto_direct` マップに追加します。

```
/mount-point -fstype=cachefs,cachedir=/directory,backfstype=nfs server:/file-system
```

3. エディタを使用して、次の 1 行を `auto_master` マップに追加します。

```
/-
```

`/-` エントリによって、`auto_direct` マップをチェックするように指示します。

4. システムをリブートします。

5. エントリが正しく作成されたか確認するには、次のようにキャッシュにマウントしたファイルシステムにカレントディレクトリを変更して内容を表示します。

```
# cd filesystem
# ls
```

AutoFS とマップを編集する方法の詳細は、『NFS の管理』の AutoFS に関する章を参照してください。

例 — AutoFS を使用してファイルシステムをキャッシュにマウントする

次の auto_master エントリは、キャッシュファイルシステムを /docs ディレクトリに自動的にマウントします。

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

キャッシュされたファイルシステムの管理

表 29-2 作業マップ：キャッシュされたファイルシステムの管理

| 作業 | 説明 | 手順の説明 |
|--------------------|--|--|
| 1. キャッシュの変更 | キャッシュの動作を変更する。 | 493ページの「キャッシュ内でファイルシステムを変更する方法」 |
| 2. キャッシュ情報の表示 | cfsadmin コマンドを使用してキャッシュされたファイルシステムに関する情報を表示する。 | 494ページの「キャッシュされたファイルシステムに関する情報を表示する方法」 |
| 3. 整合性チェックの実行 | cfsadmin 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。 | 495ページの「必要に応じて整合性チェックを指定する方法」 |
| 4. キャッシュの削除 | umount コマンドと cfsadmin コマンドを使用して、キャッシュされたファイルシステムを削除する。 | 496ページの「キャッシュされたファイルシステムを削除する方法」 |
| 5. ファイルシステムの完全性の検査 | fsck_cachefs コマンドを使用して、キャッシュされたファイルシステムの完全性をチェックする。 | 497ページの「キャッシュされたファイルシステムの完全性をチェックする方法」 |

キャッシュの管理

キャッシュを設定し終わったら、次の管理作業を実行できます。

- キャッシュ内のファイルシステムを変更する (キャッシュをマウント解除し、削除し、作成し直し、再びマウントする)
- キャッシュ情報を表示する
- キャッシュの整合性をチェックする
- キャッシュからファイルシステムを削除する
- キャッシュされたファイルシステムの完全性をチェックする

注 - /etc/vfstab ファイルを使用してファイルシステムをマウントしている場合は、このファイル内でファイルシステムオプションを編集してキャッシュを変更します。AutoFS を使用している場合は、AutoFS マップ内でファイルシステムオプションを編集してキャッシュを変更します。

▼ キャッシュ内でファイルシステムを変更する方法

ファイルシステムの特定のオプションを変更する方法については、第 28 章を参照してください。キャッシュ内でファイルシステムを変更する場合は、キャッシュを削除してから作成し直す必要があります。また、ファイルシステムの共有方法とアクセス方法によっては、マシンをシングルユーザーモードでリブートしなければならない場合があります。

次の例は、この手順を示しています。

例 — キャッシュ内でファイルシステムを変更する

次の例では、キャッシュが削除されてから再び作成され、ファイルシステム /docs に指定された demandconst オプションを使用して再びマウントされます。この例は、シングルユーザーモードでリブートするなどの手順を示しています。必要であれば他のコマンドを使用して、この例の作業を実行することもできます。

```
$ su
password:

# shutdown -g30 -y
```

(続く)

```

.
.
.
Type Cntrl-d to proceed with normal startup,
(or give root password for system maintenance):
# enter password:
.
.
.
Here is where you may be prompted from system to run fsck on the
file system where the cache is located.

# fsck /local
# mount /local
# cfsadmin -d all /local/mycache
# cfsadmin -c /local/mycache
# init 6
.
.
.
console login:
password:
# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1, demandconst merlin:/docs /docs
#

```

ファイルシステムをキャッシュに正常に戻せなかった場合は、次のようなエラーメッセージが表示されます。

```
cachefsstat: /doc: not a cachefs mount point
```

▼ キャッシュされたファイルシステムに関する情報を表示する方法

1. スーパーユーザーになります。
2. 指定したキャッシュにマウントされたすべてのファイルシステムに関する情報を表示します。

```
# cfsadmin -l cache-directory
```

cache-directory はキャッシュがあるディレクトリ名です。

例 — キャッシュされたファイルシステムに関する情報を表示する

次の例は、キャッシュディレクトリ `/local/mycache` に関する情報を示しています。この例では、ファイルシステム `/docs` が `/local/mycache` にマウントされています。最終行にはキャッシュされたファイルシステムが表示されます。

```
# cfsadmin -l /local/mycache
cfsadmin: list cache FS information
maxblocks      90%
minblocks      0%
threshblocks   85%
maxfiles       90%
minfiles       0%
threshfiles    85%
maxfilesize    3MB
merlin:_docs:_docs
#
```

▼ 必要に応じて整合性チェックを指定する方法

1. スーパーユーザーになります。
2. 次のように `mount (1M)` コマンドの `demandconst` オプションを指定して、ファイルシステムをキャッシュにマウントします。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/directory,demandconst
server:/file-system /mount-point
```

3. キャッシュされた特定のファイルシステム上で整合性チェックを有効にするには、次のように `-s` オプションを指定して `cfsadmin (1M)` コマンドを使用します。

```
# cfsadmin -s /mount-point
```

整合性チェックについての詳細は、520ページの「キャッシュされたファイルシステムとバックファイルシステムとの整合性チェック」を参照してください。

▼ キャッシュされたファイルシステムを削除する方法

1. スーパーユーザーになります。
2. キャッシュ内のすべてのファイルシステムをマウント解除します。

```
# umount mount-point
```

mount-point には、削除したい、キャッシュされたファイルシステムを指定します。

3. 次のように、`cfsadmin -l` の出力からキャッシュ ID を判別します。

```
# cfsadmin -l cache-directory
cfsadmin: list cache FS information
maxblocks      90%
minblocks      0%
threshblocks   85%
maxfiles       90%
minfiles       0%
threshfiles    85%
maxfilesize    3MB
cache-id
#
```

4. キャッシュされたファイルシステムを指定したキャッシュから削除します。

```
# cfsadmin -d cache-id cache-directory
```

cache-id

キャッシュされた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。詳細は、494ページの「キャッシュされたファイルシステムに関する情報を表示する方法」を参照。*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すべてのファイルシステムを削除できる

cache-directory

キャッシュがあるディレクトリ

5. キャッシュされたファイルシステムが削除されたことを確認します。

削除したファイルシステムのキャッシュ ID は、次のコマンドの出力に含まれなくなりません。コマンド出力で指定するフィールドの詳細は、`cfsadmin(1M)` のマニュアルページを参照してください。

```
# cfsadmin -l cache-directory
cfsadmin: list cache FS information
maxblocks      90%
minblocks      0%
threshblocks   85%
maxfiles       90%
minfiles       0%
threshfiles    85%
maxfilesize    3MB
#
```

例 — キャッシュされたファイルシステムを削除する

次の例では、キャッシュされたファイルシステムをマウント解除して、キャッシュから削除します。

```
# umount /usr/openwin
# cfsadmin -d merlin:docs /docs /local/mycache
```

次の例では、`/local/mycache` キャッシュ内のキャッシュされたファイルシステムをすべて削除します。

```
# cfsadmin -d all /local/mycache
```

▼ キャッシュされたファイルシステムの完全性をチェックする方法

`fsck(1M)` コマンドを使用して、キャッシュされたファイルシステムが完全かどうかを確認します。何も操作しなくても、`fsck` の CacheFS バージョンによって問題が自動的に解決されます。キャッシュされたファイルシステムの場合は、`fsck` を手作業で実行する必要はありません。`fsck` は、ブート時またはファイルシステムのマウント時に自動的に実行されます。完全性を手作業で確認したい場合は、次の手順を使用できます。

詳細は、`fsck_cachefs(1M)` を参照してください。

1. スーパーユーザーになります。
2. 指定されたキャッシュ内でキャッシュされたファイルシステムをチェックします。

```
# fsck -F cachefs [-m] [-o noclean] cache-directory
```

| | |
|-----------------|---------------------------------------|
| -m | fsck にキャッシュされたファイルシステムをチェックさせるが、修復しない |
| -o noclean | キャッシュされたファイルシステム上でチェックのみを実行させるが、修復しない |
| cache-directory | キャッシュがあるディレクトリ名 |

例 — キャッシュされたファイルシステムの完全性をチェックする

次の例では、/local/mycache キャッシュの一部になっているキャッシュされたファイルシステムをチェックします。

```
# fsck -F cachefs /local/mycache  
#
```

cachefspack によるキャッシュファイルシステムの管理

一般的な使い方でも、CacheFS は、ユーザーが要求しなくても自動的に適切な動作をします。最近使用されたファイルがキャッシュされます。パッキング機能を使用するとキャッシュ内で、特定のファイルまたはディレクトリを常に最新の状態で保持できるので、より積極的にキャッシュを管理できます。

パッキングリスト

`cachefspack` コマンドの機能の1つに、パッキングリストをパックするという機能があります。これによって、キャッシュにパックする各ファイルをそれぞれ指定するという手間が省けます。

パッキングリストには、キャッシュに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。

▼ パッキングリストを作成する方法

パッキングリストを作成するには、`vi` などのエディタを使用して、ファイルを開きます。パッキングリストファイルの書式は、`filesync` コマンドで使用する書式と同じです。`filesync` コマンドの詳細は、`filesync(1)` のマニュアルページを参照してください。

例 — パッキングリストを作成する

次の例は、パッキングリストファイルの内容を示しています。

```
BASE /home/ignatz
LIST plans
LIST docs
IGNORE *.ps
```

- BASE 文は、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。

▼ パッキングリストに指定してファイルをキャッシュにパックする方法

パッキングリストを使用してファイルをパックするには、次に示すように、`cachefspack` コマンドに `-f` オプションを指定します。

```
$ cachefspack -f packing-list
```


例 — 正規表現として扱うファイルをパッキングリストに指定する

次の例は、パッキングリスト `list.pkg` をキャッシュにパックするように指定する方法を示しています。`list.pkg` には、正規表現を定義する `LIST` コマンドが入っています。

```
$ cachefspack -rf list.pkg
```

このコマンドは、`list.pkg` ファイルをキャッシュにパッキングして、`LIST` コマンドに定義されたファイル名を (文字どおりのファイル名ではなく) 正規表現として扱います。

▼ 共有ディレクトリのファイルをパックする方法

1. 共有ディレクトリにある、ユーザーが所有するファイルだけをパックするには、次のように `LIST` コマンドをパッキングリストに定義します。

```
LIST !find . -user your_user_name -print
```

2. `cachefspack -sf` コマンドを使用して、パッキングリストをキャッシュにパックします。

```
$ cachefspack -sf packing_list
```

`-s` `find` コマンドの出力をパッキングリストに適するように調整する。

`-f` パッキングリストを読み取ることを指定する。

`filename` 読み取るパッキングリスト名を指定する。

注 - `-s` オプションは `-f` オプションと一っしょに使用しなければなりません。`-s` オプションのみを使用することはできません。

例 — 共有ディレクトリのファイルをバックする

次の例は、ユーザーが所有するベースディレクトリのファイルだけをバックするために、LIST コマンドをバックアップリストに定義する方法を示しています。

```
LIST !find . -user jones -print
```

次の例は、バックアップリストをバックするように指定する方法を示しています。

```
$ cachefspack -sf /projects/proj_1
```

ファイルのバックアップ解除

キャッシュからファイルを削除、つまりそのバックアップを解除する場合もあります。他よりも優先度が高い一部のファイルまたはディレクトリが存在するため、重要でないファイルのバックアップ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バックアップリストのバックアップを解除する方法

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には、cachefspack コマンドの `-u` オプションか `-U` オプションを使用します。

```
$ cachefspack -u filename | -U cache-directory
```

| | |
|-----------------------|--|
| <code>-u</code> | ファイルまたは複数のファイルのバックアップを解除することを指定する。このオプションを使用する場合は、ファイル名を指定しなければならない |
| <code>filename</code> | キャッシュからバックアップを解除するファイル名またはバックアップリスト名を指定する。cachefspack コマンドの詳細は、マニュアルページを参照 |
| <code>-U</code> | キャッシュ中のすべてのファイルのバックアップを解除することを指定する |

例 — キャッシュからファイルまたはパッキングリストのパッキングを解除する

次の例は、キャッシュからファイル `/usr/openwin/bin/xlogo` のパッキングを解除するように指定する方法を示しています。

```
$ cacheftpck -u /usr/openwin/bin/xlogo
```

次の例は、キャッシュから複数のファイルのパッキングを解除するように指定する方法を示しています。

```
$ cd /usr/openwin/bin  
$ cacheftpck -u xlogo xview xcolor
```

また、次のように、パッキングリスト (ファイルのディレクトリのパスを含むファイル) でもパッキングを解除できます。

```
$ cacheftpck -uf list.pkg
```

次の例は、キャッシュディレクトリ中のすべてのファイルのパッキングを解除するように `-U` オプションを使用する方法を示しています。

```
$ cacheftpck -U /local/mycache
```

ファイルシステムを1つもマウントしていないキャッシュのパッキングは解除できません。`-U` オプションに対して、マウント済みファイルシステムを持っていないキャッシュを指定した場合、次のような出力が表示されます。

```
$ cacheftpck -U /local/mycache  
cacheftpck: Could not unpack cache /local/mycache, no mounted  
filesystems in the cache.
```

パックされたファイルの情報の表示

パックされたファイルについての情報やパッキングの状態を表示することも可能です。

▼ パックされたファイルの情報を表示する方法

パックされたファイルの情報を表示するには、cachefspack コマンドの `-i` オプションを使用します。

```
$ cachefspack -i[v] cached-filename-or-directory
```

| | |
|---|-----------------------------|
| <code>-i</code> | パックされたファイルの情報を表示することを指定する |
| <code>-v</code> | 詳細表示オプション |
| <code>cached-filename-or-directory</code> | 情報を表示するファイル名またはディレクトリ名を指定する |

例

次の例は、ファイル `doc_file` がパックされていることを示しています。

```
$ cachefspack -i doc_file
cachefspack: file doc_file marked packed YES, packed YES
```

次の例は、ディレクトリ `/usr/openwin` に関する情報を示しています。このディレクトリには、サブディレクトリ `bin` があり、サブディレクトリ `bin` には `xterm`、`textedit`、`resize` の3つのファイルがあります。`xterm` と `resize` ファイルはパックするように指定されていますが、パックされていません。`textedit` ファイルは正常にパックされています。

```
$ cd /usr/openwin
$ cachefspack -i bin
.
.
.
cachefspack: file /bin/xterm marked packed YES, packed NO
cachefspack: file /bin/textedit marked packed YES,
packed YES
cachefspack: file /bin/resize marked packed YES,
packed NO
.
.
.
```

-v オプションを -i オプションと併用した場合、指定したファイルまたはディレクトリがキャッシュからフラッシュされたかどうかの追加情報も表示されます。たとえば、次のとおりです。

```
$ cd /usr/openwin
$ cachefspack -iv bin
.
.
.
cachefspack: file /bin/xterm marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/textedit marked packed YES,
packed YES, nocache NO
cachefspack: file /bin/resize marked packed YES,
packed NO
nocache NO
.
.
.
```

上記例の最後の行は、ディレクトリの内容がキャッシュからフラッシュされていないことを示しています。

cachefspack コマンドのヘルプの表示

cachefspack のすべてのオプションのヘルプの要約とその意味を表示するには、次のように -h オプションを使用します。

```
$ cachefspack -h
Must select 1 and only 1 of the following 5 options
-d Display selected filenames
-i Display selected filenames packing status
-p Pack selected filenames
-u Unpack selected filenames
-U Unpack all files in directory 'dir'

-f Specify input file containing rules
-h Print usage information
-r Interpret strings in LIST rules as regular expressions
-s Strip './' from the beginning of a pattern name
-v Verbose option
files - a list of filenames to be packed/unpacked
```

cachefspack エラー

cachefspack コマンドを使用すると、次のようなエラーメッセージが表示される場合があります。

```
cachefspack: pathname - ディレクトリをオープンできません。アクセス権がありません。
```

エラーが発生した理由

問題を解決する方法

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。

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はありません
```

エラーが発生した理由

問題を解決する方法

正しいファイルまたはディレクトリがありません。

入力ミスがないかチェックしてください。

```
cachefspack: pathname - ディレクトリをオープンできません。stale NFS file handle
```

エラーが発生した理由

問題を解決する方法

アクセスしたときに、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。

システム管理者に問い合わせてください。

```
cachefspack: pathname - ディレクトリをオープンできません。interrupted system call
```

エラーが発生した理由

問題を解決する方法

このコマンドの実行途中で Control-c キーを発行した可能性があります。

コマンドを再発行してください。

```
cachefspack: pathname - ディレクトリをオープンできません。I/O エラー
```

エラーが発生した理由

問題を解決する方法

ハードウェアの障害です。

ハードウェアの接続をチェックしてください。

```
cachefspack: error opening dir
```

エラーが発生した理由

問題を解決する方法

正しいファイルまたはディレクトリがありません。

入力ミスがないかチェックしてください。

ファイル書式中の BASE コマンドの後に指定したパスはファイルで、ディレクトリではない可能性があります。指定するパスはディレクトリでなければなりません。

ファイル書式中の BASE コマンドの後に指定したパスをチェックしてください。これが、ファイルではなく、ディレクトリであることを確認してください。

```
cachefspack: 共有オブジェクトを獲得できません
```

エラーが発生した理由

問題を解決する方法

実行可能ファイルが壊れている可能性があります。あるいは、そのフォーマットが認識できません。

解決方法はありません。

```
cachefspack: filename - ファイルを pack できません。アクセス権がありません
```

エラーが発生した理由

問題を解決する方法

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。

適切なアクセス権を入手してください。

```
cachefspack: filename - ファイルを pack できません。  
ファイルまたはディレクトリはありません
```

エラーが発生した理由

問題を解決する方法

正しいファイルまたはディレクトリがありません。

入力ミスがないかチェックしてください。

```
cachefspack: filename - ファイルを pack できません。stale NFS file handle
```

エラーが発生した理由

問題を解決する方法

アクセスしたときに、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。

システム管理者に問い合わせてください。

```
cachefspack: filename - ファイルを pack できません。interrupted system call
```

エラーが発生した理由

問題を解決する方法

このコマンドの実行途中で Control-c キーを発行した可能性があります。

コマンドを再発行してください。

```
cachefspack: filename - ファイルを pack できません。I/O エラー
```

エラーが発生した理由

問題を解決する方法

ハードウェアの障害です。

ハードウェアの接続をチェックしてください。

cachefspack: *filename* - ファイルを pack できません。ディスクに空き領域がありません

| エラーが発生した理由 | 問題を解決する方法 |
|-------------------|---------------------|
| ディスク容量が足りなくなりました。 | ディスク容量を増やす必要があります。 |
| キャッシュが最大容量に達しました。 | キャッシュのサイズを増やしてください。 |

cachefspack: *filename* - ファイルを unpack できません。アクセス権がありません

| エラーが発生した理由 | 問題を解決する方法 |
|--|--------------------|
| 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。 | 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はありません

| エラーが発生した理由 | 問題を解決する方法 |
|-------------------------|---------------------|
| 正しいファイルまたはディレクトリがありません。 | 入力ミスがないかチェックしてください。 |

cachefspack: *filename* - ファイルを unpack できません。stale NFS file handle

| エラーが発生した理由 | 問題を解決する方法 |
|---|---------------------|
| アクセスしたときに、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。 | システム管理者に問い合わせてください。 |

cachefspack: *filename* - ファイルを unpack できません。interrupted system call

エラーが発生した理由

問題を解決する方法

このコマンドの実行途中で Control-c キーを発行した可能性があります。

コマンドを再発行してください。

cachefspack: *filename* - ファイルを unpack できません。I/O エラー

エラーが発生した理由

問題を解決する方法

ハードウェアの障害です。

ハードウェアの接続をチェックしてください。

cachefspack: 'd'、'i'、'p'、'u' オプションのどれか 1 つを指定できます。

エラーが発生した理由

問題を解決する方法

コマンドに対して上記オプションを複数入力しました。

オプションは 1 つだけ選択してください。

cachefspack: 環境変数を検索できません。

エラーが発生した理由

問題を解決する方法

構成ファイル中で、\$ で指定した環境変数を設定していません。

環境変数を適切な場所に定義してください。

cachefspack: LIST コマンドを無視します - 動作中の base がありません

エラーが発生した理由

問題を解決する方法

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。

BASE コマンドを定義してください。

CacheFS の統計情報

CacheFS の統計情報は、次のように利用できます。

- 適切なキャッシュサイズを判断する。
- キャッシュの性能を監視する。

これらの結果から得られた情報を使用すると、キャッシュサイズと望ましい性能を取捨選択して調整できます。

CacheFS の統計に関連する、次の 3 つのコマンドがあります。

| | |
|----------------------------|---|
| <code>cachefslog</code> | 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。詳細は、 <code>cachefslog(1M)</code> のマニュアルページを参照してください。 |
| <code>cachefswssize</code> | 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。詳細は、 <code>cachefswssize(1M)</code> のマニュアルページを参照してください。 |
| <code>cachefsstat</code> | 特定のファイルシステム、またはキャッシュされたすべてのファイルシステムに関する統計情報を表示します。このコマンドの出力に含まれる情報は、キャッシュから直接取り出されません。詳細は、 <code>cachefsstat(1M)</code> のマニュアルページを参照してください。 |

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(1M)` コマンドを実行するには、スーパーユーザーにならなければなりません。

統計の累計は、ログファイルの作成時から始まります。作業時間が終わったら、`cachefslog -h` コマンドを使用してロギングを停止してください。手順については、515ページの「ロギングプロセスを停止する方法」を参照してください。

CacheFS 統計情報を設定して表示するための前提条件

CacheFS 統計情報コマンドを使用する前に、次の操作を実行しておかなければなりません。

- `cfsadmin(1M)` コマンドを使用してキャッシュを設定する。
- 作成するログファイルに統計情報を収集できるように、適切な時間を決定する。この時間は、1日、1週間、1カ月など、一般的な作業間隔に等しくする必要がある。
- ログファイルの位置またはパスを選択する。ログファイルが大きくなっても対応できる程度の領域があることを確認する。ログファイルに統計情報を収集できる時間を長くするほど、大きな領域が必要になる。

注 - 次の手順は推奨する順序を示しています。異なる順序で作業してもかまいません。

CacheFS 統計情報の設定

表 29-3 に、CacheFS 統計情報の設定に関連する手順を示します。

表 29-3 作業マップ : CacheFS 統計情報の設定

| 作業 | 説明 | 手順の説明 |
|----------------|---|--------------------------|
| 1. ログインの設定 | <code>cachefslog</code> コマンドを使用して、キャッシュされたファイルシステムのログインを設定する。 | 514ページの「ログインプロセスを設定する方法」 |
| 2. ログファイルの検索 | <code>cachefslog</code> コマンドを使用してログファイルの位置を特定する。 | 515ページの「ログファイルの場所を調べる方法」 |
| 3. ログインプロセスの停止 | <code>cachefslog</code> コマンドを使用してログインプロセスを停止する。 | 515ページの「ログインプロセスを停止する方法」 |

表 29-3 作業マップ : CacheFS 統計情報の設定 続く

| 作業 | 説明 | 手順の説明 |
|------------------|--|------------------------------------|
| 4. キャッシュサイズの表示 | cache <code>fs</code> sw <code>size</code> コマンドを使用してキャッシュサイズを表示する。 | 516ページの「作業セット (キャッシュ) のサイズを表示する方法」 |
| 5. キャッシュ統計情報の表示。 | cache <code>fs</code> s <code>stat</code> コマンドを使用して統計情報を表示する。 | 518ページの「キャッシュ統計情報を表示する方法」 |

CacheFS ロギング

この節では CacheFS ロギングを設定し、表示する方法を説明します。

▼ ロギングプロセスを設定する方法

1. cache`fs`s`log` コマンドを使用してロギングプロセスを設定します。

```
$ cachefsslog -f log-file-path mount-point
```

| | |
|----------------------------|--|
| <code>-f</code> | ロギングプロセスを設定する |
| <code>log-file-path</code> | ログファイルの位置を表す。ログファイルは、vi などのエディタで作成する標準ファイル |
| <code>mount-point</code> | 統計情報を収集するマウントポイント (キャッシュされたファイルシステム) を指定する |

2. ログファイルを正しく設定したかどうかを確認するには、次のように cache`fs`s`log` コマンドを使用します。

```
$ cachefsslog mount-point
```



```
$ cachefslog -h /home/stuff
not logged: /home/stuff
```

上記の例と異なるシステム応答が表示される場合は、ロギングプロセスが正常に停止されていません。正しいログファイル名とマウントポイントを指定したかどうかを確認してください。

キャッシュサイズの表示

キャッシュサイズを増やすべきかどうかを確認したり、特定のマウントポイントに関して前回 `cachefslog(1M)` コマンドを使用した後の作業から理想的なキャッシュサイズを決定したりできます。

▼ 作業セット (キャッシュ) のサイズを表示する方法

1. スーパーユーザーになります。
2. `cachefswssize(1M)` コマンドを使用して、現在のログのキャッシュサイズと最大キャッシュサイズを表示します。

```
# cachefswssize log-file-path
```

例 — 作業セット (キャッシュ) のサイズを表示する

次の例で、「end size」とは `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
```

(続く)

```
    /opt
      end size:    128k
    high water size: 128k

    /nfs/saturn.dist
      end size:   1472k
    high water size: 1472k

    /usr/openwin
      end size:   7168k
    high water size: 7168k

    /nfs/venus.svr4
      end size:   4688k
    high water size: 5000k

    /usr
      end size:   4992k
    high water size: 4992k

    total for cache
      initial size: 110960k
      end size:    30872k
    high water size: 30872k
```

統計情報の表示

キャッシュされた特定のファイルシステムに関する情報を表示できます。次の表は、統計情報の出力時に表示される用語を示しています。

表 29-4 統計情報の出力に表示される用語

| 用語 | 説明 |
|-----------------|--|
| ヒット率 (hit rate) | キャッシュのヒット率対ミスヒット率の比と、それに続く実際のヒット数とミスヒット数。キャッシュヒットは、ユーザーがファイル操作を実行したいときに、そのファイルが実際にはキャッシュ内にあると発生する。キャッシュのミスヒットは、ファイルがキャッシュになかったときに発生する。サーバーにかかる負荷は、キャッシュのミスヒット数、整合性チェック数、および変更数の合計である |
| チェック数 (checks) | 実行された整合性チェックの回数、合格回数、不合格回数 |
| 変更数 (modifies) | 書き込みや作成など、変更操作の回数 |

▼ キャッシュ統計情報を表示する方法

`cachefsstat (1M)` コマンドを使用して統計情報を表示します。この操作はいつでも実行できます。たとえば、ロギングを設定しなくても統計情報を表示できます。

```
$ cachefsstat mount-point
```

mount-point

統計情報を表示したいキャッシュされたファイルシステム

マウントポイントを指定しなければ、マウントされているすべての CacheFS ファイルシステムに関する統計情報が表示されます。

例 — 統計情報を表示する

```
$ cachefsstat /home/sam
cache hit rate: 73% (1234 hits, 450 misses)
consistency checks: 700 (650 pass, 50 fail)
modifies: 321
```


キャッシュの構造と動作

各キャッシュには、その動作と構造を決定する一連のパラメータが付いています。各パラメータは、表 29-5に示すデフォルト値に設定されています。デフォルト値は、フロントファイルシステム全体をキャッシュに使用するように指定しますが、これはファイルシステムをキャッシュに書き込む場合の推奨方法です。

表 29-5 キャッシュパラメータとデフォルト値

| キャッシュパラメータ | デフォルト値 | 定義 |
|--------------|--------|---|
| maxblocks | 90% | CacheFS がフロントファイルシステム内で要求できる最大ブロック数を設定する |
| minblocks | 0% | CacheFS がフロントファイルシステム内で要求できる最小ブロック数を設定する |
| threshblocks | 85% | CacheFS が minblocks で指定したより多数ブロックを要求する前に、フロントファイルシステム内で使用可能でなければならないブロック数を設定する |
| maxfiles | 90% | CacheFS がフロントファイルシステム内で要求できる使用可能 i ノードの最大数 (ファイル数) を指定する |
| minfiles | 0% | CacheFS がフロントファイルシステム内で要求できる使用可能 i ノードの最小数 (ファイル数) を指定する |
| threshfiles | 85% | CacheFS が minfiles で指定したより多数のファイルを要求する前に、フロントファイルシステム内で使用可能でなければならない i ノード数 (ファイル数) を設定する |

通常、これらのパラメータ値を変更する必要はありません。最適のキャッシュ動作が得られるデフォルト値に設定されています。ただし、キャッシュに使用されないフロントファイルシステム内に空き空間があり、それを他のファイルシステムに使用したい場合は、maxblocks と maxfiles の設定を変更できます。そのためには、cfsadmin(1M) コマンドを使用します。たとえば、次のようになります。

```
$ cfsadmin -o maxblocks=60
```

キャッシュされたファイルシステムとバック ファイルシステムとの整合性チェック

CacheFS は、キャッシュされたディレクトリとファイルが最新の状態に保たれることを保証するために、キャッシュに格納されたファイルの整合性を定期的にチェックします。整合性をチェックするために、CacheFS は現在の変更時刻を前回の変更時刻と比較します。変更時刻が異なる場合は、そのディレクトリまたはファイルに関するすべてのデータと属性がキャッシュから消去され、バックファイルシステムから新しいデータと属性が取り出されます。

ユーザーがディレクトリやファイルの処理を要求すると、CacheFS は整合性を検査する時間があるかどうかをチェックします。時間があれば、バックファイルシステムから変更時刻を取得して比較します。

必要に応じて行う整合性チェック

mount (1M) コマンドの demandconst オプションを指定すると、このオプションでマウントしたファイルシステムに関して明示的に要求するときのみ、整合性チェックを実行できます。ファイルシステムをキャッシュにマウントするときに demandconst オプションを指定すると、-s オプションを指定して cfsadmin コマンドを使用し、整合性チェックを要求します。デフォルトでは、ファイルがアクセスされるたびに、各ファイルの整合性がチェックされます。ファイルがアクセスされなければ、チェックは実行されません。demandconst オプションを使用すると、整合性チェックによってネットワークがあふれる事態を回避できます。この整合性チェックの詳細は、cfsadmin (1M) のマニュアルページを参照してください。

追加スワップ空間の構成の手順

この章の内容は次のとおりです。

- 522ページの「スワップ空間と仮想メモリー」
- 523ページの「スワップ空間と TMPFS ファイルシステム」
- 524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」
- 525ページの「スワップ空間の割り当て方法」
- 526ページの「スワップ空間の計画」
- 527ページの「スワップ資源の監視」
- 528ページの「スワップ空間の追加」
- 530ページの「スワップファイルを削除する」

スワップ空間について

この節では、スワップ空間の概念と、SunOS 4.0/4.1 と SunOS 5.7 のスワップ要件の違いについて説明します。すでに SunOS 5.7 のスワップ機構について精通している方は、526ページの「スワップ空間の計画」の節に進んでください。

管理者は、次の事柄を決定する上で SunOS 5.7 のスワップ機構を理解しておく必要があります。

- スワップ空間の要件
- tmpfs ファイルシステムとの関係

- スワップ空間に関連するエラーメッセージからの復元

スワップ空間と仮想メモリー

SunOS 5.7 システムソフトウェアは、一時記憶域にファイルシステムではなくディスクスライスを使用します。これらのスライスを「スワップ」スライスと呼びます。スワップスライスは、システムの物理メモリーが不足し現在のプロセスを処理することができないときに、仮想メモリー記憶域として使用されます。

SunOS 5.7 の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。ユーザーにはメモリーをバックアップしているファイル名はわからないため、スワップ空間から読み直されたメモリーは **anonymous** メモリーとして参照されます。

SunOS 4.0/4.1 の **anonymous** メモリーページは、システムのスワップ空間プールからランダムに割り当てられた名前を使用して対応付けられます。これらのメモリーページの用途は次のとおりです。

- 書き込み時コピー処理中に作成されるデータの専用コピー
- プロセスとスタックのセグメント
- TMPFS ファイルシステム記憶域の資源

SunOS 4.0/4.1 の **anonymous** メモリーの実装には、次の制限があります。

- アプリケーションで使用されない場合にも、**anonymous** メモリーのマッピング用に必ず物理記憶域 (ディスク上にとられたバックアップ用のスワップ空間) を確保しなければならない。

たとえば、大量のデータセグメントを持つアプリケーションは、各ページが物理記憶域にデータが書き出されない場合でも、大量のスワップ空間を使用して構成しなければならない。

- バックアップとなる記憶域はランダムに選択され、変更できないので、**anonymous** メモリーページを物理記憶域に関連付けるための方法には制限があり、柔軟性に欠ける。

SunOS 5.7 ソフトウェア環境には、「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

SunOS 5.7 環境では、TMPFS ファイルシステムは `/etc/vfstab` ファイル内のエントリによって自動的に稼働されます。TMPFS ファイルシステムは、ファイルとそれに関連付けられた情報をディスクではなくメモリー (`/tmp` ディレクトリ内) に格納するので、これらのファイルへのアクセスが高速になります。このため、コンパイラや DBMS 製品のように `/tmp` の使用量の大きいアプリケーションの場合は、性能が大幅に改善されます。

TMPFS ファイルシステムは、システムのスワップ資源から `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` 内の領域を使い果たすと、スワップ空間も使い果たしたことになります。したがって、アプリケーションの `/tmp` の使用量が大きい場合に、スワップ空間の使用状況を監視しなければ、システムがスワップ空間を使い果たす可能性があります。

TMPFS を使用したいがスワップ資源が限られている場合は、次の方法を使用してください。

- サイズオプション (`-o size`) を指定して TMPFS ファイルシステムをマウントし、スワップ資源 TMPFS をどの程度使用できるかを制御する。
- スワップ空間が足りなくなった場合は、コンパイラを `TMPDIR` 環境変数を使用して、より大きな一時ディレクトリを指すことができる。

コンパイラの `TMPDIR` 変数を使用すると、コンパイラが `/tmp` を使用するかどうかだけが制御される。他のプログラムによる `/tmp` の使用には影響しない。

スワップ空間の追加が必要かどうかを調べる方法

この節では、スワップ空間を使い果たしたときに表示されるエラーメッセージについて説明します。

スワップ関連のエラーメッセージ

次のメッセージは、アプリケーションがさらに `anonymous` メモリーを取得しようとしたが、バックアップ用のスワップ空間が残っていなかったことを示します。

```
application is out of memory
malloc error 0
WARNING: Sorry, no swap space to grow stack for pid
```

TMPFS 関連のエラーメッセージ

```
directory: File system full, swap space limit exceeded
```

このメッセージは、ファイルに書き込むときにページを割り当てることができない場合に表示されます。このエラーは、TMPFS が許容限度を超えて書き込もうとしたときや、現在実行されているプログラムが大量のメモリーを使用している場合に発生することがあります。

```
directory: File system full, memory allocation failed
```

このメッセージは、TMPFS が新しいファイルやディレクトリの作成中に物理メモリーを使い果たしたことを意味します。

TMPFS 関連のエラーメッセージから回復する方法については、TMPFS (7FS) のマニュアルページを参照してください。

スワップ空間の割り当て方法

最初に、スワップ空間は Solaris インストールプロセスの一部として割り当てられます。インストールプログラムによるディスクスライスの自動レイアウトを選択し、スワップファイルのサイズを手作業で変更しなければ、Solaris インストールプログラムは表 30-1 に示すようにデフォルトのスワップスライスを割り当てます。

表 30-1 デフォルトのスワップ空間割り当て

| システムに「n」M バイトの物理メモリーがある場合 | 割り当てられるディレクトリのスワップ空間 |
|---------------------------|----------------------|
| 16-64 | 32M バイト |
| 64-128 | 64M バイト |
| 128-512 | 128M バイト |
| 512 を超える場合 | 256M バイト |

/etc/vfstab ファイル

システムをインストールすると、スワップスライスとファイルは /etc/vfstab ファイル内に列挙され、システムのブート時に /sbin/swapadd スクリプトによって起動されます。

/etc/vfstab ファイル内のスワップデバイスエントリには、次の情報が入っています。

- スワップスライスまたはファイルの完全パス名
- スワップのファイルシステムタイプ

スワップファイルが入っているファイルシステムは、スワップファイルが稼働される前にマウントしなければならないので、ファイルシステムをマウントするエントリがスワップファイルを起動するエントリより前に入っていることを、/etc/vfstab ファイル内で確認してください。

スワップ空間の計画

スワップ空間のサイズを決定する上で最も重要な要素は、システムのソフトウェアアプリケーションの要件です。たとえば、コンピュータ支援設計シミュレータ、データベース管理製品、トランザクションモニター、地質分析システムなどの大型アプリケーションは、超大型メモリーシステム内で 200 ~ 1000M バイトのスワップ空間を消費することがあります。

一般にデータファイルのサイズが 10 ~ 20M バイトを超えるアプリケーションの場合は、スワップ空間の要件をアプリケーションベンダーに問い合わせてください。

アプリケーションベンダーからスワップ空間の要件を入手できない場合は、次のガイドラインに従ってスワップ空間を割り当ててください。

- アプリケーションをサポートするために、次のように割り当てる。
 - xterm のような通常のアプリケーションごとに 1M バイト
 - カレンダーやメールアプリケーションのような軽量アプリケーションごとに 2 ~ 3M バイト
 - デスクトップパブリッシング (DTP) ソフトウェアのような大型アプリケーションには 20 ~ 50M バイト
- クラッシュダンプを節約するために、物理メモリー全体を割り当てて最悪の場合のクラッシュダンプを保存する。
- システムまたはアプリケーションの要件がわからない場合は、システムの物理メモリーのうち 50% ないし 100% を割り当てる。たとえば、32M バイトの物理メモリーを持つシステムの場合は、16 ~ 32M バイトのスワップ空間を割り当てる。これにより、仮想スワップ空間の合計は 48 ~ 64M バイトになる。
- 大型アプリケーション (コンパイラなど) が /tmp ディレクトリを使用するかどうかを決定する。次に、/tmpfs に使用される追加のスワップ空間を割り当てる。TMPFS については、523ページの「スワップ空間と TMPFS ファイルシステム」を参照。

スワップ資源の監視

/usr/sbin/swap コマンドを使用してスワップ空間を管理します。2つのオプション `-l` と `-s` を使用して、スワップ資源に関する情報を表示します。

`swap -l` コマンドを使用してシステムのスワップ空間を調べます。有効なスワップデバイスやファイルは、`swapfile` カラムの下に表示されます。

```
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t2d0s1 32,17  8 205624 192704
```

`swap -s` コマンドを使用してスワップ資源を監視します。

```
# swap -s
total: 10492k bytes allocated + 7840k reserved = 18332k used, 21568k available
```

`used` と `available` の数値の統計は、システム上の合計スワップ空間に等しくなります。これには、物理メモリーの一部とスワップデバイス (またはファイル) が含まれます。

使用可能なスワップ空間と使用済みスワップ空間の容量 (`swap -s` の出力内) を使用して、時間経過に伴うスワップ空間の使用状況を監視できます。システム性能が適正であれば、`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。システム性能が低下したときは、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。これにより、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` からのブロック数を合計して K バイト数に換算すると、`used + (swap -s の出力内での) available` よりも少なくなります。

表 30-2 に、`swap -s` コマンドの出力とその説明を示します。

表 30-2 `swap -s` コマンドの出力

| キーワード | 説明 |
|------------------------------|--|
| <code>bytes allocated</code> | 現在バックアップ用の記憶域 (ディスク上にとられたバックアップ用のスワップ空間) として使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数 |
| <code>reserved</code> | 現在は割り当てられていないが、後から使用できるようにメモリーによって回収されるスワップ空間の合計容量を表す 1024 バイトのブロック数 |
| <code>used</code> | 割り当て済みまたは予約済みのスワップ空間の合計容量を表す 1024 バイトのブロック数 |
| <code>available</code> | 後から予約や割り当てに使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数 |

スワップ空間の追加

システム構成を変更して新しいソフトウェアパッケージをインストールした後に、スワップ空間を追加しなければならないことがあります。その場合に望ましいのは、`mkfile` コマンドと `swap` コマンドを使用して、既存の UFS または NFS ファイルシステムの一部を補助スワップ空間として指定する方法です。次に説明するように、これらのコマンドを使用すると、ディスクをパーティションに分割し直さなくても、スワップ空間を追加できます。

スワップ空間を追加するもう 1 つの方法は、ディスクをパーティションに分割し直すことです。その方法については、第 21 章で使用中のシステムに該当する箇所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。

表 30-3 は、`mkfile` コマンドのオプションを示しています。

表 30-3 `mkfile` コマンドのオプション

| オプション | 説明 |
|-----------------|---|
| <code>-n</code> | 空のファイルを作成する。サイズは表示されるが、データが書き込まれるまでディスクブロックは割り当てられない。 |
| <code>-v</code> | 詳細モード。作成されたファイル名とサイズが表示される。 |



注意 `-n` オプションは、NFS スワップファイルの作成時のみ使用してください。

▼ スワップファイルを作成して使用可能にする方法

1. スーパーユーザーになります。
`root` 権限がなくてもスワップファイルを作成できますが、スワップファイルが不注意から上書きされないように、`root` を所有者にしておくといでしょう。
2. スワップファイルを作成します。

```
# mkfile nnn[k|b|m] filename
```

サイズ「`nnn`」(K バイト数、バイト数、M バイト数) のスワップファイルと指定したファイル名が作成されます。

3. スワップファイルを有効にします。

```
# /usr/sbin/swap -a /path/filename
```

絶対パス名を使用してスワップファイルを指定しなければなりません。スワップファイルが追加され、ファイルシステムがマウント解除されるか、またはシステムがリブートされるまで使用可能になります。

4. ファイルの完全パス名を指定し、ファイルシステムのタイプとして swap を指定して、スワップファイルのエントリを /etc/vfstab ファイルに追加します。

```
/path/filename - - swap - no -
```

5. スワップファイルが追加されたことを確認するには、次のように入力します。

```
/usr/sbin/swap -l
```

例 — スワップファイルを作成して使用可能にする

次の例では、/files/swapfiles という 24M バイトのスワップファイルを作成します。

```
# mkfile 24m /files/swapfile
# swap -a /files/swapfile
# vi /etc/vfstab
(エントリがスワップファイル用に追加される)
/files/swapfile - - swap - no -
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/dsk/c0t2d0s1 32,17 8 205624 192704
/files/swapfile   -      8 40952 40952
```

スワップファイルを削除する

不要になった余分なスワップ空間は、削除することができます。

▼ 余分なスワップ空間を削除する方法

1. スーパーユーザーになります。
2. 次のように `swap -d` コマンドを使用してスワップ空間を削除します。

```
# /usr/sbin/swap -d /path/filename
```

スワップファイル名がリストから削除されるので、スワッピングに使用できなくなります。ファイルそのものは削除されません。

3. `/etc/vfstab` ファイルを編集して、スワップファイルのエントリを削除します。
4. 他の目的に使用できるようにディスク領域を復元します。

```
# rm swap-filename
```

スワップ空間がファイルの場合は削除します。また、スワップ空間が別のスライスに入っていて、不要なことがわかっている場合は、新しいファイルシステムを作成してマウントします。

詳細は、第 28 章を参照してください。

例 — 余分なスワップ空間を削除する

次の例は、`/files/swapfile` スワップファイルを削除します。

```
# swap -d /files/swapfile
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/dsk/c0t2d0s1 32,17      8 205624 192720
```


ファイルシステムの整合性チェック

この章の内容は以下のとおりです。

- 534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」
- 536ページの「fsck でチェックして修復される内容」
- 543ページの「自動ブートチェック機能の変更」
- 546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」
- 549ページの「不正なスーパーブロックの復元」
- 552ページの「fsck コマンドの構文とオプション」

fsck のエラーメッセージについては、『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」を参照してください。

UFS ファイルシステムは、一連の内部テーブルを基にして使用済み i ノード、使用可能ブロックを特定します。これらの内部テーブルがディスク上のデータと正しく同期していないと、整合性が失われ、ファイルシステムの修復が必要になります。

次のような原因でオペレーティングシステムが異常終了すると、ファイルシステムが損傷したり、整合性が失われたりすることがあります。

- 電源障害
- 不注意によるシステム電源の切断
- 正しいシャットダウン手順以外の方法によるシステム電源の切断
- カーネル内のソフトウェアエラー

ファイルシステムの破損は重大ですが、あまり頻繁に起きるものではありません。システムをブートすると、ファイルシステムの整合性チェックが (fsck プログラム

を使用して)自動的に実行されます。ほとんどの場合は、このファイルシステムのチェックによって問題が修復されます。

この章では、`fsck` プログラムでチェックされ修復される問題と `fsck` のオプションについて説明します。また、次の作業についても説明します。

- ブート時に実行される自動チェック機能を変更する方法
- ファイルシステムのチェックが必要かどうかを調べる方法
- UFS ファイルシステムを対話形式でチェックして修復する方法
- 不良スーパーブロックを復元する方法
- `fsck` で修復できない UFS ファイルシステムを修復する方法

`fsck` のエラーメッセージについては、『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」を参照してください。

`fsck` プログラムは、ファイルシステム上に配置されているが参照不可能なファイルとディレクトリを `lost+found` ディレクトリに入れます。各ファイルの `i` ノード番号が名前として割り当てられます。`lost+found` ディレクトリが存在しない場合は、`fsck` によって作成されます。`lost+found` ディレクトリ内の領域が足りない場合は、そのサイズが `fsck` によって拡張されます。

ファイルシステムの状態はどのように記録されるか

`fsck` コマンドは、スーパーブロックに格納された状態フラグを使用して、ファイルシステムの状態を記録します。また、このフラグを使用して、ファイルシステムの整合性をチェックする必要があるかどうか判断されます。このフラグはブート時には `/sbin/rcS` スクリプトによって使用され、コマンド行からは `fsck` に `-m` オプションを指定した場合に使用されます。`fsck` の `-m` オプションの結果を無視するのなら、状態フラグの設定に関係なく、すべてのファイルシステムをチェックできます。

表 31-1 に状態フラグを示します。

表 31-1 状態フラグの値

| 状態フラグの値 | 説明 |
|----------|---|
| FSACTIVE | ファイルシステムのマウント後、変更されると、状態フラグが FSACTIVE に設定される。ファイルシステムの整合性が失われている可能性がある。変更後のメタデータがディスクに書き込まれるまでは、ファイルシステムに FSACTIVE マークが付けられる。ファイルシステムが正常にマウント解除されると、状態フラグは FSCLEAN に設定される。FSACTIVE フラグが設定されたファイルシステムは、整合性がないので、fsck でチェックしなければならない。 |
| FSBAD | ルート (/) ファイルシステムが、FSCLEAN でも FSSTABLE でもない状態のときにマウントされると、状態フラグがFSBAD に設定される。カーネルが、このファイルシステムの状態を FSCLEAN または FSSTABLE に変更することはない。ブートの処理の一部として、ルート (/) ファイルシステムにFSBAD フラグが設定された場合、ルートファイルシステムは読み取り専用としてマウントされる。ルートの raw デバイスに対して fsck を実行する。その後で、ルート (/) ファイルシステムを読み取り/書き込みとしてマウントし直す。 |
| FSCLEAN | ファイルシステムが正しくマウント解除された場合は、状態フラグが FSCLEAN に設定される。FSCLEAN 状態フラグが設定されているファイルシステムは、システムのブート時にチェックされない。 |
| FSLOG | UFS ログ機能を有効にしてファイルシステムがマウントされている場合、状態フラグがFSLOG に設定される。システムのブート時、状態フラグがFSLOG のファイルシステムはチェックされない。 |
| FSSTABLE | ファイルシステムはマウントされている (またはされた) が、前回のチェックポイント (sync または fsflush) 以後に変更がなかった。チェックポイントは、通常は 30 秒ごとに発生する。たとえば、カーネルはファイルシステムがアイドル状態かどうかを定期的にチェックし、アイドル状態であれば、スーパーブロック内の情報をディスクにフラッシュさせて FSSTABLE マークを設定する。システムがクラッシュした場合、ファイルシステムの構造は正しいが、少量のデータが失われている可能性がある。FSSTABLE マークが付いたファイルシステムは、マウント前のチェックをスキップできる。ファイルシステムの状態が FSCLEAN でも FSSTABLE でもなければ、mount (2) システムコールではファイルシステムは読み取り/書き込み用にマウントされない。 |

表 31-2 に、fsck コマンドを使用して、初期状態に基づいて状態フラグを変更する方法を示します。

表 31-2 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 |
| unknown | FSSTABLE | FSSTABLE | unknown |
| FSACTIVE | FSSTABLE | FSSTABLE | FSACTIVE |
| FSSTABLE | FSSTABLE | FSSTABLE | FSACTIVE |
| FSCLEAN | FSCLEAN | FSSTABLE | FSACTIVE |
| FSBAD | FSSTABLE | FSSTABLE | FSBAD |
| FSLOG | FSLOG | FSLOG | FSLOG |

fsck でチェックして修復される内容

この節では、ファイルシステムの通常の処理中に発生する問題、原因、fsck で検出される問題、およびそれらの修正方法について説明します。

非整合状態が発生する原因

労働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、整合性のあるファイルシステムが生成されます。

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内部のインコアバッファにコピーされます。一般に、ディスクの更新は非同期に処理されるので、ユーザープロセスは、書き込みシステムコールが値を返した後すぐに処理を続けることができますが、実際へのデータの書き込みは、ずいぶん後に実行されることもあります。したがって、

ディスク上にあるファイルシステムは、インコア情報で表されるファイルシステムの状態から常に遅延することになります。

別の目的にバッファが必要になったり、カーネルが `fsflush` デーモンを自動的に (30 秒間隔で) 実行すると、インコア情報を反映するようにディスク情報が更新されます。システムがインコア情報を書き込まずに停止すると、ディスク上のファイルシステムの整合性がなくなります。

ファイルシステムの整合性は、さまざまな原因で失われることがあります。最も一般的な原因は、オペレータのエラーとハードウェア障害です。

システムを正しくシャットダウンしなかったり、マウントされているファイルシステムが正しくオフラインにされないと、「クリーンでない停止」が原因で問題が発生することがあります。クリーンでない停止を防ぐには、CPU を停止したり、ディスクをドライブから物理的に取り出したり、ディスクをオフライン状態にする前に、ファイルシステムの現在の状態をディスクに書き込まなければなりません (つまり、同期させなければなりません)。

また、ハードウェアの欠陥が原因で整合性が失われることもあります。ディスクドライブ上ではいつでもブロックが損傷する可能性があり、ディスクコントローラが正常に機能しなくなる可能性があります。

整合性がチェックされる UFS 構成要素

この節では、UFS ファイルシステムの構成要素、つまりスーパーブロック、シリンダグループブロック、i ノード、間接ブロック、データブロックに `fsck` が適用する整合性チェックの種類について説明します。

スーパーブロック

スーパーブロックには集計情報が格納されており、UFS ファイルシステム内で最も破損しがちな構成要素です。ファイルシステムの i ノードやデータブロックが変更されるたびに、スーパーブロックも変更されます。CPU が停止した場合、直前のコマンドが `sync` コマンドでなければ、スーパーブロックはほぼ確実に破損します。

スーパーブロックの非整合性は、次の面からチェックされます。

- ファイルシステムのサイズ
- i ノード数
- 空きブロック数

■ 空き *i* ノード数

ファイルシステムと *i* ノードリストのサイズ

ファイルシステムのサイズは、スーパーブロックに使用されるブロック数と *i* ノードリストに使用されるブロック数よりも大きくなければなりません。*i* ノード数は、ファイルシステムの最大許容数よりも小さくなければなりません。ファイルシステムのサイズとレイアウト情報は、`fsck` にとって最も重要な情報部分です。これらのサイズを実際にチェックする方法はありませんが、ファイルシステムの作成時に静的に判断されるので、`fsck` はサイズが妥当な範囲内にあるかどうかをチェックできます。ファイルシステムの他のすべてのチェックを行うには、これらのサイズが正確でなければなりません。`fsck` が一次スーパーブロックの静的パラメータ内に不正な情報を検出すると、オペレータに代替スーパーブロックの位置を指定するように促します。

空きブロック数

空きブロック数は、シリンダグループのブロックマップに格納されます。`fsck` は、空きマーク付きのすべてのブロックがファイルによって使用されていないかどうかをチェックします。すべてのブロックをチェックし終わると、`fsck` は空きブロック数と *i* ノードによって使用されるブロック数の合計がファイルシステム内の合計ブロック数に等しくなるかどうかをチェックします。ブロック割り当てマップ内に間違いがあると、`fsck` はブロックが割り当てられている状態のままに構築し直します。

スーパーブロック内の集計情報には、ファイルシステム内の空きブロックの合計数のカウントが入っています。`fsck` プログラムは、このブロック数をファイルシステム内で見つかった空きブロック数と比較します。数が一致しなければ、`fsck` はスーパーブロック内の空きブロック数を実際の空きブロック数で置き換えます。

空き *i* ノード数

スーパーブロック内の集計情報には、ファイルシステム内の空き *i* ノード数が入っています。`fsck` プログラムは、この *i* ノード数をファイルシステム内で見つかった空き *i* ノード数と比較します。数が一致しなければ、`fsck` はスーパーブロック内の空き *i* ノード数を実際の空き *i* ノード数で置き換えます。

i ノード数

i ノードリストは、i ノード 2 から順番にチェックされます (i ノード 0 と i ノード 1 は予約されています)。各 i ノードの非整合性は、次の面からチェックされます。

- 形式とタイプ
- リンク数
- 重複ブロック
- 不正なブロック番号
- i ノードのサイズ

i ノードのフォーマットとタイプ

各 i ノードには、そのタイプと状態を記述するモードのワードが入っています。i ノードには、次の 6 つのタイプがあります。

- 通常ファイル
- ディレクトリ
- ブロック型特殊ファイル
- キャラクタ型特殊ファイル
- FIFO (名前付きパイプ)
- シンボリックリンク

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もので空のものです。不完全に割り当て済みの状態は、i ノードが正しくフォーマットされていないことを意味します。たとえば、ハードウェア障害が原因で i ノードに不正なデータが書き込まれると、i ノードは不完全に割り当て済みの状態になることがあります。fsck が実行できる唯一の修正動作は、その i ノードを消去することです。

リンク数

各 *i* ノードには、そこにリンクされているディレクトリエントリ数が入っています。fsck プログラムは、ルートディレクトリから順番にディレクトリ構造全体を検査し、*i* ノードごとに実際のリンク数を計算して、各 *i* ノードのリンク数を検査します。

i ノードに格納されているリンク数が fsck によって判断された実際のリンク数と一致しない場合は、次の 3 つの状況が考えられます。

- 格納されたリンク数が 0 でなく、実際のリンク数が 0 の場合

この状況は、*i* ノードにリンクされているディレクトリエントリが存在しない場合に発生することがあります。この場合、fsck はリンクされていないファイルを `lost+found` ディレクトリに入れます。

- 格納されたリンク数が 0 でなく、実際のリンク数も 0 でないが、2 つのリンク数が等しくない場合

この状況は、ディレクトリエントリが追加または削除されたが、*i* ノードが更新されていない場合に発生することがあります。この場合、fsck はリード内のリンク数を実際のリンク数で置き換えます。

- 格納されたリンク数が 0 で実際のリンク数が 0 でない場合

この場合、fsck は *i* ノード内のリンク数を実際のリンク数に変更します。

重複ブロック数

各 *i* ノードには、それが使用するすべてのブロックのリスト、またはリストを指すポインタ (間接ブロック) が入っています。間接ブロックは *i* ノードによって所有されるので、間接ブロックの整合性が失われると、それを所有する *i* ノードが直接影響を受けます。

fsck プログラムは、*i* ノードから使用される各ブロック番号を、割り当て済みブロックのリストと比較します。別の *i* ノードからすでにブロック番号が使用されていると、そのブロック番号は重複ブロックのリストに入れられます。それ以外の場合は、割り当て済みブロックのリストが更新され、ブロック番号が追加されます。

重複ブロックがあると、fsck は再び *i* ノードリストを調べて、各重複ブロックを使用する他の *i* ノードを検索します (*i* ノード内に大量の重複ブロックが入っている場合は、ファイルシステムに間接ブロックが正しく書き込まれていない可能性があります)。どの *i* ノードにエラーがあるかを正確に判断することはできません。fsck

プログラムは、保持する *i* ノードと消去する *i* ノードを選択するように促すプロンプトを表示します。

不正なブロック番号

`fsck` プログラムは、*i* ノードから使用される各ブロック番号をチェックして、その値が最初のデータブロック番号よりも大きく、ファイルシステム内の最後のデータブロック番号より小さいかどうかを調べます。ブロック番号がこの範囲に含まれない場合は、不正なブロック番号と見なされます。

間接ブロックがファイルシステムに正しく書き込まれていないことが原因で、*i* ノード内に不正なブロック番号が発見されることがあります。`fsck` はその *i* ノードの消去を促すプロンプトを表示します。

i ノードサイズ

各 *i* ノードには、参照するデータブロック数が入っています。実際のデータブロック数は、割り当て済みのデータブロック数と間接ブロック数の合計です。`fsck` はデータブロック数を計算し、そのブロック数を *i* ノードから使用されるブロック数と比較します。*i* ノードに不正なブロック数が入っていると、`fsck` はその修正を促すプロンプトを表示します。

各 *i* ノードには、64 ビットのサイズフィールドがあります。このフィールドは、*i* ノードに関連付けられたファイル内の文字数 (データバイト数) を示します。*i* ノードのサイズフィールドに整合性があるかどうかは、サイズフィールド内の文字数を使用して、*i* ノードに関連付けるべきブロック数を計算し、その結果を *i* ノードから使用される実際のブロック数と比較して概算でチェックされます。

間接ブロック

間接ブロックは *i* ノードによって所有されます。したがって、間接ブロック内の整合性が失われると、それを所有する *i* ノードが影響を受けます。非整合性は、次の面からチェックできます。

- すでに別の *i* ノードから使用されているブロック
- ファイルシステムの範囲に含まれないブロック番号

また、間接ブロックの場合は整合性チェックも実行されます。

データブロック

i ノードは、3 種類のデータブロックを直接または間接に参照できます。参照されるブロックは、すべて同じ種類でなければなりません。次の 3 種類のデータブロックがあります。

- プレーンデータブロック
- シンボリックリンクデータブロック
- ディレクトリデータブロック

プレーンデータブロックには、ファイルに格納される情報が入っています。シンボリックリンクデータブロックには、シンボリックリンクに格納されるパス名が入っています。ディレクトリデータブロックには、ディレクトリエントリが入っています。fsck はディレクトリデータブロックの妥当性しかチェックできません。

ディレクトリは、i ノードの mode フィールド内のエントリによって通常ファイルと区別されます。ディレクトリに関連付けられたデータブロックには、ディレクトリエントリが入っています。ディレクトリデータブロックの非整合性は、次の面からチェックされます。

- 未割り当ての i ノードを指すディレクトリ内の i ノード番号
- ファイルシステム内の i ノード番号より大きいディレクトリ内の i ノード番号
- 「.」と「..」ディレクトリには許されないディレクトリ内の i ノード番号
- ファイルシステムから切り離されたディレクトリ

未割り当てディレクトリ

ディレクトリデータブロック内の i ノード番号が未割り当て i ノードを指す場合、fsck はそのディレクトリエントリを削除します。この状況は、ディレクトリエントリが入っているデータブロックが変更されて書き出されたが、i ノードが書き込まれていない場合に発生します。また、警告なしに CPU が停止された場合にも発生します。

不正な i ノード番号

ディレクトリエントリの i ノード番号が i ノードリストの最後を超える位置を指す場合、fsck はそのディレクトリエントリを削除します。この状況は、不正なデータがディレクトリのデータブロックに書き込まれると発生します。

不正な「.」と「..」エン트리

「.」ディレクトリの i ノード番号は、ディレクトリデータブロックの最初のエン트리でなければなりません。また、それ自体を参照しなければなりません。つまり、その値はディレクトリデータブロックの i ノード番号に等しくなければなりません。

「..」ディレクトリの i ノード番号は、ディレクトリデータブロックの第 2 のエン트리でなければなりません。その値は、親ディレクトリの i ノード番号 (または、ディレクトリがルートディレクトリの場合は、それ自体の i ノード番号) に等しくなければなりません。

「.」と「..」ディレクトリの i ノード番号が不正であれば、fsck は正しい値に置き換えます。ディレクトリへのハードリンクが複数個あると、最初に見つかった方が「..」が指す実際の親であると見なされます。この場合、fsck は他の名前を削除するように促すプロンプトを表示します。

切り離されたディレクトリ

fsck プログラムは、ファイルシステム全体で参照関係をチェックします。ファイルシステムにリンクされていないディレクトリが見つかったら、fsck はそのディレクトリをファイルシステムの lost+found ディレクトリにリンクします (i ノードはファイルシステムに書き込まれてたが、それに対応するディレクトリデータブロックが書き込まれていないと、この状態が発生することがあります)。

通常データブロック

通常ファイルに関連付けられたデータブロックには、ファイルの内容が入っています。fsck は、通常ファイルのデータブロックの内容が有効かどうかはチェックしません。

自動ブートチェック機能の変更

ブート中には、ブートスクリプト /sbin/rcS を使用して、ハードディスクからマウントされるファイルシステムごとに予備チェックが実行されます。このスクリプトは、ルート (/) と /usr のファイルシステムをチェックします。他の rc シェルスクリプトが、fsck コマンドを使用して他のファイルシステムを順番にチェックします。ファイルシステムが並行にチェックされることはありません。

「fsck pass」の数值が1より大きい場合も、ブート中にはファイルシステムは順番にチェックされます。

/etc/vfstab ファイル

ファイルシステムを直接指定せずに、ファイルシステムをチェックしたりマウントするコマンドを実行すると、各コマンドは、ファイルシステムテーブル (/etc/vfstab) を調べ、各種フィールドで指定された情報を使用して処理を行います。fsck pass フィールドは、ファイルシステムチェックのための情報を指定します。「mount at boot」フィールドは、ブート時にファイルシステムをマウントするための情報を指定します。

新しいファイルシステムを作成する場合は、ブート時にチェックするかどうかとマウントするかどうかを示すエントリを /etc/vfstab に追加します。/etc/vfstab ファイルにエントリを追加する方法の詳細は、第 28 章を参照してください。

/etc/vfstab ファイル内の情報は、各システムのスライスとファイルシステムに固有です。次の例は、/etc/vfstab ファイルを示しています。

```
$ more /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot   options
#/dev/dsk/c1d0s2 /dev/rdisk/c1d0s2 /usr      ufs     1        yes      -
/proc        -           /proc     proc    -        no       -
fd           -           /dev/fd   fd      -        no       -
swap         -           /tmp      tmpfs   -        yes      -

/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /      ufs     1        no       -
/dev/dsk/c0t0d0s1 -           -         swap    -        no       -
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /usr     ufs     2        no       -
/dev/dsk/c0t0d0s7 /dev/rdisk/c0t0d0s7 /opt     ufs     3        yes      -
pluto:/export/svr4/man - /usr/man  nfs     no      yes      -
$
```

表 31-3 に、「fsck pass」フィールドの機能を示します。

表 31-3 「fsck pass」フィールド

| フィールドの設定 | 機能 | 備考 |
|----------------------|---|--|
| - (ハイフン) | ファイルシステムの状態に関係なく、汎用 fsck コマンドはファイルシステムをチェックしない。 | 読み取り専用ファイルシステム、リモートファイルシステム、/proc などの疑似ファイルシステムに対してはハイフンを指定してチェックを行わないように指示する。 |
| 0 以上 | ファイルシステム専用の fsck コマンドが呼び出される。 | UFS ファイルシステムに対して 0 が指定された場合、ファイルシステムはチェックされない。 |
| 1 以上かつ fsck -o p を使用 | ファイルシステム専用の fsck で、自動的に UFS ファイルシステムを並列にチェックする。 | この値には 1 より大きい任意の数値を使用できる。 |

preen モードでは、fsck は 1 ディスクごとに 1 つしか有効なファイルシステムをチェックできず、あるチェックが完了した後でなければ次のチェックを開始しません。fsck は、ファイルシステムが存在するデバイスのメジャー番号とマイナー番号を使用して、自動的に異なるディスク上のファイルシステムを同時にチェックする方法を決定します。

fsck pass 番号が 1 であれば、ファイルシステムは /etc/vfstab ファイルに表示される順番にチェックされます。通常、ルート (/) ファイルシステムの fsck pass は 1 に設定されます。

注 - fsck は、fsck pass 番号を使用せずにファイルシステムのチェック順を決定します。

▼ ブート中に実行される自動チェック機能を変更する方法

1. スーパーユーザーになります。
2. 「fsck pass」フィールド内の /etc/vfstab エントリを編集し、編集結果を保存します。
次にシステムをブートしたときに、新しい値が使用されます。

UFS ファイルシステムを対話式でチェックして修復する

次の場合には、ファイルシステムを対話式でチェックする必要があります。

- マウントできない場合
- 使用中に問題が発生する場合

使用中のシステムの整合性が失われると、コンソールウィンドウにエラーメッセージが表示されたり、システムがクラッシュしたりすることがあります。

`fsck` を使用する前に、552ページの「`fsck` コマンドの構文とオプション」と『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」を参照してください。

▼ ファイルシステムのチェックが必要かどうかを調べる方法

1. スーパーユーザーになります。
2. ファイルシステムをチェックします。

```
# fsck -m /dev/rdisk/device-name
```

このコマンドで指定したファイルシステムのスーパーブロック内の状態フラグがチェックされ、ファイルシステムがクリーンであるか、あるいはチェックする必要があるかどうか判断されます。

デバイス引数を省略すると、`/etc/vfstab` 内で「`fsck pass`」の値が 0 より大きいすべての UFS ファイルシステムがチェックされます。

例 — ファイルシステムのチェックが必要かどうかを調べる

次の例では、ファイルシステムのチェックが必要なことを示しています。

```
# fsck -m /dev/rdisk/c0t0d0s6
** /dev/rdisk/c0t0d0s6
```

(続く)

```
ufs fsck: sanity check: /dev/rdsk/c0t0d0s6 needs checking
```

▼ ファイルシステムを対話式でチェックする方法

1. スーパーユーザーになります。
2. ルート (/) と /usr 以外のローカルファイルシステムをマウント解除します。

```
# umountall -l
```

3. ファイルシステムをチェックします。

```
# fsck
```

/etc/vfstab ファイル内で、「fsck pass」フィールド内のエントリが 0 より大きいすべてのファイルシステムがチェックされます。また、fsck の引数として、マウントポイントディレクトリや /dev/rdsk/device-name も指定できます。整合性が失われている場合には、そのことを示すメッセージが表示されます。エラーメッセージのプロンプトに応答して 1 つまたは複数の UFS ファイルシステムを対話式でチェックする方法については、『Solaris のシステム管理 (第 2 巻)』の「ファイルシステムで発生する問題の解決」を参照してください。

4. エラーを修正し終わったら、fsck と入力して **Return** キーを押します。
fsck は、一度の実行ですべてのエラーを修正できないことがあります。「FILE SYSTEM STATE NOT SET TO OKAY」というメッセージが表示される場合は、fsck を使って、修正作業を繰り返します。fsck では修正できない場合は、551ページの「fsck で修復できない UFS ファイルシステムを修正する方法」を参照してください。
5. lost+found ディレクトリに保存されているファイルの名前を変更して移動します。
fsck によって lost+found ディレクトリに入れられた各ファイル名は、その i ノード番号を使用して変更されます。可能であれば、ファイル名を変更し、ファイルが含まれるディレクトリに移動してください。grep コマンドを使用し

て各ファイル中の語句を探したり、file コマンドを使用してファイルタイプを識別できる場合もあります。ディレクトリ全体が lost+found に書き出されている場合の方が、復帰先のディレクトリを調べて、移動することは容易です。

例 — ファイルシステムを対話方式でチェックする

次の例では、/dev/rdisk/c0t0d0s6 がチェックされ、不正なブロック数が訂正されます。

```
# fsck /dev/rdisk/c0t0d0s6
checkfileys: /dev/rdisk/c0t0d0s6
** Phase 1 - Check Block and Sizes
INCORRECT BLOCK COUNT I=2529 (6 should be 2)
CORRECT? y

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Cylinder Groups
929 files, 8928 used, 2851 free (75 frags, 347 blocks, 0.6%
fragmentation)
/dev/rdisk/c0t0d0s6 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
```

UFS ファイルシステムの修復

fsck の修復オプション (-o p) は、UFS ファイルシステムをチェックし、単純な問題を自動的に修正します。通常、この種の問題は予期しないシステム停止によるものです。オペレータの介入が必要な問題が発見されると、このコマンドは即座に終了します。また、修復オプションを使用する場合には、ファイルシステムを並列にチェックすることも可能です。

状態がクリーンにマークされずにシステムが停止した後のファイルシステムの修復にも、-o p オプションを指定して fsck を実行することができます。このモードでは、fsck はクリーンフラグを調べずに完全チェックを実行します。これらの処理は、fsck を対話形式で実行した場合の処理のサブセットです。

▼ ファイルシステムを修復する方法

1. スーパーユーザーになります。
2. ファイルシステムをマウント解除します。

```
# umount mount-point
```

3. 修復オプションを指定して **UFS** ファイルシステムをチェックします。

```
# fsck -o p /dev/rdisk/device-name
```

`fsck` の引数として `mount-point` または `/dev/rdisk/device-name` を使用すると、個々のファイルシステムを修復できます。

例 — ファイルシステムを修復する

次の例では、`/usr` ファイルシステムが修復されます。

```
# fsck -o p /usr
```

不正なスーパーブロックの復元

ファイルシステムのスーパーブロック内のデータが破壊された場合は、復元しなければなりません。スーパーブロックが不正なときには、`fsck` からメッセージが表示されます。幸い、スーパーブロックの冗長コピーがファイルシステム内に格納されています。`fsck -o b` を使用すると、スーパーブロックをそのコピーで置き換えることができます。

▼ 不正なスーパーブロックを復元する方法

1. スーパーユーザーになります。
2. データが破壊されたファイルシステム上のディレクトリがカレントディレクトリになっている場合は、カレントディレクトリを変更します。
3. ファイルシステムをマウント解除します。

```
# umount mount-point
```



注意 - 次の手順では、必ず `newfs` に `-N` オプションを使用してください。`-N` オプションを省略すると、新しい空のファイルシステムが作成されます。

4. `newfs -N` コマンドを使用して、スーパーブロックの値を表示します。

```
# newfs -N /dev/rdisk/device-name
```

このコマンドの出力には、`newfs` によってファイルシステムが作成されたときに、スーパーブロックのコピーとして使用されることになったブロック番号が表示されます。

5. `fsck` コマンドを使用して、代替スーパーブロックを指定します。

```
# fsck -F ufs -o b=block-number /dev/rdisk/device-name
```

`fsck` は、指定された代替スーパーブロックを使用して、一次スーパーブロックを復元します。いつでも代替ブロックとして 32 を試すことができます。また、`newfs -N` で表示された代替ブロックを使用することもできます。

例 — 不正なスーパーブロックを復元する

次の例では、`/files7` ファイルシステムのスーパーブロックがブロック番号 5264 に存在するコピーを使って復元されます。

```
# cd /
# umount /files7
# newfs -N /dev/rdisk/c0t3d0s7
/dev/rdisk/c0t3d0s7: 163944 sectors in 506 cylinders of 9 tracks, 36 sectors
83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
super-block backups (for fsck -b #) at:
 32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,
47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,
93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
140528, 145760, 150992, 156224, 161456,
# fsck -F ufs -o b=5264 /dev/rdisk/c0t3d0s7
Alternate superblock location: 5264.
** /dev/rdisk/c0t3d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
```

(続く)


```
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)
/dev/rdsk/c0t3d0s7 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
#
```

fsck で修復できない UFS ファイルシステムを修正する方法

あるパスで問題が訂正されたために、前のパスで見つからなかった問題が表面化する場合があるので、fsck を何度か実行してファイルシステムを修正しなければならないことがあります。fsck はクリーンになるまで動作を続けるわけではないので、手作業で実行しなければなりません。

fsck で表示される情報に注目してください。問題を解決する上で参考になります。たとえば、メッセージは不正なディレクトリを指す場合があります。そのディレクトリを削除すると、fsck が問題なく実行されるようになる場合もあります。

それでも fsck でファイルシステムを修復できない場合は、fsdb、ff、clri、または ncheck コマンドを使用し、間違いを指定して修正します。これらのコマンドの使用方法については、fsdb(1M)、ff(1M)、clri(1M)、ncheck(1M) の各マニュアルページを参照してください。最終的には、ファイルシステムを作成し直し、その内容をバックアップ媒体から復元せざるを得ない場合があります。ファイルシステム全体を復元する方法については、第 35 章を参照してください。

ファイルシステムを完全に修復できないが、読み取り専用としてマウントできる場合は、cp、tar、または cpio を使用して、データのすべてまたは一部をファイルシステムから取り出してください。

問題の原因がハードウェア上のディスクエラーであれば、ファイルシステムを作成し直して復元する前に、ディスクをフォーマットし直して再びスライスに分割しなければならない場合があります。一般に、ハードウェアエラーが発生すると、さまざまなコマンドで同じエラーが繰り返し表示されます。format(1M) コマンドはディスク上の不良ブロックを使用しないようにします。ただし、ディスクの損傷が致命的な場合、フォーマットし直した後も問題が解決されないことがあります。format コマンドの使用方法については、format(1M) のマニュアルページを

参照してください。新しいディスクをインストールする方法については、第 23 章または第 24 章を参照してください。

fsck コマンドの構文とオプション

fsck コマンドは、ファイルシステム内の非整合状態をチェックして修復します。次の 4 つのオプションがあります。

- ファイルシステムがマウント可能かどうかのチェックだけを行う (fsck -m)
- 修復する前に確認を促すプロンプトを表示する (fsck)
- すべての修復時のプロンプトに yes または no の応答が指定されたものとして処理を行う (fsck -y)
- 確認を促すプロンプトを表示することなくファイルシステムを修復し、想定される (軽微な) 非整合箇所をすべて修復 (preen) するが、重大な問題にぶつかると終了する (fsck -o p)

汎用 fsck コマンドの構文、オプション、引数

fsck には、汎用コマンドとファイルシステム専用の 2 種類のコマンドがあります。汎用コマンドはほとんどのタイプのファイルシステムに使用でき、専用のコマンドは特定のタイプのファイルシステムにのみ使用できます。通常は汎用コマンドを使用してください。汎用コマンドは必要に応じてファイルシステム専用のコマンドを呼び出します。

通常、fsck を実行するにはスーパーユーザーにならなければなりません。スーパーユーザーにならなくても fsck コマンドの実行はできますが、修復するにはファイルシステムをマウント解除しなければなりません。また、スライスの raw デバイスファイルの読み取り権を持っていないければなりません (セキュリティ上の問題のため、一般のユーザーには raw デバイスファイルの読み取り権はありません)。

汎用 fsck コマンドは、/etc/vfstab にアクセスして、どのファイルシステムをチェックするかを調べます。fsck pass 番号に - (ハイフン) が指定されている任意のファイルシステムと、0 が指定されている UFS のファイルシステムを除き、指定された各ファイルシステム上で、該当するファイルシステム専用の fsck コマンドを実行します。

汎用 fsck コマンドの構文は次のとおりです。

```

/usr/sbin/fsck [-F type] [-V] [-m] [special]
/usr/sbin/fsck [-F type] [-V] -[y|Y]|[n|N] [-o specific-options] [special]

```

表 31-4 は、汎用 fsck コマンドのオプションと引数を示しています。

表 31-4 fsck コマンドのオプションと引数

| オプションのタイプ | オプション | 説明 |
|-----------|-----------------|---|
| 汎用 | -F | ファイルシステムのタイプ (type) を指定する。コマンド行で type を指定しない場合は、/etc/vfstab ファイル内のエントリを、指定した raw デバイス名と照合して取得される。エントリが見つからない場合は、/etc/default/fs 内で指定されたデフォルトのローカルのファイルシステムのタイプが使用される。 |
| | -v | 実行される完全なコマンド行が表示される (詳細モード)。表示行には、/etc/vfstab から取り出された追加情報が含まれる。このオプションを使用すると、コマンド行を検査して有効性を確認できる。コマンド自体は実行されない。 |
| | -m | 予備チェックのみを実行する。ファイルシステムの状態を示すコードを返す。0 は「クリーン」を示し、32 は「ダーティ」を示す。起動スクリプト /sbin/rcS は、このオプションを使用して、ファイルシステムのチェックが必要かどうかを判断する。 |
| | -y、-Y、-n、または -N | 実行されるコマンドのすべてのプロンプトに対して、自動的に、yes または no で応答する。 |

表 31-4 fsck コマンドのオプションと引数 続く

| オプションのタイプ | オプション | 説明 |
|-----------|----------------|--|
| | c | <p>静的にテーブルが割り当てられている古い形式のファイルシステムを、新しい形式の動的に割り当てられたテーブルに変換する。静的割り当ては最大テーブルサイズにハード制限が適用されることを示し、動的割り当ては、初期割り当ての後で必要に応じてテーブル用の領域を追加できることを意味する。ファイルシステムが新しい形式であれば、テーブル割り当てが古い形式で許される固定最大サイズを超えない限り、古い形式に変換される。fsck は変換方法を表示する。対話モードでは、fsck は変換前に確認を促すプロンプトを表示する。-o p オプションを使用すると、確認を促すプロンプトは表示されずに変換試行が実行される。このオプションは、多数のファイルシステムを一度に変換したい場合に便利である。ファイルシステムが古い形式か新しい形式かは、fstyp (1M) コマンドを実行し、最初に表示される行を調べれば判断できる。</p> |
| | w | <p>書き込みアクセスできるファイルシステムのみをチェックする。</p> |
| | <i>special</i> | <p>1 つまたは複数のファイルシステムのマウントポイントまたは raw デバイス名を指定する。マウントポイントのエントリは、/etc/vfstab 内になければならない。<i>special</i> 引数を省略すると、/etc/vfstab 内で fsck デバイスが指定されていて、fsck pass の値が 0 より大きいエントリがチェックされる。修復 (-o p) が有効で、fsck pass の値が 1 より大きいエントリが複数個あると、異なるディスク上のファイルシステムは並列してチェックされる。</p> |
| 専用 | | <p>-o オプションに続けて、オプションをコマンド区切ったリスト。UFS 固有の fsck コマンドが解釈できるように渡されるオプションを指定する。</p> |

表 31-4 fsck コマンドのオプションと引数 続く

| オプションのタイプ | オプション | 説明 |
|-----------|---------------|---|
| | p | 修復。コマンドはプロンプトを表示せずに実行され、検出されたエラーは自動的に訂正されるが、オペレータの介入が必要な問題が見つかったら終了する。このオプションを使用すると、UFS ファイルシステムを並列にチェックすることもできる。 |
| | b=blocknumber | 指定した位置にある代替 (冗長) スーパーブロックを使用する。このオプションを使用すると、不良スーパーブロックを修復できる。newfs -N コマンドを使用すると、代替スーパーブロックのリストを表示できる。 |

ファイルシステムの参照情報

この章の内容は次のとおりです。

- 557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」
- 表 32-4
- 565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」
- 569ページの「カスタムファイルシステムパラメータの決定」
- 573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

Solaris 2.5 から、プラットフォームに依存するカーネルモジュールとコマンドは新しい場所に移動されました。/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 32-4 を参照してください。

表 32-1 および表 32-2 に、ルート (/) のデフォルトファイルシステム中のディレクトリを示します。表 32-3 に、/usr のデフォルトファイルシステム中のディレクトリを示します。

表 32-1 ルート (/) ファイルシステムのデフォルトディレクトリ (その 1)

| ディレクトリ | 説明 |
|--------------|---|
| / | ファイルシステムの名前空間全体のルート |
| /dev | 特殊ファイルの一次位置 |
| /dev/dsk | ブロックディスクデバイス |
| /dev/pts | pty スレーブデバイス |
| /dev/rdsk | 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/term | 端末デバイス |
| /etc | ホスト固有のシステム管理構成ファイルとデータベース |
| /etc/acct | アカウントシステム構成情報の構成情報 |
| /etc/cron.d | cron の構成情報 |
| /etc/default | 各種プログラムのデフォルト情報 |
| /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/fs | /usr をマウントする前に必要な処理のためにファイルシステムタイプ別に編成されたバイナリ |
| /etc/inet | インターネットサービスの構成ファイル |
| /etc/init.d | 実行レベルを切り替えるためのスクリプト |
| /etc/lp | プリンタサブシステムの構成情報 |
| /etc/mail | メールサブシステムの構成 |

表 32-1 ルート (/) ファイルシステムのデフォルトディレクトリ (その 1) 続く

| ディレクトリ | 説明 |
|-------------|---|
| /etc/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/opt | オプションパッケージの構成情報 |
| /etc/rc0.d | 実行レベル 0 を開始/停止した時に起動されるスクリプト |
| /etc/rc1.d | 実行レベル 1 を開始/停止した時に起動されるスクリプト |
| /etc/rc2.d | 実行レベル 2 を開始/停止した時に起動されるスクリプト |
| /etc/rc3.d | 実行レベル 3 を開始/停止した時に起動されるスクリプト |
| /etc/rcS.d | システムを単一ユーザーモードで起動するスクリプト |
| /etc/saf | サービスアクセス機能ファイル (FIFO など) |
| /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/sm | 状態モニター情報 |
| /etc/sm.bak | 状態モニター情報のバックアップコピー |
| /etc/tm | 商標ファイル。内容はブート時に表示 |
| /etc/uucp | uucp 構成情報 |
| /export | エクスポートされるファイルシステムツリーのデフォルトのルート |
| /home | ユーザー命令用のサブツリーのデフォルトのルート |
| /kernel | ブートプロセスの一部として必要なプラットフォーム独立型のロード可能カーネルモジュールのサブツリー。プラットフォーム独立型のコアカーネル /kernel/genunix の汎用部分が含まれる。/platform と /usr/platform ディレクトリ構造については、表 32-4 を参照 |

表 32-2 ルート (/) ファイルシステムのデフォルトディレクトリ (その 2)

| ディレクトリ | 説明 |
|---------------|---|
| /mnt | ファイルシステムの一般的な一次マウントポイント |
| /opt | 追加アプリケーションパッケージ用のサブツリーのルート |
| /opt/SUNWspro | バンドルされていない言語製品のマウント/インストールポイント |
| /sbin | ブートプロセスと手作業によるシステム障害の回復に使用される重要な実行可能プログラム |
| /stand | スタンドアロンプログラム |
| /tmp | 一次ファイル。ブートシーケンス中に消去される |
| /usr | /usr ファイルシステムのマウントポイント |
| /var | 各種ファイルのサブツリーのルート |
| /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/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/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 (注: USENET 方式のニュースとは異なる) |
| /var/nis | NIS+ データベース |
| /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 |

表 32-2 ルート (/) ファイルシステムのデフォルトディレクトリ (その 2) 続く

| ディレクトリ | 説明 |
|------------------------------------|--|
| <code>/var/sadm</code> | 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 |
| <code>/var/saf</code> | saf (サービスアクセス機能) のログファイルとアカウントングファイル |
| <code>/var/spool</code> | スプール化された一時ファイルのディレクトリ |
| <code>/var/spool/cron</code> | cron と at のスプールファイル |
| <code>/var/spool/locks</code> | スプールロックファイル |
| <code>/var/spool/lp</code> | ラインプリンタのスプールファイル |
| <code>/var/spool/mqueue</code> | 配信用に待ち行列に入れられたメール |
| <code>/var/spool/pkg</code> | スプール化されたパッケージ |
| <code>/var/spool/uucp</code> | 待ち行列化された uucp のジョブ |
| <code>/var/spool/uucppublic</code> | uucp によって格納されるファイル |
| <code>/var/tmp</code> | 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 |
| <code>/var/uucp</code> | uucp のログファイルと状態ファイル |
| <code>/var/yp</code> | NIS データベース (NIS との下位互換性を保つため。NIS+ への移行が完了した後は不要) |

表 32-3 /usr ファイルシステムのデフォルトディレクトリ

| ディレクトリ | 説明 |
|-------------------|---------------|
| <code>bin</code> | 標準システムコマンドの位置 |
| <code>demo</code> | デモプログラムとデータ |

表 32-3 /usr ファイルシステムのデフォルトディレクトリ 続く

| ディレクトリ | 説明 |
|------------|---|
| games | 空のディレクトリ。SunOS 4.0/4.1 ソフトウェアで使用されていた |
| include | ヘッダファイル (C プログラム用など) |
| kernel | その他モジュール |
| kvm | 実装されたアーキテクチャ固有のバイナリとライブラリ |
| lib | 各種プログラムのライブラリ、アーキテクチャ依存データベース、またはユーザーが直接呼び出さないバイナリ |
| lib/acct | アカウントिंगスクリプトとバイナリ |
| lib/class | スケジューラ固有のディレクトリ。priocntl コマンドと dispadm コマンドの実行可能プログラムが入っている |
| lib/font | troff フォント記述ファイル |
| lib/fs | ファイルシステムタイプ依存モジュール。ユーザーは直接呼び出さない |
| lib/iconv | iconv(1) の変換テーブル |
| lib/libp | プロファイルライブラリ |
| lib/locale | 各国対応のローカライズデータベース |
| lib/lp | ラインプリンタサブシステムのデータベースとバックエンドの実行可能プログラム |
| lib/mail | mail サブシステムの補助プログラム |
| lib/netsh | インターネットネットワークサービス |
| lib/nfs | NFS 関連の補助プログラムとデーモン |
| lib/pics | 実行時リンカの構築に必要な PIC アーカイブ |

表 32-3 /usr ファイルシステムのデフォルトディレクトリ 続く

| ディレクトリ | 説明 |
|---------------------|---|
| lib/refer | 参照関連の補助プログラム |
| lib/sa | システム活動レポートパッケージ用のスクリプトとコマンド |
| lib/saf | サービスアクセス機能に関連する補助プログラムとデーモン |
| lib/sparcv9 | 64 ビット Solaris ライブラリ |
| lib/uucp | uucp 関連の補助プログラムとデーモン |
| lib/zoneinfo | 時間帯情報 |
| local | サイトのローカルコマンド |
| old | 段階的に使用されなくなっているプログラム |
| openwin | OpenWindows ソフトウェアのマウント/インストールポイント |
| 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。以下の各サブディレクトリを参照 |
| sadm/bin | FMLI スクリプトに使用する「valtools」境界 |
| sadm/install | pkg 管理用の実行プログラムとスクリプト |
| sbin | システム管理用の実行可能プログラム |
| sbin/static | /usr/bin と /usr/sbin から選択したプログラムの静的リンクバージョン |
| share | アーキテクチャに依存しない共有可能ファイル |
| share/lib | アーキテクチャに依存しないデータベース |
| share/lib/keytables | キーボード配置記述ファイル |
| share/lib/mailx | mailx 関連のヘルプファイル |

表 32-3 /usr ファイルシステムのデフォルトディレクトリ 続く

| ディレクトリ | 説明 |
|--------------------|---------------------------|
| share/lib/nterm | nroff 端末テーブル |
| share/lib/pub | 各種データファイル |
| share/lib/spell | spell 関連の補助データベースとスクリプト |
| share/lib/tabset | タブ設定のエスケープシーケンス |
| share/lib/terminfo | terminfo スタイルの端末記述ファイル |
| share/lib/tmac | [nt]roff マクロパッケージ |
| 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 |
| ucb | UCB 互換パッケージのバイナリ |
| ucbinclude | UCB 互換パッケージのヘッダファイル |
| ucbllib | UCB 互換パッケージのライブラリ |

プラットフォームに依存するディレクトリ

表 32-4 に、/platform ディレクトリと /usr/platform ディレクトリに入っているすべてのプラットフォームに依存するオブジェクトを示します。

表 32-4 /platform と /usr/platform ディレクトリ

| ディレクトリ | 説明 |
|---------------------|--|
| /platform | ルート (/) ファイルシステムに存在すべき一連のディレクトリが、サポートされるプラットフォームごとに入っている。 |
| /platform/*/kernel | プラットフォームに依存するカーネル構成要素が入っている。プラットフォームに依存するコアカーネルであるファイル <code>unix</code> も含む。kernel (1M) のマニュアルページを参照。 |
| /usr/platform | ルート (/) ファイルシステムに存在する必要がない、プラットフォームに依存するオブジェクトが入っている。削除された <code>/usr/kvm</code> の内容に置き換わるオブジェクトを含む。 |
| /usr/platform/*/lib | <code>/usr/lib</code> ディレクトリ中のオブジェクトに類似した、プラットフォームに依存するオブジェクトが入っている。 |
| /platform/*/sbin | <code>/usr/sbin</code> 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。

- ブートブロック – システムのブート時に使用される情報が格納される。
- 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が格納される。
- **i** ノード – ファイルに関する名前以外のすべての情報が格納される。
- 記憶域またはデータブロック – 各ファイルのデータが格納される。

この節では、これらのブロックの編成と機能について説明します。

ブートブロック

ブートブロックには、システムのブート時に使用されるプログラムが格納されます。ファイルシステムがブートに使用されなければ、ブートブロックは空白のままです。ブートブロックは最初のシリンダグループ (シリンダグループ 0) にのみ表示され、スライス内の最初の 8K バイトです。

スーパーブロック

スーパーブロックには、ファイルシステムに関する大部分の情報が格納されます。その中でも、特に次の情報が重要です。

- ファイルシステムのサイズと状態
- ラベル (ファイルシステム名とボリューム名)
- ファイルシステムの論理ブロックのサイズ
- 最終更新日時
- シリンダグループのサイズ
- シリンダグループ内のデータブロック数
- 集計データブロック
- ファイルシステムの状態: クリーン、安定、または有効
- 最後のマウントポイントのパス名

スーパーブロックは、ディスクスライスの先頭にあり、各シリンダグループ内で複製されます。スーパーブロックには重要なデータが入っているので、ファイルシステムの作成時には複数のスーパーブロックが作成されます。各スーパーブロックの複製は、シリンダグループの先頭からさまざまな大きさだけオフセットされます。複数プラッタを持つディスクドライブの場合、オフセットはスーパーブロックがドライブの各プラッタに表示されるように計算されます。つまり、最初のプラッタが失われても、いつでも代替スーパーブロックを取り出せます。最初のシリンダグループ内の先行ブロックを除き、オフセットによって作成される先行ブロックがデータの格納に使用されます。

集計情報ブロックは、スーパーブロックといっしょに保管されます。複製されませんが、通常はシリンダグループ 0 内で最初のスーパーブロックといっしょにグループ化されます。集計ブロックレコードには、ファイルシステムの使用時に発生した変化が記録され、ファイルシステム内の i ノード数、ディレクトリ数、フラグメント数、および記憶ブロック数が表示されます。

i ノード

i ノードには、ファイルに関して名前以外のすべての情報が入っており、ディレクトリ内に保管されます。i ノードは 128 バイトです。i ノード情報はシリンダ情報ブロック内に保管され、次の情報が入っています。

- ファイルのタイプ
 - 通常ファイル
 - ディレクトリ
 - ブロック型特殊ファイル
 - キャラクタ型特殊ファイル
 - シンボリックリンク
 - FIFO (名前付きパイプとも呼ばれます)
 - ソケット
- ファイルのモード (読み込み権-書き込み権-実行権のセット)
- ファイルへのハードリンク数
- ファイルの所有者のユーザー ID
- ファイルが属するグループ ID
- ファイル内のバイト数
- 15 個のディスクブロックアドレスの配列
- ファイルの最終アクセス日時
- ファイルの最終変更日時
- ファイルの作成日時

15 個のディスクアドレス (0 から 14 まで) の配列は、ファイルの内容が格納されるデータブロックを指します。最初の 12 個は直接アドレスで、ファイルの内容のうち最初の 12 個の論理記憶ブロックを直接指します。ファイルが論理ブロック 12 個分より大きい場合は、13 番目のアドレスは間接ブロックを指します。間接ブロックには、ファイルの内容ではなく直接ブロックのアドレスが入っています。14 番目のアドレスは、二重間接ブロックを指します。二重間接ブロックには、間接ブロッ

クのアドレスが入っています。15 番目のアドレスが必要な場合は、三重間接アドレスが格納されます。図 32-1 は、i ノードから始まって、このアドレスブロックチェーンを示しています。

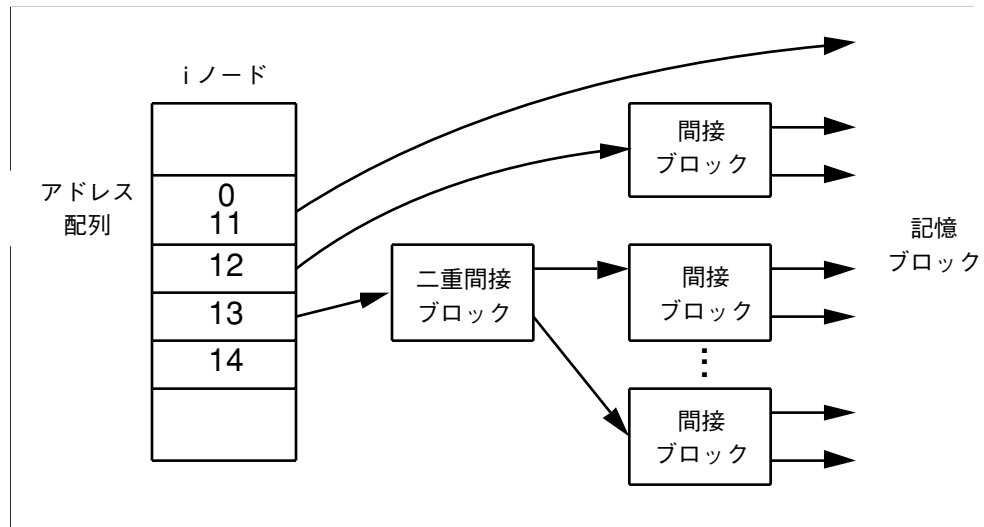


図 32-1 UFS システム内のファイルシステムアドレスチェーン

データブロック

ファイルシステムに割り当てられた残りの領域には、データブロックが入っています。この種のブロックは、記憶ブロックと呼ばれることもあります。これらのデータブロックのサイズは、ファイルシステムの作成時に決定されます。デフォルトでは、データブロックは 2 つのサイズ、つまり 8K バイトの論理ブロックサイズと 1K バイトのフラグメントサイズで割り当てられます。

通常ファイルの場合、データブロックにはファイルの内容が入っています。ディレクトリの場合、データブロックにはディレクトリ内のファイルの i ノード番号とファイル名を示すエントリが入っています。

空きブロック

現在、i ノード、間接アドレスブロック、または記憶ブロックとして使用されていないブロックには、シリンダグループマップ内で空きを示すマークが付けられます。また、このマップはフラグメントを追跡し、ディスク性能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、図 32-2 に一般的な UFS システム内の一連のシリンダグループを示しています。

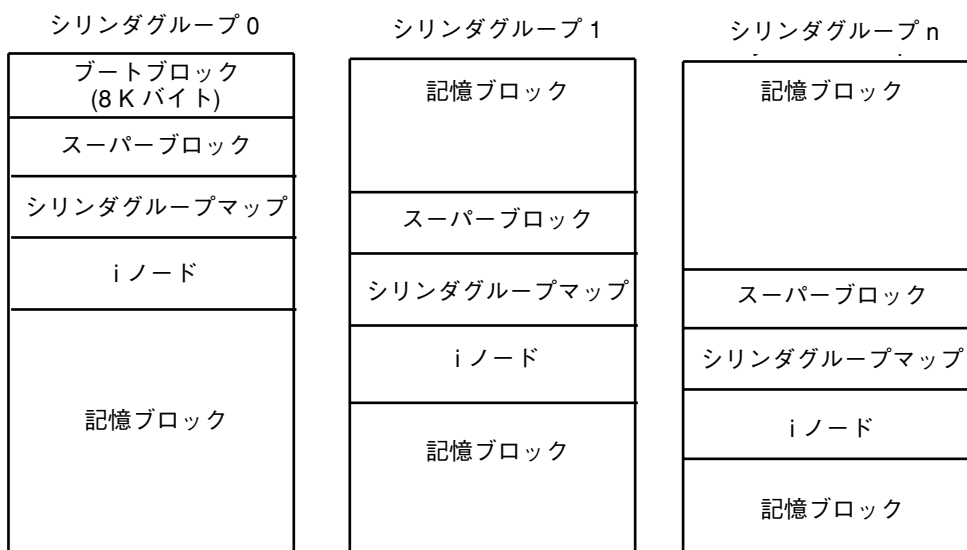


図 32-2 一般的な UFS ファイルシステム

カスタムファイルシステムパラメータの決定

`newfs` コマンドによって割り当てられるデフォルトのファイルシステムパラメータを変更しようとする前に、各パラメータについて理解しておく必要があります。この節では、次の各パラメータについて説明します。

- ブロックサイズ
- フラグメントサイズ
- 最小空き領域
- 回転の遅れ
- 最適化タイプ
- i ノード数

論理ブロックサイズ

論理ブロックサイズは、UNIX カーネルがファイルの読み書きに使用するブロックのサイズです。一般に、論理ブロックサイズは物理ブロックサイズ (通常は 512K バイト) とは異なります。物理ブロックサイズは、ディスクコントローラが読み書きできる最小ブロックのサイズです。

ファイルシステムの論理ブロックサイズを指定できます。ファイルシステムの作成後は、ファイルシステムを再構築しなければ、このパラメータを変更できません。論理ブロックサイズの異なるファイルシステムを、同じディスクに格納できます。

デフォルトでは、UFS ファイルシステムの論理ブロックサイズは 8192 バイト (8K バイト) です。UFS ファイルシステムでは、ブロックサイズとして 4096 バイトまたは 8192 バイト (4K または 8K バイト) がサポートされます。8K バイトは、論理ブロックの推奨サイズです。

システムに最善の論理ブロックサイズを選択するには、必要な性能と使用可能容量を検討してください。ほとんどの UFS システムの場合は、8K バイトのファイルシステムが最高の性能を発揮し、ディスク性能と一次メモリーやディスク上の領域の使用量が適切なバランスに保たれます。

原則として、効率を高めるには、ほとんどのファイルがきわめて大きいファイルシステムには大きめの論理ブロックサイズを使用します。ほとんどのファイルがきわめて小さいファイルシステムには、小さめの論理ブロックサイズを使用します。ファイルシステム上で `quot -c file-system` コマンドを使用すると、ファイルの分散に関する詳細なレポートをブロックサイズ別に表示できます。

フラグメントサイズ

ファイルが作成または拡張されると、論理ブロック全体または「フラグメント」と呼ばれる部分のディスク容量が割り当てられます。ファイルのデータを保持するディスク容量が必要になると、まずブロック全体が割り当てられ、次に残りの部分にブロックのうち 1 つまたは複数のフラグメントが割り当てられます。小型ファイルの場合、割り当てはフラグメントから始まります。

ブロック全体ではなく、そのフラグメントを割り当てることができるので、ブロック内の未使用のホールによってディスク容量の「フラグメント」が低下し、容量の節約になります。

UFS ファイルシステムを作成するときに、「フラグメントサイズ」を定義します。デフォルトのフラグメントサイズは 1K バイトです。各ブロックは、1 個、2 個、4

個、または 8 個のフラグメントに分割できます。この場合、フラグメントサイズは 8192 バイトから 512K バイトまでです (4K バイトのファイルシステムのみ)。実際には、下限はディスクのセクターサイズ、通常は 512 バイトに連動します。

注 - 上限を、まったくフラグメントのない場合の完全ブロックのサイズに等しくすることができます。容量よりも速度を重視する場合、きわめて大型のファイルがあるファイルシステムには、この構成が最適なことがあります。

フラグメントサイズを選択するときには、処理時間と容量を取捨選択してください。フラグメントサイズが小さければ容量の節約になりますが、割り当てには時間がかかります。原則として、格納効率を高めるには、ほとんどのファイルが大型のファイルシステムには、大きめのフラグメントサイズを使用します。ほとんどのファイルが小型のファイルシステムには、小さめのフラグメントサイズを使用します。

最小空き容量

「最小空き容量」とは、ファイルシステムの作成時に予約分として保持されるディスク容量です。デフォルトの予約分は、 $((64\text{M バイト}/\text{パーティションサイズ}) * 100)$ で算出し、その値を最も近い整数に切り捨てます。値は、ディスク容量の 1% から 10% の範囲に制限されます。ファイルシステム内の空き容量が少なくなるほど、アクセス速度が低下するので、空き容量は重要です。十分な空き容量があれば、UFS ファイルシステムは効率よく動作します。ファイルシステムがいっぱいになって、使用可能なユーザー領域を使い果たすと、スーパーユーザー以外は予約済みの空き容量にアクセスできなくなります。

df などのコマンドは、最小空き容量として割り当て済みの分を差し引いて、ユーザーに使用可能な容量をパーセントで表示します。コマンドでファイルシステム内のディスク容量の 100 パーセント以上が使用中であると表示される場合は、予約分の一部がルートに使用されています。

ユーザーに制限を適用する場合に、各ユーザーが使用可能な容量には予約分の空き容量は含まれません。tuneufs コマンドを使用すると、既存のファイルシステムの最小空き容量の値を変更できます。

回転の遅れ (ギャップ)

「回転の遅れ」は、CPU がデータ転送を完了し、同じディスクシリンダ上で次のデータ転送を開始するまでに予想される最小所要時間 (ミリ秒単位) です。デフォル

トの遅れは、ディスクのタイプによって異なり、通常はディスクタイプごとに最適化されています。

ファイルに書き込むときに、UFS 割り当てルーチンは新しいブロックを同じファイル内の直前のブロックと同じディスクシリンダ上に配置しようとします。また、新しいブロックをトラック内で最適の位置に配置して、そこへのアクセスに必要なディスクの回転を最小限度に抑えようとします。

ファイルブロックを「回転して適切に動作」するように配置するには、割り当てルーチンは CPU による転送処理速度と、ディスクが 1 ブロックをスキップする所要時間を認識しなければなりません。mkfs コマンドのオプションを使用すると、ディスクの回転速度と 1 トラック当たりのディスクブロック (セクター) 数を指定できます。割り当てルーチンは、この情報を使用して、1 ディスクブロックをスキップするミリ秒数を求めます。次に、割り当てルーチンは予想転送時間 (回転の遅れ) を使用して、システムの読み込み準備ができたときに次のブロックがディスクヘッドの真下にくるようにブロックを配置します。

注 - 回転の遅れ (newfs の `-d` オプション) を指定しなくてもよいデバイスがあります。

各ブロックは、システムが同じディスクの回転中に読み込める処理速度の場合のみ、連続して配置されます。システムが低速であれば、ディスクはファイル内の次のブロックの先頭を通り過ぎてしまうので、そのブロックを読み込むには、もう 1 回転しなければならず、長時間かかります。次のディスク要求が発生するときに該当するブロックにヘッドがきているように、ギャップに適切な値を指定してください。

既存のファイルシステムの場合は、tunefs コマンドを使用してこのパラメータの値を変更できます。変更結果は、それ以後のブロック割り当てにのみ適用され、すでに割り当て済みのブロックには適用されません。

最適化のタイプ

「最適化のタイプ」には、「領域」と「時間」があります。

- 領域 - 領域の最適化を選択すると、フラグメントを最小限度に抑え、ディスクの使用状況が最適化されるようにディスクブロックが割り当てられる。
- 時間 - 時間の最適化を選択すると、配置はあまり重視されず、できるだけ高速になるようにディスクブロックが割り当てられる。十分な空き領域があれば、それほど細かく分割しなくても、比較的簡単にディスクブロックを効率よく割り当てることができる。デフォルトは「時間」です。

既存のファイルシステムの場合は、`tunefs` コマンドを使用して最適化タイプのパラメータ値を変更できます。

ファイルの数

`i` ノード数によって、ファイルシステム内で保持できるファイル数が決まります。ファイルごとに `i` ノードが 1 つあります。`i` ノード 1 個あたりのバイト数によって、ファイルシステムの作成時に作成される合計 `i` ノード数が決まります。これは、ファイルシステムの合計サイズを、`i` ノード 1 個あたりのバイト数で割った値です。`i` ノードが割り当てられたら、ファイルシステムを作成し直さないかぎり、その数は変更できません。

`i` ノード 1 個あたりのデフォルトのバイト数は 2048 バイト (2K バイト) で、これは各ファイルの平均サイズが 2K バイト以上であることを想定しています。ほとんどのファイルは、2K バイトを超えています。多数のシンボリックリンクを持つファイルシステムでは、平均ファイルサイズを小さくすることができます。ファイルシステムに多数の小型ファイルが格納される場合は、このパラメータに小さい値を与えてもかまいません。ただし、`i` ノード数が少ないために `i` ノードが不足するよりも、多すぎる方が好ましいことを留意してください。`i` ノード数が少なすぎると、実際には空のディスクスライス上でも最大ファイル数に達してしまうことがあります。

カスタマイズされたファイルシステムを作成するためのコマンド

この節では、カスタマイズされたファイルシステムの作成に使用する 2 つのコマンドについて説明します。

- `newfs`
- `mkfs`

`newfs` コマンドの構文、オプション、引数

`newfs` コマンドは、ファイルシステムの作成に使用する `mkfs` コマンドの簡便バージョンです。`newfs` コマンドは、`/usr/sbin` ディレクトリに入っています。

構文は次のとおりです。

```
newfs [-Nv] [mkfs_options] raw_device
```

表 32-5 に、newfs コマンドのオプションと引数を示します。

表 32-5 newfs コマンドのオプションと引数

| オプション | 説明 |
|--------------|---|
| N | ファイルシステムの作成に使用されるファイルシステムパラメータが表示されるが、実際には作成されない。このオプションでは、既存のファイルシステムの作成に使用されたパラメータは表示されない。 |
| -v | mkfs コマンドに渡されるパラメータが表示され、-N オプションを指定しなければファイルシステムが作成される。 |
| mkfs-options | 後続のオプションを使用して、mkfs コマンドに渡されるパラメータが設定される。次のオプションは、mkfs に渡される順番に記述されている。各オプションは、先行キーワードを付けずに空白で区切る。 |
| -s size | ファイルシステムのセクター数。デフォルトは、ディスクラベルから自動的に判別される。 |
| -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判別される。 |
| -b bsize | データ転送に使用される論理ブロックのバイト数。サイズとして 4096 または 8192 バイト (4K または 8K バイト) を指定する。デフォルトは 8192 バイト (8K バイト)。 |
| -f fragsize | ファイルに割り当てられるディスク容量の最小バイト数。フラグメントサイズを、512 バイトから 8192 バイトまでの 2 の乗数単位で指定する。デフォルトは 1024 バイト (1K バイト)。 |
| -c cgsize | 1 シリンダグループあたりのディスクシリンダ数。この数値は 1 から 32 までの範囲内であればならない。デフォルトは 16。 |
| -m free | 空きディスク領域の最小許容率。デフォルトの予約分は、 $((64\text{M バイト} / \text{パーティションサイズ}) * 100)$ で算出した値を最も近い整数に切り捨てます。値は、ディスク容量の 1% から 10% の範囲に制限されます。 |
| -r rpm | 1 分当たりのディスクの回転速度。デフォルトは 3600 である。このパラメータは、mkfs に渡される前に 1 秒当たりの回転数に変換される。 |

表 32-5 newfs コマンドのオプションと引数 続く

| オプション | 説明 |
|---------------------|--|
| -i <i>nbpi</i> | 作成できる <i>i</i> ノードの計算に使用される <i>i</i> ノード 1 個当たりのバイト数。デフォルトは 2048。 |
| -o <i>opt</i> | ディスクブロックをファイルに割り当てるときに使用される最適化のタイプ。 <i>opt</i> には <i>time</i> または <i>space</i> を指定する。デフォルトは <i>time</i> です。 |
| -a <i>apc</i> | 不良ブロックを配置するために予約される 1 ディスクシリンダ (SCSI デバイスのみ) の代替ブロック数。デフォルトは 0。 |
| -d <i>gap</i> | (回転の遅れ) CPU がデータ転送を完了し、同じディスクシリンダ上で次のデータ転送を開始するまでに予想される最小ミリ秒数。デフォルトは 4。 |
| -d <i>nrpos</i> | シリンダグループを分割するさまざまな回転位置の数。デフォルトは 8。 |
| -C <i>maxcontig</i> | あるファイルに属し、回転の遅れが挿入される前に連続して割り当てられる最大ブロック数。デフォルトはドライブごとに異なる。内部 (トラック) バッファを持たないドライブ (または、内部バッファの存在をうたっていないドライブ/コントローラ) の場合は、デフォルトは 1 で、バッファを持つドライブの場合はデフォルトは 7。 このパラメータには、 <i>maxphys</i> は、入出力サブシステムが満たせる最大ブロック転送サイズ (バイト数) を指定する読み込み専用のカーネル変数である (この制限は、 <i>newfs</i> や <i>mkfs</i> ではなく <i>mount</i> によって適用される)。 また、このパラメータはクラスタ化も制御する。 <i>rotdelay</i> の値に関係なく、 <i>maxcontig</i> が 1 より大きいときのみクラスタ化できる。クラスタ化すると、入出力が高速になる。詳細は、 <i>tunefs(1M)</i> のマニュアルページを参照。 |
| <i>raw_device</i> | ファイルシステムを入れるパーティションの特殊文字 (<i>raw</i>) デバイスファイル名。この引数は必須。 |

例 — newfs コマンドのオプションと引数

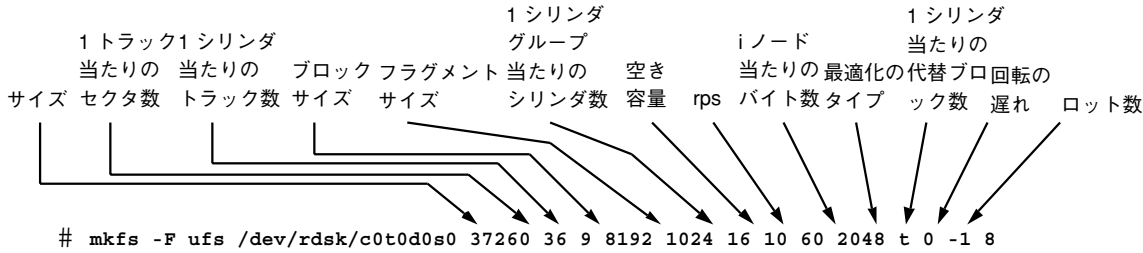
次の *newfs* の例では、-N オプションを使用して、バックアップスーパーブロックなど、ファイルシステム情報を表示します。

```
# newfs -N /dev/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 を呼び出して、指定したディスクスライス上で指定したタイプのファイルシステムを作成させます。mkfs では各種のファイルシステムがサポートされますが、実際には UFS ファイルシステムの作成に使用します。他のタイプのファイルシステムを作成するには、ファイルシステム専用の mkfs コマンドを使用できるようにソフトウェアを作成する必要があります。通常は、mkfs を直接実行せずに、newfs コマンドで呼び出します。

次の例は、mkfs コマンドのすべての引数を示しています。



汎用 mkfs コマンドは、/usr/sbin にあります。引数とオプションについては、mkfs(1M) のマニュアルページを参照してください。

パート IX データのバックアップと復元

このパートでは、Solaris 環境でデータのバックアップを取る手順とデータを復元する手順について説明します。次の章で構成されています。

| | |
|--------|--|
| 第 33 章 | ufsdump コマンドと ufsrestore コマンドを使用してデータのバックアップを取り、復元する際のガイドラインと計画について説明します。 |
| 第 34 章 | 個々のファイルとファイルシステム全体をローカルデバイスまたはリモートデバイスからバックアップする手順を説明します。 |
| 第 35 章 | 個々のファイルとファイルシステム全体を復元する詳しい手順を説明します。 |
| 第 36 章 | ufsdump の機能、ufsdump コマンドと ufsrestore コマンドの構文とオプションについて説明します。 |
| 第 37 章 | ファイルシステムをディスクにコピーする手順、dd、cpio、tar コマンドをさまざまなバックアップ媒体で使用する手順、ファイルを異なるヘッダー形式でコピーする手順について説明します。 |

第 38 章

テープドライブを追加する方法、テープドライブ、バックアップデバイス名を判断し、テープドライブと磁気テープカートリッジで作業する手順を説明します。

ファイルシステムのバックアップと復元の概要

この章では、`ufsdump` コマンドと `ufsrestore` コマンドを使用し、ファイルシステム全体のバックアップをとって復元する際のガイドラインと計画の作成について説明します。

この章の内容は以下のとおりです。

- 579ページの「バックアップと復元についての参照先」
- 580ページの「ファイルシステムのバックアップと復元とは」
- 581ページの「ファイルシステムをバックアップしなければならない理由」
- 581ページの「テープデバイスの選択」
- 582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」
- 585ページの「バックアップコマンドと復元コマンドの概要」
- 586ページの「バックアップタイプの選択」
- 587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」
- 590ページの「バックアップスケジュールの例」

バックアップと復元についての参照先

`ufsdump` コマンドと `ufsrestore` コマンドを使用してファイルシステムをバックアップおよび復元する手順については、次を参照してください。

- 第 34 章
- 第 35 章

ファイルシステムのバックアップと復元とは

ファイルシステムのバックアップとは、消失、損傷、または破損に備えて、ファイルシステムを取り外し可能な媒体 (テープなど) にコピーすることを意味します。ファイルシステムの復元とは、最新のバックアップファイルを取り外し可能な媒体から作業ディレクトリにコピーすることを意味します。

この章では、「スケジュールされた」バックアップ処理と復元処理に使用するコマンド (ufsdump と ufsrestore) について説明しますが、他にもファイルを共有または転送するためにファイルとファイルシステムのコピーに使用できるコマンドがあります。表 33-1 は、個々のファイルやファイルシステムを媒体にコピーする各コマンドの説明箇所を示しています。

表 33-1 ファイルとファイルシステムをコピーするコマンド

| 操作 | 使用するコマンド | 参照先 |
|--|---|------------------------------|
| ファイルシステム全体または個々のファイルシステムをローカルまたはリモートのテープデバイスにコピーする | ufsdump(1M) コマンド | 第 34 章 または 第 36 章 |
| ネットワーク上のすべてのシステムのファイルシステム全体をサーバーからバックアップする | Solstice Backup™ ソフトウェア | 『Solstice Backup 5.1 管理者ガイド』 |
| NIS+ マスターサーバーをバックアップ、復元する | nisbackup(1M) コマンドと nisrestore(1M) コマンド | 『Solaris ネーミングの管理』 |
| ファイルをテープ上でコピー、表示、検索する | tar(1)、cpio(1)、または pax(1) コマンド | 第 37 章 |
| ファイルをフロッピーディスク上でコピー、表示、検索する | tar(1) コマンド | |

表 33-1 ファイルとファイルシステムをコピーするコマンド 続く

| 操作 | 使用するコマンド | 参照先 |
|--|---------------------|--------|
| マスターディスクをクローンディスクにコピーする | dd(1M) コマンド | 第 37 章 |
| ファイルシステム全体または個々のファイルシステムを、取り外し可能な媒体から作業ディレクトリに復元する | ufsrestore(1M) コマンド | 第 35 章 |

ファイルシステムをバックアップしなければならない理由

ファイルのバックアップは、最も重要なシステム管理作業の 1 つです。次のような原因によるデータの消失に備えて、定期的にバックアップを実行する必要があります。

- システムのクラッシュ
- 不注意によるファイルの削除
- ハードウェア障害
- 天災 (火災、台風など)
- システムをインストールし直したりアップグレードするときの問題

テープデバイスの選択

表 33-2 に、バックアップ処理中にファイルシステムを格納するための典型的なテープデバイスを示します。テープデバイスについての詳細は、第 38 章を参照してください。

表 33-2 ファイルシステムのバックアップに使用する典型的な媒体

| 媒体 | 容量 ¹ |
|-----------------------------------|--------------------|
| 1/2 インチのリールテープ | 140M バイト (6250bpi) |
| 2.5G バイト、1/4 インチのカートリッジ (QIC) テープ | 2.5G バイト |
| DDS3 4-mm カートリッジテープ (DAT) | 12 ~ 24 G バイト |
| 14-Gbyte 8-mm カートリッジテープ | 14G バイト |
| DLT™ 7000 1/2 インチ カートリッジテープ | 35 ~ 70G バイト |

1. 容量は、ドライブのタイプとテープに書き込むデータによって異なります。

バックアップを作成するファイルシステムの計画

頻繁に更新されるファイルシステムなど、ユーザーにとって重要なファイルシステムはバックアップしておく必要があります。表 33-3 と表 33-4 に、スタンドアロンシステムとサーバー用にバックアップを作成するファイルシステムの一般的なガイドラインを示します。

表 33-3 ファイルシステムをスタンドアロン用にバックアップする

| バックアップするファイルシステム ¹ | 理由 | バックアップ間隔 |
|----------------------------------|--|----------|
| ルート (/) - パーティション 0 ² | ルート (/) には、メールやアカウントリングなど、頻繁に変更されるファイルが保管されるカーネルと /var ディレクトリが入っている。 | 定期 |
| /usr - パーティション 6 ² | 一般に、新しいソフトウェアをインストールして新しいコマンドを追加すると、/usr と /opt ファイルシステムが影響を受ける。/opt は、ルート (/) の一部であるか、独自のファイルシステムである。 | 随時 |

表 33-3 ファイルシステムをスタンドアロン用にバックアップする 続く

| バックアップするファイルシステム ¹ | 理由 | バックアップ間隔 |
|-------------------------------------|--|--|
| /export/home | /export/home には、スタンドアロンシステム上のユーザー全員のディレクトリとサブディレクトリが入っている。 | ルート (/) や /usr よりも頻繁に、サイトのニーズによっては毎日 1 度 |
| /export または /var を他の使用可能スライスにバックアップ | Solaris ソフトウェアのインストール中に、これらのファイルシステムを作成した可能性がある。 | サイトの必要に応じて |

1. df コマンドを使用するか、/etc/vfstab ファイルを調べて、どのスライスにファイルシステムが入っているかを調べる。
2. Solaris ソフトウェアのインストール時にデフォルトで作成されるファイルシステムを示す。

表 33-4 ファイルシステムをサーバー用にバックアップする

| バックアップするファイルシステム ¹ ... | 理由 | バックアップ間隔 |
|--|---|---|
| ルート (/) - パーティション 0 ² /export - パーティション 3 ² /usr - パーティション 6 ² , /opt | これらのファイルシステムには、カーネル、主要コマンド、実行可能プログラムが入っている。 | サイトのニーズに応じて毎日 1 度ないし月に 1 度 ルート (/) - ネットワーク上でクライアントとハードウェアを頻繁に追加したり削除したりする場合は、カーネル構成ファイルなどの重要なファイルをルート (/) 内で変更せざるを得ない。この場合は、週に 1 度から月に 1 度の間隔で、ルート (/) ファイルシステム上で完全バックアップを実行する必要がある。サイトでユーザーのメールをメールサーバー上の /var/mail ディレクトリに保管している場合は (その後クライアントシステムがマウント)、ルート (/) ディレクトリを毎日バックアップした方がよい。(あるいは、別のファイルシステムの場合は /var) /export - ディスクレスクライアントのルート (/) ディレクトリは、/export ファイルシステムに保管される。そこに入っている情報はサーバーのスライス 0 にあるルートディレクトリと同様なので、頻繁に変化することはない。随時バックアップを実行すればよい。ただし、サイトでメールをクライアントシステムに送信している場合は、/export のバックアップをさらに頻繁に実行する必要がある。 /usr と /opt - 内容に変更がなく、週に 1 度ないし月に 1 度バックアップするだけでかまわない。 |
| /export/home - パーティション 7 ² | /export/home には、システム上のユーザー全員のホームディレクトリとサブディレクトリが入っている。そのファイルは変更が多い。 | 毎日ないし毎週 |

- df コマンドを使用するか、/etc/vfstab ファイルを調べて、どのスライスにファイルシステムが入っているかを調べる。
- Solaris ソフトウェアのインストール時にデフォルトで作成されるファイルシステムを示す。

注 - サーバーの /export/swap のバックアップをとる必要はありません。

バックアップコマンドと復元コマンドの概要

`ufsdump` コマンドと `ufsrestore` コマンドは、ファイルシステム全体のバックアップを定期的に行う場合の推奨コマンドです。表 33-5 に、この 2 つのコマンドで実行できる作業を示します。各コマンドの機能と構文については、第 36 章を参照してください。

表 33-5 `ufsdump` と `ufsrestore` で実行できる作業

| コマンド | 実行できる作業 | 備考 |
|----------------------|--|--|
| <code>ufsdump</code> | ファイルシステム全体または一部のファイルシステムのバックアップを、ローカルまたはリモートのテープドライブに作成する。 | テープデバイスは、ネットワーク内でユーザーがアクセスするどのシステム上にあってもかまわない。このコマンドは UFS ファイルシステムタイプの構造を認識し、 <code>raw</code> デバイスインターフェース経由で直接処理するので高速である。 |
| | 増分バックアップの実行 | これにより、前回のバックアップ以降に変更があったファイルのバックアップのみを作成できる。 |
| | 単一システムからネットワーク経由でシステムグループをバックアップする。 | あるシステムから、リモートシェルまたはリモートログインを通して、各リモートシステム上で <code>ufsdump</code> を実行し、ドライブがあるシステムに出力を転送できる。また、出力をパイプを通して <code>dd</code> コマンドまたはファイルに渡すことができる。 |
| | 自動バックアップ | <code>crontab</code> ユーティリティを使用して、 <code>ufsdump</code> コマンドを起動するスクリプトを実行する。 |
| | バックアップテーブルへのユーザーアクセスを制限する。 | <code>-a</code> オプションを使用する。 |
| | 実際にバックアップを実行せずに、バックアップのサイズを決定する。 | <code>-s</code> オプションを使用する。 |
| | ファイルシステムがバックアップされたときのログを保管する。 | <code>-u</code> オプションを使用する。 |

表 33-5 ufsdump と ufsrestore で実行できる作業 続く

| コマンド | 実行できる作業 | 備考 |
|------------|---|----------------|
| | テープの内容をソースファイルシステムと対照して検査する。 | -v オプションを使用する。 |
| ufsrestore | 個々のファイルシステムまたはファイルシステム全体を、ローカルまたはリモートのテープドライブから復元する | |

バックアップタイプの選択

ufsdump コマンドを使用すると、完全バックアップまたは増分バックアップを実行できます。表 33-6 に、この 2 つのタイプのバックアップ手順の違いを示します。

表 33-6 完全バックアップと増分バックアップの違い

| バックアップ のタイプ | コピー内容 | 長所 | 短所 |
|----------------|--|---------------------------|---|
| 完全 | ファイルシステムまたはディレクトリ全体 | すべてを 1 箇所にとめることができる。 | 大量のバックアップテープが必要であり、書き込みに時間がかかる。ドライブはテープ上でファイルが入っている位置に順番に移動しなければならないので、個々のファイルの検索時間が長くなる。複数のテープを検索しなければならない場合もある。 |
| 増分 | 前回のバックアップ以降に変更があり、ファイルシステム上で指定したファイルのみ | ファイルシステム内の小さな変化を簡単に検索できる。 | どの増分テープにファイルが入っているかを探するのに時間がかかることがある。最後の完全ダンプに戻らなければならない場合もある。 |

バックアップスケジュールを設定する際のガイドライン

「バックアップスケジュール」とは、`ufsdump` コマンドを実行するように設定するスケジュールです。この節では、バックアップスケジュールを作成する際に検討すべき要素、ファイルシステムのバックアップ頻度に関するガイドライン、バックアップスケジュールの例について説明します。

バックアップスケジュールに影響する要素

どのようなスケジュールを設定するかは、次の要素によって左右されます。

- テープの本数を最小限度に抑える必要があるかどうか
- バックアップの実行に使用できる時間

- 損傷したファイルシステムの完全復元に使用できる時間
- 不注意に削除した個々のファイルの検索に使用できる時間

バックアップ頻度

バックアップに費やす時間と媒体を最小限度に抑える必要がない場合は、完全バックアップを毎日実行してもかまいません。しかし、多くのサイトの場合、これは現実的ではないので、ほとんどの場合は増分バックアップが使用されます。その場合は、サイトが過去4週間分のバックアップからファイルを十分復元できるようにしてください。そのためには、少なくとも1週分ごとに1組ずつ、合計4組のテープが必要で、各組を毎月使い廻ることになります。また、少なくとも一年分の月別のバックアップを保存し、数年分の年度別バックアップを保管しておく必要があります。

ダンプレベルを使用して増分バックアップを作成する

`ufsdump` コマンドで指定するダンプレベル (0-9) によって、どのファイルのバックアップが作成されるかが決まります。ダンプレベル 0 を指定すると、完全バックアップが作成されます。1 から 9 までの番号は、増分バックアップのスケジュール設定に使用されますが、特に意味が定義されているわけではありません。これらの番号は、累積バックアップまたは個別バックアップのスケジュール設定に使用する番号の範囲にすぎません。レベル 1 から 9 までが意味するのは、大小による相互関係だけです。

次の例は、1 から 9 までのレベルを使用する増分ダンプ手順を示しています。

日単位累積バックアップのダンプレベル

累積増分バックアップを毎日実行するのが、最も一般に使用される方法で、ほとんどの場合に推奨できます。次の例は、毎日レベル 9 のダンプを使用し、金曜日にはレベル 5 のダンプを使用してプロセスを再開するスケジュールを示しています。

注 - 次の例で、1 から 9 までの範囲内で他の番号を使用しても同じ結果になります。ポイントは、毎日同じ番号を使用し、金曜日にはそれより「小さい」番号を使用することです。たとえば、レベル 4、4、4、4、2 や 7、7、7、7、5 を指定してもかまいません。

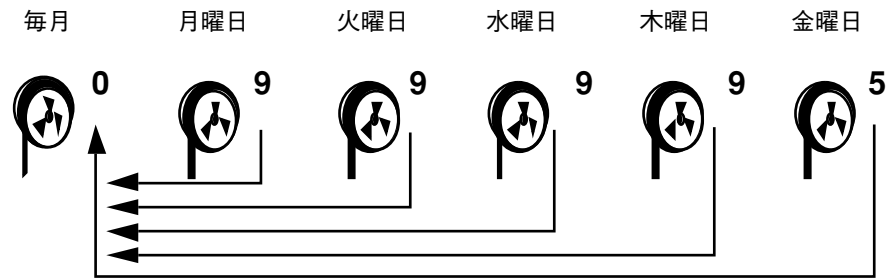


図 33-1 増分バックアップ: 日単位累積

日単位個別バックアップのダンプレベル

次の例は、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ています。この場合、月曜日から木曜日までは連続するダンプレベル番号 (3、4、5、6) を使用し、金曜日にはそれより小さい番号 (2) を使用します。

注 - 次の例では、連番 6、7、8、9 の次に 2 を使用しても、5、6、7、8 の次に 3 を使用してもかまいません。番号自体の意味は定義されておらず、その大小に意味があります。

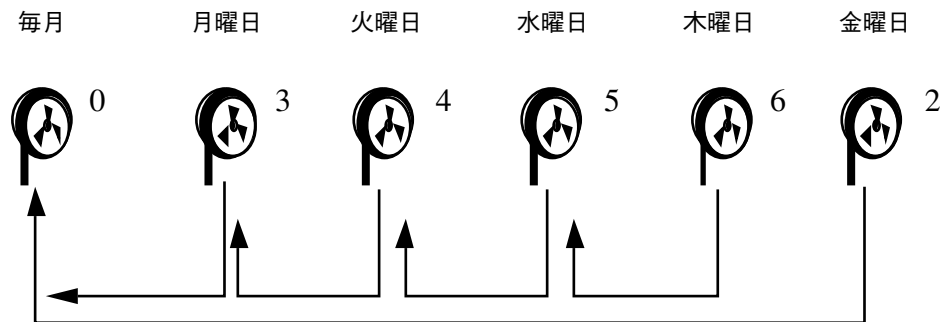


図 33-2 増分バックアップ: 日単位個別

バックアップスケジュールの例

この節では、バックアップスケジュールのサンプルを示します。どのスケジュールも、完全バックアップ(レベル 0)から始めることと、-u オプションを使用して各バックアップを記録することを前提としています。

例 — 日単位累積、週単位累積バックアップ

表 33-7 に、最も一般的に使用される増分バックアップをスケジュールを示します。これは、ほとんどの場合に推奨できるスケジュールです。このスケジュールで実行される処理は次のとおりです。

- 前週の終わりの下位レベルのバックアップ以降に変更があったすべてのファイルが毎日保存される。
- 月～金のレベル 9 のバックアップの場合は、直前のレベル 0 またはレベル 5 が最も近い下位バックアップレベルになる。したがって各週のテープには、前週の終わり(第 1 週の場合は初期レベル 0)以降に変更があったすべてのファイルが累積される。
- 毎週金曜日のレベル 5 の場合、レベルのうち一番近いレベルのバックアップは、月初めに実行されたレベル 0 である。したがって、毎週金曜日のテープには、月初めからその時点までに変更があったすべてのファイルが入っている。

表 33-7 日単位累積/週単位累積バックアップスケジュール

| | 未固定 | 月 | 火 | 水 | 木 | 金 |
|---------|-----|---|---|---|---|---|
| 月の 1 日目 | 0 | | | | | |
| 第 1 週 | | 9 | 9 | 9 | 9 | 5 |
| 第 2 週 | | 9 | 9 | 9 | 9 | 5 |
| 第 3 週 | | 9 | 9 | 9 | 9 | 5 |
| 第 4 週 | | 9 | 9 | 9 | 9 | 5 |

表 33-8 は、前のスケジュールを使用して、テープの内容が 2 週間でどのように変化するかを示します。各文字は別のファイルを表します。

表 33-8 日単位/週単位累積スケジュールのテープの内容

| | 月曜日 | 火曜日 | 水曜日 | 木曜日 | 金曜日 |
|-------|-----|-------|---------|-----------|--------------------------|
| 第 1 週 | a b | a b c | a b c d | a b c d e | a b c d e f |
| 第 2 週 | g | g h | g h i | g h i j | a b c d e f g h i j k |

必要なテープの本数

このスケジュールでは、6 本 (日単位テープを再利用したい場合) または 9 本 (曜日ごとに 4 本の日単位テープを別々に使用したい場合) のテープが必要になります。その内訳は、レベル 0 に 1 本、金曜日用に 4 本、日単位テープ用に 1 本、または 4 本です。

ファイルシステム全体を復元する必要がある場合、次のテープが必要になります。その内訳は、レベル 0 が 1 本、最後の金曜日のテープ 1 本、前週の金曜日以降の最新の日単位テープ 1 本です。

例 — 日単位累積、週単位増分バックアップ

表 33-9 は、各曜日のテープに月曜日 (第 1 週の場合は初期レベル 0) 以降に変更があったすべてのファイルが累積され、毎週金曜日のテープにはその週に変更があったすべてのファイルが入っているスケジュールを示しています。

表 33-9 日単位累積/週単位増分バックアップスケジュール

| | 未固定 | 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 |

表 33-9 日単位累積/週単位増分バックアップスケジュール 続く

| | 未固定 | 月 | 火 | 水 | 木 | 金 |
|-------|-----|---|---|---|---|---|
| 第 3 週 | | 9 | 9 | 9 | 9 | 5 |
| 第 4 週 | | 9 | 9 | 9 | 9 | 6 |

表 33-10 は、以前のスケジュールからテープの内容が2週間でのどのように変化するかを示しています。各文字はそれぞれ異なるファイルを表します。

表 33-10 日単位累積/週単位増分スケジュールのテープの内容

| | 月曜日 | 火曜日 | 水曜日 | 木曜日 | 金曜日 |
|-------|-----|-------|---------|-----------|-------------|
| 第 1 週 | a b | a b c | a b c d | a b c d e | a b c d e f |
| 第 2 週 | g | g h | g h i | g h i j | g h i j k |

必要なテープの本数

このスケジュールでは、6本(日単位テープを再利用したい場合)または9本(曜日ごとに4本の日単位テープを別々に使用したい場合)のテープが必要になります。その内訳は、レベル0に1本、金曜日用に4本、日単位テープ用に1本または4本です。

ファイルシステム全体を復元する必要がある場合は、次の本数のテープが必要になります。その内訳は、レベル0が1本、すべての金曜日のテープ、前週の金曜日以降の最新の日単位テープ1本です。

例 — 日単位増分、週単位累積バックアップ

表 33-11 は、各曜日のテープには前日以降に変更があったファイルのみが入っており、毎金曜日のテープには月初めの初期レベル0以降に変更があったすべてのファイルが入っているスケジュールを示しています。

表 33-11 日単位増分/週単位累積バックアップスケジュール

| | 未固定 | 月 | 火 | 水 | 木 | 金 |
|---------|-----|---|---|---|---|---|
| 月の 1 日目 | 0 | | | | | |
| 第 2 週 | | 3 | 4 | 5 | 6 | 2 |
| 第 3 週 | | 3 | 4 | 5 | 6 | 2 |
| 第 4 週 | | 3 | 4 | 5 | 6 | 2 |

表 33-12 は、テープの内容が 2 週間でどのように変化するかを示しています。各文字はそれぞれ異なるファイルを示します。

表 33-12 日単位累積/週単位累積バックアップスケジュールのテープの内容

| | 月曜日 | 火曜日 | 水曜日 | 木曜日 | 金曜日 |
|-------|-------|-----|-------|-----|--|
| 第 1 週 | a b | c d | e f g | h | a b c d e f g h i |
| 第 2 週 | j k l | m | n o | p q | a b c d e f g h i j k l m n o p q r s |

必要なテープの本数

このスケジュールでは、少なくとも 9 本のテープが必要になります。その内訳は、レベル 0 に 1 本、金曜日用に 4 本、日単位テープ 4 本ですが、これは日単位テープを毎週再利用することが前提となっており、お勧めできません。週単位テープを 1 カ月保存する場合は、21 本のテープが必要になります。

ファイルシステム全体を復元する必要がある場合は、次のテープが必要になります。その内訳は、レベル 0 に 1 本、前回の金曜日のテープ、最後の金曜日以降のその週のすべての日単位テープです。

例 — サーバーのバックアップスケジュール

表 33-13 は、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

プ方針の例を示しています。この例は、バックアップ期間が日曜日に始まり、1 週 7 日間で 4 週間行うものと想定しています。

表 33-13 サンプルサーバーのバックアップスケジュール

| ディレクトリ | 日付 | レベル | テープ名 |
|--------------|---------|-----|----------------|
| / | 第 1 日曜日 | 0 | <i>n</i> 個のテープ |
| /usr | 第 1 日曜日 | 0 | " |
| /export | 第 1 日曜日 | 0 | " |
| /export/home | 第 1 日曜日 | 0 | " |
| | 第 1 月曜日 | 9 | A |
| | 第 1 火曜日 | 9 | B |
| | 第 1 水曜日 | 5 | C |
| | 第 1 木曜日 | 9 | D |
| | 第 1 金曜日 | 9 | E |
| | 第 1 土曜日 | 5 | F |
| / | 第 2 日曜日 | 0 | <i>n</i> 個のテープ |
| /usr | 第 2 日曜日 | 0 | " |
| /export | 第 2 日曜日 | 0 | " |
| /export/home | 第 2 日曜日 | 0 | " |
| | 第 2 月曜日 | 9 | G |
| | 第 2 火曜日 | 9 | H |
| | 第 2 水曜日 | 5 | I |
| | 第 2 木曜日 | 9 | J |

表 33-13 サンプルサーバーのバックアップスケジュール 続く

| ディレクトリ | 日付 | レベル | テープ名 |
|--------------|---------|-----|---------|
| | 第 2 金曜日 | 9 | K |
| | 第 2 土曜日 | 5 | L |
| / | 第 3 日曜日 | 0 | n 個のテープ |
| /usr | 第 3 日曜日 | 0 | " |
| /export | 第 3 日曜日 | 0 | " |
| /export/home | 第 3 日曜日 | 0 | " |
| | 第 3 月曜日 | 9 | M |
| | 第 3 火曜日 | 9 | N |
| | 第 3 水曜日 | 5 | O |
| | 第 3 木曜日 | 9 | P |
| | 第 3 金曜日 | 9 | Q |
| | 第 3 土曜日 | 5 | R |
| / | 第 4 日曜日 | 0 | n 個のテープ |
| /usr | 第 4 日曜日 | 0 | " |
| /export | 第 4 日曜日 | 0 | " |
| /export/home | 第 4 日曜日 | 0 | " |
| | 第 4 月曜日 | 9 | S |
| | 第 4 火曜日 | 9 | T |
| | 第 4 水曜日 | 5 | U |
| | 第 4 木曜日 | 9 | V |

表 33-13 サンプルサーバーのバックアップスケジュール 続く

| ディレクトリ | 日付 | レベル | テープ名 |
|--------|---------|-----|------|
| | 第 4 金曜日 | 9 | W |
| | 第 4 土曜日 | 5 | X |

このスケジュールでは、4n本のテープ(ルート(/)、/usr、/export、/export/homeの4回の完全バックアップに必要な本数)に加えて、/export/homeの増分バックアップ用に24本のテープを使用します。このスケジュールは、増分バックアップごとに1本ずつテープを使用し、それを1カ月は保存することを前提としています。

このスケジュールの機能は次のとおりです。

1. 日曜日ごとに、ルート(/)、/usr、/export、/export/homeの完全バックアップ(レベル0)を実行します。レベル0のテープを少なくとも3カ月は保存します。
2. 月の第1月曜日に、テープAを使用して/export/homeのレベル9のバックアップを実行します。ufsdumpは下のレベルのバックアップ、この場合は日曜日に実行したレベル0のバックアップ以降に変更があったすべてのファイルをコピーします。
3. 月の第1火曜日に、テープBを使用して/export/homeのレベル9のバックアップを実行します。この場合も、ufsdumpは、下のレベル、つまり日曜日のレベル0のバックアップ以降に変更があったすべてのファイルをコピーします。
4. 第1水曜日に、テープCを使用してレベル5のバックアップを実行します。ufsdumpは日曜日以降に変更があったすべてのファイルをコピーします。
5. 木曜日と金曜日には、テープDとEを使用してレベル9のバックアップを実行します。ufsdumpは、下のレベルのバックアップ、つまり水曜日のレベル5のバックアップ以降に変更があったすべてのファイルをコピーします。
6. 月の第1土曜日に、/export/homeのレベル5のバックアップを実行します。このバックアップでは、下のレベルのバックアップ、この場合は日曜日に実行したレベル0のバックアップ以降に変更があったすべてのファイルがコピーされます。テープを再利用する場合は、テープAからFまでを次の4週間の第1月曜日までは保存しておきます。

7. 次の3週間は、テープGからLまでと、日曜日のレベル0用に4n本のテープを使用して、手順1から6までを繰り返します。
8. 4週ごとに、レベル0用に新しいテープ1組と、増分バックアップ用のテープAからXまでを再利用して、手順1から7までを繰り返します。レベル0のテープは、3カ月後に再利用できるようになります。

このスケジュールでは、各ファイルを1カ月間で段階別に保存できます。多数のテープが必要ですが、テープのライブラリを確実に用意できます。テープの本数を減らすには、テープAからFまでを毎週再利用します。

他のバックアップスケジュールに関する推奨事項

表 33-14 は、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ています。

表 33-14 システムのバックアップスケジュールに関する他の推奨事項

| 必要事項 | 推奨事項 | 備考 |
|--|---|---|
| 各ファイルの別バージョン(ワード処理に使用するファイルシステムなど)を復元する必要がある | <ul style="list-style-type: none"> ■ 作業日ごとに日単位増分バックアップを実行する ■ 日単位増分バックアップには同じテープを再利用しない | このスケジュールでは、その日に変更があったすべてのファイルが保存されるだけでなく、下のレベルの最後のバックアップ以降に変更があったファイルがディスク上に残る。ただし、このスケジュールの場合は、火曜日に変更があったファイルが木曜日にも変更されると、金曜日の下のレベルのバックアップでは、火曜日の夜ではなく木曜日の夜に変更されたように見えるので、毎日異なるテープを使用する必要がある。ユーザーが火曜日のバージョンを必要とする場合には、火曜日(または水曜日)のバックアップテープを保存しておかなければ、それを復元できない。また、火曜日と水曜日には存在したファイルが木曜日に削除されても、金曜日の下のレベルのバックアップには表示されない。 |
| ファイルシステム全体を短時間で復元する必要がある | 下位のレベルのバックアップを頻繁に実行する | — |
| 同じサーバー上で多数のファイルシステムのバックアップを作成している | ファイルシステムごとにスケジュールをずらすことを検討する | この方法では、すべてのレベル0のバックアップを同じ日に実行しないことになる。 |

表 33-14 システムのバックアップスケジュールに関する他の推奨事項 続く

| 必要事項 | 推奨事項 | 備考 |
|----------------------|----------------------------|--|
| テープの本数を最小限度に抑える必要がある | 1 週間に実行する増分バックアップのレベルを上げる | これは、毎日の変更のみが各日単位テープに保存されることを意味する。 |
| | 週末に実行するバックアップのレベルを上げる | これは、変更が (月単位ではなく) 週単位でしか週単位テープに保存されないことを意味する。 |
| | 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 | これは、 <code>ufsdump</code> コマンドの <code>rewind</code> オプションを使用して実行される。 |

ファイルとファイルシステムのバックアップの手順

この章では、`ufsdump` コマンドを使用してファイルシステムのバックアップを行う手順について説明します。

この章の内容は以下のとおりです。

- 600ページの「ファイルシステム名を検索する方法」
- 600ページの「完全バックアップをとるために必要なテープの本数を決定する方法」
- 602ページの「テープにバックアップをとる方法」

`ufsdump` コマンドの構文、オプション、引数についての詳細は、第36章を参照してください。

バックアップを実行する準備

ファイルシステムのバックアップを実行する準備は、計画の作成から始まります。詳細については、第33章を参照してください。この段階では、次の項目を決定します。

- テープドライブ
- バックアップを作成するファイルシステム
- バックアップのタイプ (完全または増分)
- バックアップのスケジュール

この節では、次のように、ファイルシステムのバックアップを作成する前に実行しておかなければならない作業について説明します。

- バックアップを作成するファイルシステムの名前を検索する。
- 完全バックアップをとるために必要なテープの本数を決定する。

▼ ファイルシステム名を検索する方法

1. /etc/vfstab ファイルの内容を表示します。

```
$ more /etc/vfstab
```

2. mount point カラムに表示されるファイルシステム名を調べます。
3. ファイルシステムをバックアップするときには、mount point カラムのマウントポイントを使用することになります。

例 — ファイルシステム名を検索する

```
$ more /etc/vfstab
#device          device          mount   FS   fsck   mount   mount
#to mount        to fsck         point   type pass  at boot options
#
/proc            -               /proc   proc -     no    -
swap            -               /tmp    tmpfs -     yes   -
/dev/dsk/c0t3d0s0 /dev/rdisk/c0t3d0s0 /        ufs   1     no    -
/dev/dsk/c0t3d0s1 -                -        swap -     no    -
/dev/dsk/c0t1d0s6 /dev/rdisk/c0t1d0s6 /usr     ufs   2     no    -
mars:/share/kit -                /kit    nfs   -     yes   -
mars:/db/doc    -                /db/doc nfs   -     yes   -
```

▼ 完全バックアップをとるために必要なテープの本数を決定する方法

1. スーパーユーザーになります。
2. `usfdump S` コマンドを使用して、バックアップのサイズをバイト単位で見積もります。

```
# usfdump S filesystem
```

S バックアップの実行に必要な予想バイト数を表示する。

3. 予想バイト数をテープの容量で除算して、必要な本数を調べます。
テープの容量リストについては、表 33-2 を参照してください。

例 — テープの本数を決定する

次の例では、150M バイトのテープに 489472 バイトのファイルシステムが入ります。

```
# ufsdump S /export/home  
489472
```

バックアップの実行

バックアップを実行する際の一般的なガイドラインは次のとおりです。

- シングルユーザーモードを使用するか、ファイルシステムをマウント解除する。
- ディレクトリレベルの処理 (ファイルの作成、削除、名前変更など) とファイルレベルの処理が行われているときにファイルシステムのバックアップを実行すると、バックアップに組み込まれないデータがあるので注意する。
- 1 台のシステムから `ufsdump` コマンドを実行し、リモートシェルまたはリモートログインを通じてネットワーク上のシステムグループのバックアップを実行し、その出力をテープドライブがあるシステムに転送できる (通常、テープドライブは `ufsdump` コマンドを実行するシステム上にあるが、他の場所にあってもかまわない)。

ファイルのバックアップをリモートドライブに作成するには、`ufsdump` コマンドの出力を `dd` コマンドにパイプする方法もある。`dd` コマンドの使用方法については、第 37 章を参照。

- ネットワーク上でリモートバックアップを実行する場合、テープドライブを持つシステムの `.rhosts` ファイル中には、ドライブを使用する各クライアントのエントリが入っていないなければならない。また、バックアップを実行する各システム

の `./rhosts` ファイルには、バックアップを開始するシステムのエンタリが入っていないなければならない。

- システム上でリモートドライブを指定するには、リモートテープドライブを持つシステムの OS バージョンと一致する命名規則を使用する。たとえば、SunOS 4.1.1 またはその互換バージョンを実行しているシステム上のリモートドライブには `/dev/rst0` を使用し、Solaris 7 またはその互換バージョンを実行しているシステムには `/dev/rmt/0` を使用する。

注 - Solaris 2.5 およびその互換バージョンが動作している NIS+ マスターサーバーをバックアップするためには `nisbackup` コマンドを使用して下さい。このコマンドを使用する方法については、『Solaris ネーミングの管理』を参照してください。

▼ テープにバックアップをとる方法

`ufsdump` コマンドを使用してファイルシステムのバックアップを作成する一般的な手順を次に示します。この例では、オプションと引数の使用方法を示しています。

1. スーパーユーザーになります。
2. システムをレベル **S** (シングルユーザーモード) に移行します。

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

3. [省略可能] `fsck` コマンドを指定してファイルシステムの整合性をチェックします。

`-m` オプションを使用して `fsck` コマンドを実行すると、ファイルシステムの整合性がチェックされます。たとえば、電源障害が発生すると、ファイルが非整合状態になることがあります。`fsck` コマンドについての詳細は、第 31 章を参照してください。

```
# fsck -m /dev/rdsk/ device-name
```

4. ファイルシステムをリモートテープドライブにバックアップする場合
 - a. テープドライブが接続されているシステム (テープサーバー) の `./rhosts` ファイルに、次のエンタリを追加します。

```
host root
```

host `ufsdump` を使用してバックアップを実行するシステムの名前

b. テープサーバー上で、上記の `.rhosts` ファイルに追加したホストに、ネットワークサービス経由でアクセスできることを確認します。

5. テープドライブのデバイス名を確認します。

デフォルトのテープドライブは、`/dev/rmt/0` です。

6. 書き込み保護されていないテープをテープドライブに挿入します。

7. `ufsdump` コマンドを使用してファイルシステムのバックアップを作成します。

次の表を参照して、`ufsdump` コマンドの最も一般的なオプションと引数を選択します。その他のオプションと引数については、第 36 章を参照してください。

| 操作 | 使用するオプションまたは引数 | 例 | 参照先 |
|--|-----------------------|---|-------------------------------|
| 完全バックアップを実行する | 0 オプション | <code>ufsdump 0ucf /dev/rmt/0 /</code> | 604ページの「例 — 完全バックアップ、ルート (/)」 |
| 増分バックアップを実行する | 1 から 9 までのオプション | <code>ufsdump 9ucf /dev/rmt/0/</code> | 605ページの「例 — 増分バックアップ、ルート (/)」 |
| 個々のファイルのバックアップを実行する | ファイルまたはディレクトリを指定する | <code>ufsdump ucf /dev/rmt/0 /export/home/kryten</code> | |
| ダンプを <code>/etc/dumpdates</code> ファイルに記録する | <code>-u</code> オプション | <code>ufsdump 9ucf /dev/rmt/ 0 /export/home</code> | 605ページの「例 — 増分バックアップ、ルート (/)」 |

| 操作 | 使用するオプションまたは引数 | 例 | 参照先 |
|---------------------------------------|--------------------------------|--|---|
| カートリッジテープを指定する | -c オプション | ufsdump 9ucf /dev/rmt/0 /export/home | 605ページの「例 — 増分バックアップ、ルート (/)」 |
| テープドライブを指定する | -f <i>dump-file</i> | ufsdump 9ucf /dev/rmt/0 /export/home | 605ページの「例 — 増分バックアップ、ルート (/)」 |
| ローカルファイルシステムをリモートホストのテープデバイスにバックアップする | <i>remote-system:dump-file</i> | ufsdump 0ucf pluto:/dev/rmt/0 /export/home | 607ページの「例 — リモートシステムへの完全バックアップ (Solaris 2.6 のデータを Solaris 7 システムへ)」 |

8. プロンプトが表示されたら、テープを取り出して次のテープと交換します。
9. 各テープにボリューム番号、レベル、日付、システム名、ディスクスライス、ファイルシステム名を記入したラベルを貼ります。
10. **Control-d** キーを押してシステムをレベル **3** の動作に戻します。
11. `ufsrestore` コマンドでテープの内容を表示して、バックアップが正常に実行されたことを確認します。
このコマンドについては、第 35 章で説明します。

例 — 完全バックアップ、ルート (/)

次の例では、ルート (/) ファイルシステムの完全バックアップを QIC-150 テープ (/dev/rmt/0) 上に作成します。

```
# shutdown -g30 -y
# ufsdump 0ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Wed Nov 06 15:32:21 1996
```

```

DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t3d0s0 (mars:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 41236 blocks (20.13MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 41200 blocks (20.12MB) on 1 volume at 95 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Wed Nov 06 15:32:21 1996
# ufsrestore tf /dev/rmt/0
  2 .
  3 ./lost+found
 5696 ./usr
11392 ./export
17088 ./opt
22784 ./var
28480 ./var/sadm
34176 ./var/sadm/install
39872 ./var/sadm/install/admin
39879 ./var/sadm/install/admin/default
      .
      .
      .
#

```

例 — 増分バックアップ、ルート (/)

この例では、ルート (/) ファイルシステムの増分バックアップを 4 mm の DAT テープデバイスに作成します。

```

# ufsdump 9ucf /dev/rmt/0 /
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 9 dump: Tue Jun 09 11:04:41 1998
DUMP: Date of last level 0 dump: Tue Jun 09 10:14:54 1998
DUMP: Dumping /dev/rdsk/c0t3d0s0 (pluto:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 1502 blocks (751KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 1384 blocks (692KB) on 1 volume at 51 KB/sec
DUMP: DUMP IS DONE

```

(続く)

続き

```
DUMP: Level 9 dump on Tue Jun 09 11:04:41 1998
# ufsrestore tf /dev/rmt/0
 39872 ./var
 62671 ./var/adm
 39915 ./var/adm/sa
 40018 ./var/adm/sa/sa09
 62810 ./var/adm/sulog
 62888 ./var/adm/pacct
 91154 ./var/cron
 91311 ./var/cron/log
  5716 ./var/mail
  5835 ./var/mail/adm
 45585 ./var/spool
 51388 ./var/spool/mqueue
 91155 ./var/tmp
.
.
.
```

例 — 完全バックアップ、個々のホームディレクトリ

この例では、/export/home/kryten ディレクトリの完全バックアップを 4 mm DAT テープに作成します。

```
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Jun 09 11:12:44 1998
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t3d0s7 (pluto:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 232 blocks (116KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 124 blocks (62KB) on 1 volume at 8 KB/sec
DUMP: DUMP IS DONE
# ufsrestore tf /dev/rmt/0
 2      .
 2688  ./kryten
 5409  ./kryten/letters
 5410  ./kryten/letters/letter1
 5411  ./kryten/letters/letter2
 5412  ./kryten/letters/letter3
 2689  ./kryten/.profile
 8096  ./kryten/memos
  30   ./kryten/reports
  31   ./kryten/reports/reportA
  32   ./kryten/reports/reportB
```

(続く)


```
# 33 ./kryten/reports/reportC
```

例 — リモートシステムへの完全バックアップ (Solaris 2.6 のデータを Solaris 7 システムへ)

次の例では、Solaris 2.6 システム上のローカルの /export/home ファイルシステムを、リモートの Solaris 7 システム pluto 上のテープデバイスに完全バックアップします。

```
# ufsdump 0ucf pluto:/dev/rmt/0 /export/home
# ufsdump 0ucf pluto:/dev/rmt/0 /export/home
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Jun 09 13:54:27 1998
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t3d0s7 (venus:/export/home) to pluto:/dev/rmt/
0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 38310 blocks (18.71MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 38302 blocks (18.70MB) on 1 volume at 98 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue Jun 09 13:54:27 1998
# # ufsrestore tf pluto:/dev/rmt/0
2 .
3 ./lost+found
34180 ./kryten
34182 ./kryten/.login
34183 ./kryten/.cshrc
51266 ./kryten/letters
51272 ./kryten/letters/letter1
51273 ./kryten/letters/letter2
51274 ./kryten/letters/letter3
57032 ./kryten/memos
74095 ./kryten/reports
74096 ./kryten/reports/reportA
74097 ./kryten/reports/reportB
74098 ./kryten/reports/reportC
.
.
.
#
```

例 — リモートシステムへの完全バックアップ (Solaris 7 のデータを Sun 4.1.3 システムへ)

次の例では、Solaris 7 システム上のローカルの /export/home ファイルシステムを、リモートの SunOS 4.1.3 システム mars 上のテープデバイスに完全バックアップします。

注 - ufsdump コマンドといっしょに SunOS 4.0/4.1 用のデバイス名 (/dev/rst0) が使用されているので注意してください。

```
# ufsdump 0ucf mars:/dev/rst0 /export/home
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Fri Oct 24 15:06:47 1997
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t3d0s7 (earth:/export/home) to (mars:/dev/
rst0).
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 19574 blocks (9.56MB)
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: level 0 dump on Fri Oct 24 15:06:47 1997
DUMP: Tape rewinding
DUMP: 19574 blocks (9.56MB) on 1 volume
DUMP: DUMP IS DONE
#

ufsrestore tf mars:/dev/rst0
 2      .
 3      ./lost+found
2688    ./kryten
5409    ./kryten/letters
5410    ./kryten/letters/letter1
5411    ./kryten/letters/letter2
5412    ./kryten/letters/letter3
2689    ./kryten/.profile
8096    ./kryten/memos
 30     ./kryten/reports
 31     ./kryten/reports/reportA
 32     ./kryten/reports/reportB
 33     ./kryten/reports/reportC
      .
      .
#
```

例 — リモートシステムへの完全バックアップ (SunOS 4.1.3 のデータを Solaris 7 システムへ)

次の例では、Sun 4.1.3 システム mars 上のローカルのルート (/) ファイルシステムを、Solaris 7 システム pluto 上のリモートテープデバイスに完全バックアップします。

注 - SunOS 4.1.3 およびその互換バージョンのシステム上のデータをバックアップするときには、`ufsdump` コマンドではなく `dump` コマンドを使用しなければならないことに注意してください。

```
# dump 0ucf pluto:/dev/rmt/0 /
DUMP: Date of this level 0 dump: Tue Oct 21 16:05:19 1997
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rsd2a (/) to /dev/rmt/0 on host pluto
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 8686 blocks (4.24MB) on 0.10 tape(s).
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: level 0 dump on Tue Oct 21 16:05:19 1997
DUMP: Tape rewinding
DUMP: 8690 blocks (4.24MB) on 1 volume
DUMP: DUMP IS DONE
# restore tf pluto:/dev/rmt/0
  2      .
  3      ./lost+found
 3776    ./export
  7552    ./home
11328    ./usr
15104    ./pcfs
  3777    ./tftpboot
  3778    ./tftpboot/tftpboot
  3794    ./tftpboot/boot.sun4c.sunos.4.1.3
  7553    ./etc
  7554    ./etc/sendmail.cf
  7555    ./etc/aliases
  7556    ./etc/aliases.dir
  7557    ./etc/aliases.pag
  7558    ./etc/holidays
  7559    ./etc/dumpdates
      .
      .
#
```


ファイルとファイルシステムの復元の手順

この章では、ファイルシステムを復元する手順について説明します。

この章で説明する手順は次のとおりです。

- 614ページの「使用するテープを決定する方法」
- 615ページの「対話式でファイルを復元する方法」
- 618ページの「対話式でない方法で特定のファイルを復元する方法」
- 620ページの「リモートドライブを使ってファイルを復元する方法」
- 621ページの「ファイルシステム全体を復元する方法」
- 624ページの「ルート (/) と /usr を復元する方法」

この章では、`ufsdump` コマンドによりバックアップしたファイルやファイルシステムを復元する場合の、`ufsrestore(1M)` コマンドの使用方法を説明します。ファイルやファイルシステムをアーカイブ、復元、コピー、または移動できる他のコマンドについては、第 37 章を参照してください。

ファイルとファイルシステムを復元する準備

`ufsrestore` コマンドを使うと、`ufsdump` コマンドで作成されたバックアップから現在の作業ディレクトリに対する相対パスで指定されるディスク上の位置にファイルがコピーされます。`ufsrestore` を使うと、ファイルシステムの階層構造全体をレベル 0 ダンプとそのあとに行われた増分ダンプから再ロードしたり、個々のファイルを任意のダンプテープから復元できます。`ufsrestore` がスーパーユー

ザーの権限で実行された場合には、ファイルの所有者、最新の変更時刻、モード (ファイルのアクセス権) は元のまま、ファイルが復元されます。

ファイルやファイルシステムの復元を始める前に、次のことがらをあらかじめ確認しておく必要があります。

- 必要なテープ (またはフロッピーディスク) はどれか
- ファイルシステム全体を復元したい raw デバイス名
- 使用するテープドライブのタイプ
- テープドライブのデバイス名 (ローカルまたはリモート)

ディスクデバイス名の決定

バックアップテープに適切な名前が付いている場合は、テープラベルに入っているディスクデバイス名 (`/dev/rdisk/devicename`) が使えるはずですが、詳細は、600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。

使用するテープドライブのタイプの決定

ファイルを復元するには、バックアップ媒体と互換性のあるテープドライブを使用しなければなりません。バックアップ媒体の形式によって、ファイルの復元にどんなドライブを使用しなければならないかが決まります。たとえば、使用するバックアップ媒体が 8mm テープの場合、ファイルの復元には 8mm テープドライブを使わなければなりません。

テープドライブ名の決定

テープデバイス名 (`/dev/rmt/n`) をバックアップテープラベル情報の一部として指定しているかも知れません。同じドライブを使ってバックアップテープを復元しようとする場合には、ラベル内にあるデバイス名を使うことができます。媒体のデバイスとデバイス名についての詳細は、第 38 章を参照してください。

ファイルシステム全体の復元

ファイルシステムは全面的に復元しなければならないほど破壊される場合があります。一般的な例として、ディスクヘッドが破損した場合は、ファイルシステムは全面的に復元しなければなりません。ただし、ソフトウェアを復元する前に、ハードウェアを取り替える必要があります。ディスクの取り替え方については、第 23 章または第 24 章を参照してください。/export/home などのファイルシステムを全面的に復元するには、長時間かかります。ファイルシステムを一貫性のある方法でバックアップしていれば、最後の増分バックアップ時の状態に復元することができます。

個別のファイルとディレクトリの復元

バックアップを実行すると、ファイルやディレクトリは、それらが含まれるファイルシステムからの相対的な位置に保存されます。ファイルとディレクトリを復元するときは、ufsrestore が現在の作業ディレクトリにファイル階層を作成し直します。たとえば、/export/doc/books ディレクトリ (/export はファイルシステムです) からバックアップされたファイルは、/export からの相対的な位置に保存されます。つまり、docs ディレクトリ内の book1 ファイルは、テープ上で ./doc/books/book1 として保存されます。後で、./doc/books/book1 ファイルを /var/tmp ディレクトリに復元する場合、そのファイルは /var/tmp/doc/books/book1 に復元されます。

個別のファイルやディレクトリを復元するときには、/var/tmp などの一時的な場所に復元するのが賢明です。ファイルやディレクトリを確認したら、それを適当な位置に移動させてもかまいません。個別のファイルやディレクトリはそれぞれ元の位置に復元できます。その場合には、新しいファイルをバックアップテープからの古いバージョンで上書きしないかどうか確かめてください。

注 - 一時的な場合でも、/tmp ディレクトリにファイルを復元してはなりません。一般的に /tmp ディレクトリは、TMPFS ファイルシステムとしてマウントされており、TMPFS は ACL などの UFS ファイルシステム属性をサポートしていません。

ファイルシステムの復元

次のことを知っておく必要があります。

- 復元したいファイルはどのテープに入っているか
- 復元したいファイルのパス名

▼ 使用するテープを決定する方法

1. 復元したいファイルが前回いつごろ変更されたかをユーザーにたずねます。
2. 自分が作成したバックアップ計画を参照して、そのファイルまたはファイルシステムが入っていた前回のバックアップの日付を調べます。
ファイルの最新バージョンを検索するには、増分バックアップファイルを最高レベル (0) から最低レベル (9) へ、最新バージョンから最古バージョンへ逆方向に調べます。
3. オンラインのアーカイブファイルがある場合には、`ufsrestore` コマンドを使って正しい媒体を指定します。

```
# ufsrestore ta archive-name ./path/filename ./path/filename
```

| | |
|------------------------------|---|
| <code>t</code> | テープ上に現れる各ファイルを表示する。 |
| <code>a</code> | テープではなくオンラインアーカイブファイルから内容一覧を読み取る。 |
| <code>archive-name</code> | オンラインアーカイブファイル名を指定する。 |
| <code>./path/filename</code> | オンラインアーカイブ上で検索するファイル名を指定する。コマンドが成功した場合は、 <code>ufsrestore</code> は 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。 |

4. バックアップが入っている媒体をドライブに挿入し、`ufsrestore` コマンドを使って正しい媒体であるかどうかを確認します。

```
# ufsrestore tf device-name ./path/filename ./path/filename
```


ファイル名には必ず完全パス名を使用してください。ファイルがバックアップに入っていれば、その名前とiノード番号が表示されます。入っていなければ、そのファイルはそのボリュームに入っていないというメッセージが表示されます。

5. 同じテープに複数のダンプファイルが入っている場合は、`s /dev/rmt/n` オプションを使って、自分が使用したいダンプがあるテープの位置に移動してください。

```
# ufsrestore tfs /dev/rmt/n tape_number
```

例 — 使用するテープを決定する

`ufsdump` を使って `/usr` スライスをダンプする場合、内容一覧には `/usr` の下にあるファイルとディレクトリが列挙されます。`/usr/bin/pwd` がオンラインアーカイブに入っているかどうかを調べるには、次のように入力します。

```
# ufsrestore ta archive-name ./bin/pwd
```

`/usr/bin/pwd` がバックアップテープに入っているかどうかを調べるには、次のように入力します。

```
# ufsrestore tf /dev/rmt/n ./bin/pwd
```

▼ 対話式でファイルを復元する方法

1. スーパーユーザーになります。
2. テープを書き込み保護の状態にします。
3. 最初のボリュームをテープドライブに挿入します。
4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

他のユーザーとの衝突を防ぐために、`/var/tmp/restore` などのサブディレクトリを作成し、そのディレクトリに移動してファイルを復元することをお勧めします。

階層を復元する場合は、ファイルを置くファイルシステム上の一時ディレクトリにファイルを復元します。復元後、mv コマンドで、階層全体を目的の場所に移動します。

5. ufsrestore コマンドを使って対話式で復元を開始します。
情報を伝えるためのメッセージと ufsrestore> プロンプトが表示されます。

```
# ufsrestore if /dev/rmt/n
```

6. 復元したいファイルのリストを作成します。
 - a. ディレクトリの内容を表示します。

```
ufsrestore> ls directory
```

- b. ディレクトリを変更します。

```
ufsrestore> cd directory-name
```

- c. 復元したいファイルとディレクトリのリストを作成します。

```
ufsrestore> add filename filename
```

- d. ディレクトリや、復元したいファイルのリストからファイル名を削除しなければならない場合は、delete コマンドを使います。

```
ufsrestore> delete filename
```

7. verbose モードをオンにして、復元するファイル名を表示します。

```
ufsrestore> verbose
```

8. リストが完了したら、extract コマンドを使います。

```
ufsrestore> extract
```

どのボリューム番号を使用するかを指定するプロンプトが表示されます。

9. ボリューム番号を入力して、**Return** キーを押します。ボリュームが **1** つしかない場合には、**1** を入力して **Return** キーを押します。

```
Specify next volume #: 1
```

リスト内のファイルとディレクトリが抽出され、現在の作業ディレクトリに復元されます。

10. 現在の作業ディレクトリのモードを変更したくない場合は、`set owner/mode` プロンプトが表示されたときに `n` を入力します。

```
set owner/mode for `.'? [yn] n
```

`ufsrestore` が最後の整理を行なっている間は、`ufsrestore` プロンプトが表示されるまでにはしばらく時間がかかります。

11. `ufsrestore` プログラムを終了します。

```
ufsrestore> quit
```

シェルプロンプトが表示されます。

12. ファイルが復元されたことを確認します。

- a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

- b. リストで指定したファイルとディレクトリがすべて復元されているかどうかを確認するため、リストを調べます。
- c. ファイルを正しいディレクトリに移動します。

例 — 対話方式でファイルを復元する

次の例では、ファイル `/etc/passwd` と `/etc/shadow` がバックアップテープから復元されています。

```
# cd /var/tmp
# ufsrestore if /dev/rmt/0
ufsrestore> ls
.:
.cpr_config  etc/          lost+found/  sbin/        usr/
TT_DB/       export/       mnt/         sccs/        var/
b/           home/         net/         share/       vol/
bin          kernel/       opt/         shared/      ws/
dev/         lib           platform/    src/         xfn/
devices/     license/      proc/        tmp/

ufsrestore> cd etc
ufsrestore> add passwd shadow
ufsrestore> verbose
verbose mode on
ufsrestore> extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./etc/shadow
extract file ./etc/passwd
Add links
Set directory mode, owner, and times.
set owner/mode for `.'? [yn] n
ufsrestore> quit
#
```

▼ 対話式でない方法で特定のファイルを復元する方法

1. スーパーユーザーになります。
2. 安全のためにテープを書き込み保護の状態にします。
3. ボリューム 1 のテープをテープドライブに挿入します。
4. ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

他のユーザーとの衝突を防ぐために、`/var/tmp` などのサブディレクトリを作成し、そのディレクトリに移動して、ファイルを復元することをお勧めします。階層を復元する場合は、ファイルを置くファイルシステム上の一時ディレクトリにファイルを復元します。復元後、`mv` コマンドで、階層全体を目的の場所に移動します。

5. `ufsrestore` コマンドを使ってファイルを復元します。

```
# ufsrestore xvf /dev/rmt/n filename ...
```

| | |
|---------------------------|--|
| <code>x</code> | 引数 <code>filename</code> 内に指定されたファイルまたはディレクトリをコピーするように <code>ufsrestore</code> コマンドに指定する。 |
| <code>v</code> | 復元するファイル名を表示する。 |
| <code>f /dev/rmt/n</code> | テープデバイス名を識別する。 |
| <code>filename ...</code> | 空白で区切られた 1 つ以上の個別のファイルまたはディレクトリの名前。例： <code>./export/home/user1/mail ./export/home/user2/mail</code> 。 |

6. ファイルが入っているボリューム番号を入力して、**Return** キーを押します。

```
Specify next volume #: 1
```

ファイルは現在の作業ディレクトリに復元されます。

7. 現在のディレクトリのモードを変更したくない場合は、`set owner/mode` プロンプトが表示されたときに `n` と入力して **Return** キーを押します。

```
set owner/mode for './?' [yn] n
```

8. 復元されたファイルを確認します。
 - a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

- b. リストをチェックして、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。
- c. ファイルを適切なディレクトリに移動します。

例 — 特定のファイルを復元する

この例では、passwd と shadow ファイルが /var/tmp ディレクトリに復元されます。

```
# cd /var/tmp
# ufsrestore xvf /dev/rmt/0 ./etc/passwd ./etc/shadow
Verify volume and initialize maps
Media block size is 126
Dump date: Tue Jun 09 14:30:16 1998
Dumped from: the epoch
Level 0 dump of / on pluto:/dev/dsk/c0t3d0s0
Label: none
Extract directories from tape
Initialize symbol table.
Warning: ./etc: File exists
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
Add links
Set directory mode, owner, and times.
set owner/mode for `.'? [yn] n
Directories already exist, set modes anyway? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l etc
```

▼ リモートドライブを使ってファイルを復元する方法

ufsrestore コマンドを使用するときにテープデバイス名の前に *remote-host:* を追加して、ファイルをリモートドライブから復元することができます。構文は次のようになります。

```
ufsrestore xf [user@]remote-host:/dev/rmt/n filename
```

例 — リモートドライブを使用してファイルを復元する

次の例では、システム `venus` 上にあるリモートテープドライブ `/dev/rmt/0` を使用してファイルを復元します。

```
# ufsrestore xf venus:/dev/rmt/0 filename
```

▼ ファイルシステム全体を復元する方法

注 - この手順は、ルート (`/`) または `/usr` の復元には使用できません。これらのファイルシステムを復元する方法については、624ページの「ルート (`/`) と `/usr` を復元する方法」を参照してください。

1. スーパーユーザーになります。
2. 必要があれば、ファイルシステムをマウント解除します。

```
# umount /dev/rdisk/device-name
```

3. `newfs` コマンドを使って新しいファイルシステムを作成します。

```
# newfs /dev/rdisk/device-name
```

`raw` デバイス上で新しいファイルシステムを構築したいかどうかをたずねるプロンプトが表示されます。デバイス名は正しく、削除しなければならないような間違ったファイルシステムがないことを確認します。

4. 新しいファイルシステムを作成しなければならないかどうかを確認します。

```
newfs: construct a new file system /dev/rdisk/cwtxdysz: (y/n)? y
```

新しいファイルシステムが作成されます。

5. 一時的なマウントポイントに新しいファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /mnt
```

6. ディレクトリを `/mnt` に変更します。

```
# cd mnt
```

マウントポイントディレクトリに変更しました。

7. テープを書き込み保護します。
8. レベル **0** テープの第 **1** ボリュームをテープドライブに挿入します。
9. `ufsrestore` コマンドを使ってテープ上のファイルを復元します。

```
# ufsrestore rvf /dev/rmt/n
```

レベル **0** ダンプが復元されます。ダンプの実行に複数のテープが必要な場合は、次のテープをロードするようにプロンプトが表示されます。

10. テープを取り出して、ドライブに次のレベルのテープをロードします。
テープの復元は必ず **0** から始め、一番高いレベルまで続けます。
11. 最低のレベルから最高のレベルまで、ダンプのレベルごとに、622ページの手順 **7** から **622**ページの手順 **10** までを繰り返します。
12. ファイルシステムが復元されていることを確認します。

```
# ls
```

13. `restoresymtable` ファイルを削除します。

```
# rm restoresymtable
```

`ufsrestore` によって作成された `restoresymtable` が削除されます。

14. 別のディレクトリに変更します。

```
# cd /
```

15. 新しく復元されたファイルシステムをマウント解除します。

```
# umount /mnt
```


16. 最後のテープを外し、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. `ufsdump` コマンドを使って、新しく復元されたファイルシステムのレベル **0** のバックアップをとります。

```
# ufsdump 0uf /dev/rmt/n /dev/rdisk/device-name
```

`ufsrestore` はファイルの位置を移動し、i ノード割り当てを変更するので、新しく作成したファイルシステムについては、ただちにバックアップをとるようにしてください。(復元されたファイルシステムは、以前のバックアップ以降に変更されているように見えます)。

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はマウントされて、使用できるようになります。

19. 復元されマウントされたファイルシステムが使用できることを確認します。

```
# ls /mount-point
```

例 — ファイルシステム全体を復元する

この例では、`/export/home` ファイルシステムが復元されます。

```
# umount /export/home
# newfs /dev/rdisk/c0t3d0s7
newfs: construct a new file system /dev/rdisk/c0t3d0s7: (y/n)? y
/dev/rdisk/c0t3d0s7:
410400 sectors in 270 cylinders of 19 tracks, 80 sectors
 200.4MB in 17 cyl groups (16 c/g, 11.88MB/g, 5696 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 24432, 48832, 73232, 97632, 122032, 146432, 170832, 195232, 219632,
 244032, 268432, 292832, 317232, 341632, 366032, 390432,
# mount /dev/dsk/c0t3d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump date: Tue Jun 09 15:01:03 1998
Dumped from: the epoch
```

(続く)

```
Level 0 dump of /export/home on pluto:/dev/dsk/c0t3d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./kryten
Make node ./kryten/letters
Make node ./kryten/memos
Make node ./kryten/reports
Make node ./rimmer
Make node ./rimmer/sc.directives
Make node ./rimmer/tests
Make node ./rimmer/answers
Extract new leaves.
Check pointing the restore
# ls
# rm restoresymtable
# cd /
# umount /mnt
# ufsdump 0ucf /dev/rmt/0 /export/home
.
.
.
# mount /dev/dsk/c0t3d0s7 /export/home
# ls /export/home
```

▼ ルート (/) と /usr を復元する方法

1. 新しいシステムディスクを、ルート (/) と /usr ファイルシステムが復元されるシステムに追加します。

システムディスクを追加する方法についての詳細は、第 23 章または第 24 章を参照してください。

2. 新しいファイルシステムを一時的なマウントポイントにマウントします。

```
# mount /dev/dsk/device-name /mnt
```

3. /mnt ディレクトリに変更します。

```
# cd /mnt
```

4. テープを書き込み保護します。
5. ufsrestore コマンドを使ってルートファイルシステムを復元します。

```
# ufsrestore rvf /dev/rmt/n
```

レベル 0 のテープが復元されます。

6. テープを外し、次のレベルのテープをドライブにロードします。
テープの復元は必ず 0 から始め、最も低いレベルから最も高いレベルまで続けます。
7. ufsrestore コマンドは必要なだけ続けて使用します。

```
# ufsrestore rvf /dev/rmt/n
```

次のレベルのテープが復元されます。

8. テープを追加するたびに、手順 6 と手順 7 を繰り返します。
9. ファイルシステムが復元されていることを確認します。

```
# ls
```

10. restoresymtable ファイルを削除します。

```
# rm restoresymtable
```

ufsrestore によって作成され、復元のチェックポイントに使用された restoresymtable ファイルを削除します。

11. ルートディレクトリに変更します。

```
# cd /
```

12. 新しく作成されたファイルシステムをマウント解除します。

```
# umount /mnt
```

13. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdisk/device-name
```

復元されたファイルシステムが完全であるかどうかチェックされます。

14. `installboot` コマンドで、ルートパーティションにブートブロックを作成します。

```
# installboot /usr/platform/`uname-i`/lib/fs/ufs/bootblk /dev/rdisk/ device-name
```

SPARC システム上で `installboot` コマンドを使用する方法については 627 ページの「例 — SPARC システム上でルート (/) と /usr を復元する」を、x86 システム上で `installboot` コマンドを使用する方法については 627 ページの「例 — x86 システム上でルート (/) ファイルシステムを復元する」をそれぞれ参照してください。

15. 新しいテープをテープドライブに挿入します。

16. 新しいファイルシステムのバックアップをとります。

```
# ufsdump 0uf /dev/rmt/n /dev/rdisk/device-name
```

レベル 0 のバックアップが実行されます。 `ufsrestore` はファイルの位置を移動し、i ノード割り当てを変更するので、新しく作成されたファイルシステムのバックアップは、必ずすぐにとるようにしてください。

17. 必要があれば、/usr ファイルシステムについて 625 ページの手順 5 から 626 ページの手順 18 を実行します。

18. システムをリブートします。

```
# init 6
```

システムがリブートされます。

例 — SPARC システム上でルート (/) と /usr を復元する

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```

例 — x86 システム上でルート (/) ファイルシステムを復元する

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdisk/c0t3d0s0
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot
/usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0
# init 6
```


ufsdump コマンドと ufsrestore コマンドの参照情報

この章では、ufsdump コマンドと ufsrestore コマンドの参照情報を示します。

この章の内容は以下のとおりです。

- 629ページの「ufsdump の機能」
- 635ページの「ufsdump コマンドのオプションと引数」
- 638ページの「ufsdump とセキュリティに関する注意事項」
- 639ページの「ufsrestore コマンドのオプションと引数」

ufsdump の機能

ufsdump コマンドは、ファイルシステムのバックアップを作成するときに 2 つのパスを作成します。最初のパスでは、ufsdump は raw デバイスファイル内でファイルシステムを走査し、メモリー内にディレクトリとファイルのテーブルを作成します。次に、そのテーブルをバックアップ媒体に書き込みます。2 つ目のパスでは、ufsdump は i ノードに番号順にアクセスし、ファイルの内容を読み込んで媒体に書き込みます。

デバイス特性の判断

ufsdump コマンドに必要なことは、適切なブロックの大きさを認識することと、どのようにして媒体の終りを検出するかということです。

媒体の終りの検出

ufsdump は一連の固定長レコードを書き込みます。レコードの一部にしか書き込まれていないという通知を受け取ると、ufsdump は媒体の物理的な終りに達したものと想定します。この方法は、ほとんどのデバイスに有効です。部分的なレコードしか書き込まれなかったことをデバイスが ufsdump に通知できない場合は、ufsdump が書き込もうとすると媒体エラーが発生します。

注 - DAT デバイスと 8mm テープデバイスでは、媒体の終りが検出されます。カートリッジテープデバイスと 1/2 インチテープデバイスでは、媒体の終りは検出されません。

データのコピー

ufsdump コマンドは、raw ディスクスライスからデータのみをコピーします。ファイルシステムがまだ有効であれば、メモリーバッファ内のデータがコピーされていない可能性があります。ufsdump によるバックアップでは、空きブロックはコピーされず、ディスクスライスのイメージも作成されません。シンボリックリンクが他のスライス上のファイルを指す場合は、リンク自体がコピーされます。

/etc/dumpdates ファイルの役割

ufsdump コマンドを `-u` オプションと一っしょに使用すると、`/etc/dumpdates` というファイルを管理し、更新できます。`/etc/dumpdates` 内の各行は、バックアップが作成されたファイルシステム、前回のバックアップレベル、バックアップ日時と曜日を示します。次の例は、ファイルサーバーからの典型的な `/etc/dumpdates` ファイルを示しています。

```
/dev/rdsk/c0t3d0s0          0 Wed Nov  6 15:32:21 1998
/dev/rdsk/c0t3d0s0          9 Wed Nov  6 16:14:51 1998
/dev/rdsk/c0t3d0s7          0 Wed Nov  6 16:18:08 1998
```


増分バックアップを実行するときに、`ufsdump` コマンドは `/etc/dumpdates` を検索して、下のレベルの最後のバックアップ日付を調べます。次に、下のレベルのバックアップ以降に更新されたすべてのファイルを媒体にコピーします。バックアップが完了すると、完了したばかりのバックアップを記述する新しい情報行によって、そのレベルの最後のバックアップの情報行が置き換えられます。

`/etc/dumpdates` ファイルを使用して、バックアップが実行中であるかどうかを検査してください。機器に問題が発生している場合は、この検査が特に重要です。機器の障害が原因でバックアップを完了できないと、そのバックアップは `/etc/dumpdates` ファイルに記録されません。

ディスク全体を復元する必要がある場合は、`/etc/dumpdates` ファイル内で最後のバックアップの日付とレベルをチェックできるので、ファイルシステム全体を復元するために必要なファイルを判断できます。

注 - `/etc/dumpdates` ファイルは編集可能なテキストファイルですが、編集するかどうかはユーザーの判断によります。ファイルに変更を加えた結果、アーカイブテープと一致なくなると、必要なテープ (またはファイル) がどれか分からなくなることがあります。

バックアップデバイス (*dump-file*) 引数

dump-file 引数 (`-f` オプションで使用) では、バックアップ先を指定します。次のいずれかを指定できます。

- ローカルのテープドライブかフロッピーディスクドライブ
- リモートのテープドライブかフロッピーディスクドライブ
- 標準出力

この引数は、バックアップ先がデフォルトのローカルテープドライブ `/dev/rmt/0` でないときに使用します。`-f` オプションを使用する場合は、*dump-file* の値を指定しなければなりません。

注 - また、*dump-file* 引数では、ローカルディスクまたはリモートディスク上のファイルを指すこともできるので、誤用するとファイルシステムがいっぱいになる可能性があります。

ローカルのテープドライブまたはフロッピーディスクドライブ

通常、*dump-file* ではテープドライブかフロッピーディスクドライブの *raw* デバイスファイルを指定します。*ufsdump* が出力デバイスに書き込むときには、バックアップファイルを1つ作成しますが、このファイルは複数のテープやフロッピーディスクにまたがってもかまいません。

デバイスの省略形を使用して、システム上のテープデバイスかフロッピーディスクデバイスを指定します。第1のデバイスは常に0です。たとえば、SCSI テープコントローラが1つと、媒体密度のコマンドを使用する QIC-24 テープドライブが1つある場合は、次のデバイス名を使用します。

```
/dev/rmt/0m
```

テープデバイス名を指定するときは、名前の末尾に文字“-n”を付けて、バックアップの完了後にテープドライブを巻き戻さないように指定することもできます。たとえば、次のように入力します。

```
/dev/rmt/0mn
```

テープに複数のファイルを格納したい場合は、“no-rewind” オプションを使用します。バックアップ中に領域を使い果たすと、*ufsdump* から新しいテープの挿入を促すプロンプトが表示されるまで、テープは巻き戻されません。デバイスの命名規則についての詳細は、676ページの「バックアップデバイス名」を参照してください。

リモートのテープドライブまたはフロッピーディスクドライブ

host:device という形式で、リモートのテープドライブまたはフロッピーディスクドライブを指定します。ローカルシステム上の *root* がリモートシステムへのアクセス権を持っていると、*ufsdump* はリモートデバイスに書き込みます。通常 *root* として *ufsdump* を実行するのであれば、ルートシステム名がリモートシステムの *.rhosts* ファイルに記述されている必要があります。デバイスを *user@host:device* と指定した場合、*ufsdump* は指定されたユーザーとしてリモートシステム上でデバイスにアクセスしようとします。この場合、指定されたユーザーの名前が、リモートシステム上の *.rhosts* ファイル中に含まれている必要があります。

デバイスには、*ufsdump* コマンドを実行するシステムではなく、そのデバイスが存在するシステムのオペレーティングシステムに合った命名規則を使用してください。デバイスが SunOS の旧バージョン (4.1.1 など) を実行するシステム上にある場合は、SunOS 4.1 でのデバイス名 (*/dev/rst0* など) を使用します。システムが Solaris ソフトウェアを実行中の場合は、SunOS 5.7 でのデバイス名 (*/dev/rmt/0*) を使用します。

注 - *dump-file* 引数を使用して、リモートデバイスを明示的に指定しなければなりません。SunOS の旧バージョンでは、*rdump* コマンドは出力を *dumphost* 別名で定義されたリモートデバイスに送っていました。*ufsdump* にはそれに相当する *rufsdump* がありません。

標準出力

dump-file 引数としてダッシュ (-) を指定すると、*ufsdump* は標準出力に書き込みます。

注 - *dump-file* 引数として標準出力を指定すると、*-v* オプション (検査) は機能しません。

ufsdump コマンドを使用して標準出力に書き込み、*ufsrestore* コマンドを使用して標準入力から読み込むと、パイプライン内でファイルシステムをコピーできます。次のようになります。

```
# ufsdump 0f - /dev/rdisk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

バックアップを作成するファイルを指定する

コマンド行の最後の引数として、必ず *files-to-backup* を指定しなければなりません。この引数は、バックアップのコピー元または内容を指定します。通常はファイルシステムを識別しますが、個々のファイルやディレクトリを識別することもできます。

ファイルシステムの場合は、ディスクスライスの *raw* デバイスファイルを指定します。これには、ファイルコントローラの省略形 (c)、SCSI デバイス専用のターゲット番号 (t)、ディスク番号 (d) とスライス番号 (s) の識別番号が含まれます。たとえば、スタンドアロンシステム (またはサーバー) 上に SCSI ディスクコントローラがあり、スライス 6 に入っている */usr* のバックアップを作成したい場合は、デバイスを次のように指定します。

```
/dev/rdisk/c0t0d0s6
```

ファイルシステムは、そのエントリが */etc/vfstab* ファイルに入っていれば、マウントポイントディレクトリ (*/home* など) を使用して指定できます。

デバイスの命名規則についての詳細は、676ページの「バックアップデバイス名」を参照してください。

個々のファイルやディレクトリごとに、1つまたは複数の名前を空白で区切って入力します。

注 - `ufsdump` を使用して (ファイルシステム全体ではなく) 1つまたは複数のディレクトリまたはファイルのバックアップを作成するときには、レベル 0 のバックアップが実行されます。増分バックアップは適用されません。

媒体の終りの検出

`ufsdump` は、ほとんどのデバイスの媒体の終りを自動的に検出します。したがって、通常は `-c`、`-d`、`-s`、`-t` オプションを使用しなくても、複数のボリュームのバックアップを実行できます。

媒体の終りオプションを使用する必要があるのは、デバイスによる媒体の終りの検出方法が `ufsdump` で認識されないときや、`restore` コマンドの旧バージョンを使用してシステム上のファイルを復元しようとしているときです。`restore` コマンドの旧バージョンとの互換性を確保するために、`size` オプションを使用すると、従来どおり、現在のテープやフロッピーディスクの終りに達する前に、`ufsdump` を次のテープやフロッピーディスクに強制的に進ませることができます。

テープの性質を指定する

テープの性質を指定しなければ、`ufsdump` コマンドはデフォルト設定を使用します。テープカートリッジ (`-c`)、密度 (`-d`)、サイズ (`-s`)、トラック数 (`-t`) を指定できます。引数がオプションの順序と合っていれば、オプションはいくつでも指定できます。

`ufsdump` の制限

表 36-1 は、`ufsdump` コマンドでは実行できない作業を示しています。

表 36-1 ufsdump で実行できない作業

| 実行できない作業 | 備考 |
|--|--|
| ファイルシステムのバックアップに必要なテープやフロッピーディスクの数を自動的に計算する。 | 仮実行 (ドライラン) モード (S オプション) を使用すると、実際にファイルシステムをバックアップする前に必要な容量を判断できる。 |
| アクティブなファイルシステムをバックアップするときの問題を最小限度に抑えるために、組み込みエラーチェック機能を提供する。 | — |
| サーバーからリモートにマウントされたファイルをバックアップ可能にする。 | サーバー上のファイルのバックアップは、そのサーバー上で実行しなければならない。ユーザーがサーバー上で所有するファイル上で ufsdump を実行するアクセス権は拒否される。 |

ufsdump コマンドのオプションと引数

この節では、ufsdump コマンドのオプションと引数について詳しく説明します。ufsdump コマンドの構文は次のとおりです。

```
/usr/sbin/ufsdump [options] [arguments] files-to-back-up
```

| | |
|-------------------------|---|
| <i>options</i> | 1 文字のオプション名からなる 1 つの文字列 |
| <i>arguments</i> | オプションの引数を指定する。複数の文字列も可。オプション文字と引数とは同じ順序で並べなければならない。 |
| <i>files-to-back-up</i> | バックアップするファイルを指定する。これらの引数は常に最後に指定する。 |

デフォルトのコマンドオプション

オプションを指定せずに ufsdump コマンドを実行する場合は、次の構文を使用します。

```
# ufsdump files-to-back-up
```

ufsdump では、デフォルトで次のオプションが使用されます。

```
ufsdump 9uf /dev/rmt/0 files-to-back-up
```

これらのオプションでは、デフォルトのテープドライブ上にその推奨密度でレベル 9 の増分バックアップが作成されます。

ufsdump コマンドのオプション

表 36-2 は、ufsdump コマンドのオプションを示しています。

表 36-2 ufsdump コマンドのオプション

| オプション | 説明 |
|-----------------------|---|
| 0-9 | バックアップレベル。レベル 0 は <i>files-to-backup</i> で指定したファイルシステム全体の完全バックアップ用。レベル 1～9 は、最後の下位バックアップ以降に変更があったファイルの増分バックアップ用。 |
| a <i>archive-file</i> | アーカイブファイル。指定したディスク上のファイルにバックアップ用の内容一覧を格納 (アーカイブ) する。このファイルは、 <i>ufsrestore</i> でしか認識できず、復元すべきファイルがバックアップファイル内にあるかどうかと、もしあればどの媒体ボリュームに入っているかを判断するために使用される。 |
| b <i>factor</i> | ブロック係数。1 処理ごとにテープに書き込まれる 512 バイトのブロック数。 |
| c | カートリッジ。カートリッジテープにバックアップを作成する。媒体の終りの検出を適用するときは、このオプションでブロックサイズを 126 に設定する。 |
| d <i>bpi</i> | テープ密度。このオプションは、ufsdump で媒体の終りを検出できないときにのみ使用する必要がある。 |
| D | フロッピーディスク。フロッピーディスクにバックアップを作成する。 |

表 36-2 ufsdump コマンドのオプション 続く

| オプション | 説明 |
|--------------------------|--|
| <code>f dump-file</code> | 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 <code>user@system:device</code> として指定すると、 <code>ufsdump</code> は指定されたユーザーとしてリモートシステム上で実行しようとする。このコマンドをローカルシステム上で実行しリモートシステムにアクセスできるように、指定されたユーザーはリモートシステム上に <code>.rhosts</code> ファイルを持っていないなければならない。 |
| <code>l</code> | 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。テープの終りに達すると、このオプションはドライブをオフラインにして、テープドライブの準備ができるまで 2 分間待つ。2 分以内にドライブの準備ができると、続行する。2 分経過しても準備ができなければ、オペレータに別のテープをロードするように促すプロンプトを表示する。 |
| <code>n</code> | 通知。介入が必要になると、 <code>sys</code> グループのユーザー全員の端末にメッセージを送る。 |
| <code>o</code> | オフライン。テープやフロッピーディスクの処理が終わると、ドライブをオフラインにして巻き戻し (テープの場合)、可能であれば媒体をはずす (たとえば、フロッピーディスクを取り出したり、8mm の自動ロードテープをはずす)。 |
| <code>s size</code> | サイズ。テープの場合はフィート数、フロッピーディスクの場合は 1024 バイトのブロック数を指定する。このオプションは、 <code>ufsdump</code> で媒体の終りを検出できないときのみ使用する必要がある。 |
| <code>S</code> | バックアップの予想サイズ。バックアップを実際に行わずに必要な容量を判断し、バックアップの予想バイト数を示す数値を 1 つ出力する。 |
| <code>t tracks</code> | トラック数。1/4 インチカートリッジテープのトラック数を指定する。このオプションを使用する必要があるのは、 <code>ufsdump</code> で媒体の終りを検出できないときだけである。 |

表 36-2 ufsdump コマンドのオプション 続く

| オプション | 説明 |
|-------|---|
| u | ダンプレコードのアップデート。ファイルシステムの完全バックアップをとる場合には、ファイル <code>/etc/dumpdates</code> ファイルにエントリを追加する。このエントリは、ファイルシステムのディスクスライスのデバイス名、バックアップレベル (0 ~ 9)、日付を示す。u オプションを使用しないと、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が書き込まれた後に、ソースファイルシステムと対照して媒体の内容を検査する。矛盾が発生すると、オペレータに新しい媒体のマウントを促すプロンプトを表示してプロセスを繰り返す。ファイルシステム内で動作が発生すると矛盾が報告されるので、このオプションはマウント解除されたファイルシステム専用である。 |
| w | 警告。 <code>/etc/dumpdates</code> ファイルに表示されるファイルシステムのうち、特定の日にバックアップされていないファイルシステムを表示する。このオプションを使用すると、他のすべてのオプションは無視される。 |
| W | 強調表示付きの警告。 <code>/etc/dumpdates</code> 内のすべてのファイルシステムを表示し、特定の日にバックアップされていないファイルシステムを強調表示する。このオプションを使用すると、他のすべてのオプションは無視される。 |

注 - `/etc/vfstab` ファイルには、ファイルシステムのバックアップ頻度に関する情報は入っていません。

ufsdump とセキュリティに関する注意事項

セキュリティに関する注意事項は次のとおりです。

- ufsdump コマンドへの root アクセス権が必要である。
- 集中バックアップを実行する場合は、クライアント上とサーバー上の `.rhosts` ファイルから root アクセスエントリを削除しておかなければならない。

セキュリティについての詳細は、『Solaris のシステム管理 (第 2 巻)』の「システムセキュリティの管理の概要」を参照してください。

ufsrestore コマンドのオプションと引数

コマンド構文

ufsrestore の構文は次のとおりです。

```
ufsrestore [options] [arguments] [filename ...]
```

| | |
|------------------|---|
| <i>options</i> | 1 文字のオプション名からなる 1 つの文字列。i、r、R、t、x から 1 つだけ選択する。 |
| <i>arguments</i> | オプションに対応する引数。オプションと同じ順序で並んでいなければならない。 |
| <i>filename</i> | x または t オプションの引数として、復元するファイルを指定する。常に最後に指定する。 |

オプションと引数

ufsrestore コマンドには、表 36-3 に示すオプションのうちの 1 つを指定する必要があります。

表 36-3 ufsrestore コマンドに必要なオプション

| オプション | 説明 |
|-----------------|---|
| i | 対話式。ufsrestore を対話モードで実行する。このモードでは、限られたシェルコマンドセットを実行して媒体の内容を表示し、復元するファイルやディレクトリを個別に選択できる。利用できるコマンドのリストについては、642ページの「対話式の復元コマンド」を参照。 |
| r | 再帰。媒体の内容全体を現在の作業ディレクトリ (ファイルシステムの最上位レベル) に復元する。完全ダンプ (例、restoresymtable) の最上部に増分ダンプを復元するための情報も含まれる。ファイルシステムを完全に復元するには、このオプションを使用して完全 (レベル 0) ダンプを復元してから、各増分ダンプを復元する。これは新しいファイルシステム (newfs コマンドで作成したばかりのファイルシステム) 用だが、バックアップ媒体にないファイルが保存される。 |
| R | 復元の再開。復元を再開するボリュームをたずねるプロンプトを表示し、チェックポイントから再開する。完全復元 (r オプション) が中断された後は、このオプションを指定して ufsrestore コマンドを実行し直す。 |
| x [filename...] | 抽出。filename 引数で指定したファイルを選択的に復元する。filename では、ファイルとディレクトリのリストを指定できる。h オプションも指定しなければ、指定したディレクトリの下ファイルがすべて復元される。filename を省略するか、ルートディレクトリを表す「.」を入力すると、媒体のすべてのボリューム上 (または標準入力から) のすべてのファイルが復元される。既存のファイルは上書きされ、警告が表示される。 |
| t [filename...] | 内容一覧。filename 引数で指定したファイルが媒体と対照してチェックされる。ファイルごとに、完全ファイル名と i ノード番号 (ファイルが見つかった場合) が表示されるか、ファイルが「ボリューム」上にないことを示す (複数のボリュームダンプのボリュームを意味する)。filename 引数を入力しなければ、媒体のすべてのボリューム上のファイルが表示される (どのボリュームにファイルが入っているかは区別されない)。h オプションも指定すると、内容ではなく filename で指定したディレクトリファイルのみがチェックされ表示される。内容一覧は、媒体の最初のボリュームから読み込まれるか、a オプションを使用すると指定したアーカイブファイルから読み込まれる。このオプションは x、r オプションといっしょに使用できない。 |

表 36-3 からオプションを 1 つ選択する他に、表 36-4 からオプションを選択できます。

表 36-4 ufsrestore コマンドの追加オプション

| オプション | 説明 |
|--|---|
| a <i>archive-file</i> [<i>filename...</i>] | ダンプの内容一覧は、媒体 (最初のボリューム) ではなく、指定した <i>archive-file</i> からダンプの内容一覧が取り出される。このオプションを <i>t</i> 、 <i>i</i> 、または <i>x</i> オプションと組み合わせて使用すると、媒体をマウントしなくてもダンプ内のファイルをチェックできる。 <i>x</i> コマンドと対話型抽出オプションといっしょに使用すると、ファイルを抽出する前に適切なボリュームのマウントを促すプロンプトが表示される。 |
| b <i>factor</i> | ブロック係数。1 処理ごとにテープに書き込む 512 バイトのブロック数。デフォルトでは、 <i>ufsrestore</i> はテープの書き込みに使用したブロック数を表示しようとする。 |
| d | デバッグ。デバッグメッセージ機能をオンにする。 |
| f <i>backup-file</i> | バックアップファイル。ファイルは、デフォルトのデバイスファイル <i>/dev/rmt/0m</i> ではなく <i>backup-file</i> で指定したソースから読み込まれる。f オプションを使用する場合は、 <i>backup-file</i> の値を指定しなければならない。 <i>backup-file</i> が <i>system:device</i> 形式であれば、 <i>ufsrestore</i> はリモートデバイスから読み込む。 <i>backup-file</i> 引数を使用すると、ローカルディスクやリモートディスク上のファイルも指定できる。 <i>backup-file</i> が "." であれば、ファイルは標準入力から読み込まれる。 |
| h | ディレクトリの展開をオフにする。指定したディレクトリファイルのみが抽出または表示される。 |
| m | 指定した http://www.aaapc.co.jp/Hokkaido/area/hidaka/try/t-1c.htm ファイルが、バックアップ階層内の位置に関係なくディスク上の現在のディレクトリに復元され、 <i>i</i> ノード番号を使用して名前が変更される。たとえば、現在の作業ディレクトリが <i>/files</i> であれば、 <i>i</i> ノード番号が 42 のバックアップ <i>./dready/fcs/test</i> 内のファイルは、 <i>/files/42</i> として復元される。このオプションが便利なのは、少数のファイルを抽出するときだけである。 |
| s <i>n</i> | 最初のボリューム媒体上の <i>n</i> 番目のバックアップファイルまでスキップする。このオプションは、1 本のテープに複数のバックアップを入れるときに便利である。 |

表 36-4 ufsrestore コマンドの追加オプション 続く

| オプション | 説明 |
|-------|--|
| v | 詳細表示。各ファイルが復元されるたびに、その名前と i ノード番号が表示される。 |
| y | 媒体の読み込みエラーが発生しても処理を続行する。処理を停止して続行するかどうかを選択するプロンプトを表示せずに、不良ブロックをスキップしようとする。 |

対話式の復元コマンド

表 36-5 対話式の復元コマンド

| オプション | 説明 |
|------------------------------|---|
| ls [<i>directory-name</i>] | 現在のディレクトリまたは指定したディレクトリの内容が表示される。ディレクトリは接尾辞 / 付きで表示され、現在のリスト内で復元 (抽出) されるエントリは接頭辞 * 付きで表示される。詳細オプションを使用すると、i ノード番号が表示される。 |
| cd <i>directory-name</i> | バックアップ階層内の指定したディレクトリに切り替える。 |
| add [<i>filename</i>] | 現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストに追加する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のすべてのファイルがリストに追加される。ディレクトリに復元したいすべてのファイルが 1 つのバックアップテープやフロッピーディスクに入っていないことがあるので注意する必要がある。すべてのファイルの最新の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ことがある。 |
| delete [<i>filename</i>] | 現在のディレクトリまたは指定したファイルやディレクトリを、抽出 (復元) するファイルのリストから削除する。h オプションを使用しない場合は、指定したディレクトリとそのサブディレクトリ内のすべてのファイルが削除される。ファイルとディレクトリは、構築中の抽出リストからのみ削除されるので注意する必要がある。媒体またはファイルシステムからは削除されない。 |

表 36-5 対話式の復元コマンド 続く

| オプション | 説明 |
|----------|--|
| extract | リスト内のファイルを抽出し、ディスク上の現在の作業ディレクトリからの相対パスで指定される位置に復元する。1つのボリュームのバックアップについてボリューム番号をたずねるプロンプトが表示されたら 1 を指定する。複数テープや複数フロッピーディスクから少数のファイルを復元する場合は、最後のテープまたはフロッピーディスクから始める。 |
| help | 対話モードで使用できるコマンドのリストが表示される。 |
| pwd | バックアップ階層内の現在の作業ディレクトリのパス名が表示される。 |
| q | それ以上ファイルを復元しないで対話モードを終了する。 |
| setmodes | バックアップ元となったファイルシステムのルートディレクトリのモードに合わせて、復元するファイルのモードを設定できる。set owner/mode for '.' [yn]? というプロンプトが表示される。y (yes の意味) を入力すると、バックアップ元となったファイルシステムのルートディレクトリに合わせて、現在のディレクトリのモード (アクセス権、所有者、時刻) を設定できる。このモードは、ファイルシステム全体を復元するときに使用する。n (no の意味) を入力すると、現在のディレクトリのモードは変更されずにそのまま残る。このモードは、バックアップの一部をファイルのバックアップ元とは異なるディレクトリに復元するときに使用する。 |
| verbose | 詳細オプションのオンとオフを切り替える (対話型モードの外側では、コマンド行から v と入力することもできる)。詳細モードがオンになっていると、対話型の ls コマンドでは i ノード番号が表示され、ufsrestore コマンドでは各ファイルが抽出されるたびにファイル情報が表示される。 |
| what | テープやフロッピーディスク上のバックアップヘッダが表示される。 |

UFS ファイルとファイルシステムのコピー手順

この章では、各種のバックアップコマンドを使用して、UFS ファイルとファイルシステムをディスク、テープ、フロッピーディスクにコピーする方法について説明します。

この章で説明する手順は次のとおりです。

- 649ページの「ディスクをクローン化する方法 (dd)」
- 652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)」
- 656ページの「ファイルをテープにコピーする方法 (tar)」
- 657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)」
- 658ページの「テープからファイルを取り出す方法 (tar)」
- 661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)」
- 662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)」
- 663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)」
- 664ページの「指定したファイルをテープから取り出す方法 (cpio)」
- 665ページの「ファイルをリモートテープドライブにコピーする方法 (tar と dd)」
- 667ページの「ファイルをリモートテープドライブから取り出す方法」
- 669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)」

- 670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)」
- 671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)」
- 672ページの「ファイルを複数のフロッピーディスクにアーカイブする方法」
- 673ページの「SunOS の旧リリース用のアーカイブを作成する方法」
- 674ページの「bar ファイルをフロッピーディスクから取り出す方法」

ファイルシステムのコピーコマンド

ファイルシステム全体をバックアップして復元したいときは、第 36 章で説明した `ufsdump` コマンドと `ufsrestore` コマンドを使用します。個々のファイル、ファイルシステムの一部、またはファイルシステム全体をコピーまたは移動したいときは、`ufsdump` と `ufsrestore` の代わりに、この章で説明する手順を使用できます。

表 37-1 は、各種バックアップコマンドの用途を示しています。

表 37-1 バックアップコマンドの用途

| 目的 | 使用するコマンド | 参照ページ |
|-----------------------|--|----------------------------------|
| ファイルシステムをテープにバックアップする | <code>ufsdump(1M)</code> | 602ページの「テープにバックアップをとる方法」 |
| ファイルシステムをテープから復元する | <code>ufsrestore(1M)</code> | 621ページの「ファイルシステム全体を復元する方法」 |
| ファイルを他のシステムに転送する | <code>pax(1)</code> , <code>tar(1)</code> , または <code>cpio(1)</code> | 654ページの「ファイルとファイルシステムをテープにコピーする」 |

表 37-1 バックアップコマンドの用途 続く

| 目的 | 使用するコマンド | 参照ページ |
|----------------------------|----------|--|
| ファイルまたはファイルシステムをディスクにコピーする | dd (1M) | 649ページの「ディスクをクローン化する方法 (dd)」 |
| ファイルをフロッピーディスクにコピーする | tar (1) | 669ページの「ファイルを1枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)」 |

表 37-2 に、各種のバックアップコマンドと復元コマンドを示します。

表 37-2 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 | 物理コピーか論理コピーか |
|--------------------|---------------|--------------------|--------------|
| volcopy | する | する | 物理 |
| tar | しない | しない | 論理 |
| cpio | しない | する | 論理 |
| pax | する | する | 論理 |
| dd | する | しない | 物理 |
| ufsdump/ufsrestore | する | する | 論理 |

次の節では、各方法の長所と短所を説明し、コマンドの使用例を示します。

ファイルシステムをディスクにコピーする

ファイルシステムをディスクにコピーするには、次の2つのコマンドを使用します。

- volcopy
- dd

次の節では、dd コマンドを使用してファイルシステムをディスク間でコピーする方法について説明します。

ファイルシステムのリテラルコピーを作成する

dd コマンドでは、UFS ファイルシステムのリテラル (ブロックレベル) コピーを別のファイルシステムやテープに作成します。デフォルトでは、dd コマンドはその標準入力を標準出力にコピーします。

注 - 可変長テープドライブで dd コマンドを使用するときは、必ず適切なブロックサイズを指定してください。

標準入力、標準出力、またはその両方の代わりに、デバイス名を指定できます。次の例では、フロッピーディスクの内容が /tmp ディレクトリ内のファイルにコピーされます。

```
$ dd < /floppy/floppy0 > /tmp/output.file
2400+0 records in
2400+0 records out
```

dd コマンドは、読み込みブロック数と書き込みブロック数をレポートします。+ の次の数値は、部分的にコピーされたブロックの数です。デフォルトのブロックサイズは 512 バイトです。

dd コマンドの構文は、他のほとんどのコマンドとは異なっています。オプションは *keyword=value* のペアで指定します。この場合、*keyword* は設定したいオプションで、*value* はそのオプションの引数です。たとえば、標準入力と標準出力を次の構文に置き換えることができます。

```
$ dd if=input-file of=output-file
```

たとえば、上記の例のリダイレクト記号の代わりに *keyword=value* の形式で指定するには、次のように入力します。

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

▼ ディスクをクローン化する方法 (dd)

1. コピー元とコピー先のディスクが同じディスクジオメトリを持っているかどうかを確認します。
2. スーパーユーザーになります。
3. /reconfigure ファイルをシステムに作成します。これによって、システムは追加されるクローンディスクをリブート時に認識します。

```
# touch /reconfigure
```

4. システムをシャットダウンします。

```
# init 0
```

5. クローンディスクをシステムに接続します。
6. システムをブートします。

```
ok boot
```

7. dd コマンドを使用してマスターディスクをクローンディスクにコピーします。

```
# dd if=/dev/dsk/device-name of=/dev/dsk/device-name bs=blocksize
```

| | |
|--------------------------------------|---|
| <code>if=/dev/dsk/device-name</code> | マスターディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。 |
| <code>of=/dev/dsk/device-name</code> | クローンディスクデバイスのオーバーラップスライスを指定する。通常はスライス 2。 |
| <code>bs=blocksize</code> | ブロックサイズ (128K バイト、256K バイトなど)。ブロックサイズの値を大きくすると、コピーに要する時間を短縮できる。 |

8. 新しいファイルシステムをチェックします。

```
# fsck /dev/rdsk/device-name
```

9. クローンディスクのルート (/) ファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /mnt
```

10. クローンディスクの `/etc/vfstab` を編集して、正しいデバイス名を参照するようにします。

たとえば、`c0t3d0` をすべて `c0t1d0` に変更します。

11. クローンディスクのルート (/) ファイルシステムをマウント解除します。

```
# umount /mnt
```

12. システムを停止します。

```
# init 0
```

13. クローンディスクからシングルユーザーモードにブートします。

```
# boot disk1 -s
```

注 - `installboot` コマンドをクローンディスクで実行する必要ありません。これは、ブートブロックがオーバーラップスライスの一部としてコピーされるためです。

14. クローンディスクの構成を解除します。

```
# sys-unconfig
```

構成を解除すると、システムが停止します。

15. 再びクローンディスクからブートし、ホスト名や時間帯などのシステム情報を与えます。

```
# boot diskn
```

16. スーパーユーザーとしてログインして、一度システムがブートした後のシステム情報を確認します。

```
hostname console login:
```

例 — ディスクのクローン化 (dd)

```
# init 0
ok boot
# dd if=/dev/dsk/c0t0d0s2 of=/dev/dsk/c0t2d0s2 bs=128k
# fsck /dev/rdisk/c0t2d0s2
# mount /dev/dsk/c0t2d0s2 /mnt
# cd /mnt/etc
# vi vfstab
(新しいディスクのエントリを変更)
# cd /
# umount /mnt
# init 0
# boot disk2 -s
# sys-unconfig
# boot disk2
```

cpio 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

cpio (コピーインとコピーアウト) コマンドを使用して、個々のファイル、ファイルグループ、またはファイルシステム全体をコピーできます。この節では、cpio コマンドを使用してファイルシステム全体をコピーする方法について説明します。

cpio コマンドは、ファイルのリストを取り出して1つの大型出力ファイルにコピーするアーカイブプログラムです。また、復元しやすいように、個々のファイルの間にヘッダを挿入します。cpio コマンドを使用すると、ファイルシステム全体を別のスライス、別のシステム、またはテープやフロッピーディスクなどの媒体デバイスにコピーできます。

cpio コマンドは媒体の終わりを認識し、別のボリュームの挿入を促すプロンプトを表示するので、複数のテープやフロッピーディスクが必要なアーカイブを作成するには最も (ufsdump よりも) 効率のよいコマンドです。

cpio の使用時には、しばしば `ls` や `find` などのコマンドを使用し、コピーしたいファイルを選択して、出力を cpio コマンドにパイプします。

▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)

1. スーパーユーザーになります。
2. 目的のディレクトリに移動します。

```
# cd /filesystem1
```

3. `find` コマンドと `cpio` コマンドを組み合わせて実行し、ディレクトリツリーを `filesystem1` から `filesystem2` へコピーします。

```
# find . -print -depth | cpio -pdmu /filesystem2
```

| | |
|--------|-------------------------------------|
| . | 現在の作業ディレクトリ内で処理を始める。 |
| -print | ファイル名を出力する。 |
| -depth | ディレクトリ階層を下位へたどってバックアップ中にファイル名を出力する。 |
| -p | ファイルのリストを作成する。 |
| -d | 必要に応じてディレクトリを作成する。 |
| -m | ディレクトリ上で正しい変更時刻を設定する。 |

指定したディレクトリ名からファイルがコピーされ、シンボリックリンクは保持されます。

また、`-u` オプションも指定できます。このオプションは、無条件にコピーを実行します。`-u` オプションを指定しないと、古いファイルは、新しいファイルで置換されません。これは、ディレクトリを確実にコピーしたいとき、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. 該当する場合は、ソースディレクトリを削除します。

```
# rm -rf /filesystem1
```

例 — ファイルシステム間でディレクトリをコピーする (cpio)

```
# cd /data1
# find . --print --depth | cpio --pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

詳細は、cpio(1) のマニュアルページを参照してください。

ファイルとファイルシステムをテープにコピーする

pax、tar、cpio コマンドを使用すると、ファイルとファイルシステムをテープにコピーできます。どのコマンドを選択するかは、コピーする目的に応じて異なります。3つのコマンドはすべて raw デバイスを使用するので、使用する前にテープ上でファイルシステムをフォーマットまたは作成する必要はありません。

表 37-3 cpio、pax、tar コマンドの長所と短所

| コマンド | 機能 | 長所 | 短所 |
|------|---|--|---|
| pax | POSIX 準拠システムとの間でファイルをコピーする場合、または複数のテープボリュームが必要となる、ファイル、特殊ファイル、またはファイルシステムのコピー | <ul style="list-style-type: none"> ■ POSIX 準拠システムに対する可搬性は、tar コマンドや cpio コマンドよりもよい。 ■ マルチベンダーサポート | tar コマンドの欠点を参照。ただし、pax は、複数のテープボリュームを作成できる。 |
| tar | 1 本のテープへの、ファイルやディレクトリのサブツリーのコピー | <ul style="list-style-type: none"> ■ ほとんどの UNIX オペレーティングシステムで利用できる。 ■ パブリックドメインバージョンもすぐに利用できる。 | <ul style="list-style-type: none"> ■ ファイルシステムの境界を認識しない。 ■ 絶対パス名の長さが 255 文字を超えることができない。 ■ 空のディレクトリや特殊ファイル (デバイスファイルなど) をコピーしない。 ■ 複数のテープボリュームを作成する場合は使用できない。 |
| cpio | SunOS 5.7 システムから SunOS 4.0/4.1 システムにファイルをコピーする場合、または複数のテープボリュームが必要となる、ファイル、特殊ファイル、またはファイルシステムのコピー | <ul style="list-style-type: none"> ■ tar よりも効率的に、データをテープに書き込む。 ■ 復元時、テープ中の不良箇所をスキップする。 ■ 異なるシステムタイプ間の可搬性のために、異なるヘッダーフォーマット (tar、ustar、crc、odc、bar) でファイルを書き込むオプションを提供する。 ■ 複数のテープボリュームを作成する。 | |

使用するテープドライブとデバイス名は、各システムのハードウェアと構成によって異なります。テープドライブとデバイス名の詳細については、675ページの「使用する媒体の選択」を参照してください。

tar を使用してファイルをテープにコピーする

tar コマンドでファイルをテープにコピーする前に、次のことを知っておかなければなりません。

- tar コマンドに `-c` オプションを指定してファイルをテープにコピーすると、テープに入っているすべての既存のファイルまたはテープの現存の位置以降にある既存のファイルすべてが破壊(上書き)される。
- ファイル名の一部にファイル名置換ワイルドカード文字(?) と (*) を使用して指定できる。たとえば、接尾辞 `.doc` が付いたすべての文書をコピーするには、ファイル名引数として `*.doc` と入力する。
- tar アーカイブからファイルを抽出するときには、ファイル名置換ワイルドカードは使用できない。

▼ ファイルをテープにコピーする方法 (tar)

1. コピーしたいファイルが入っているディレクトリに変更します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. tar コマンドを使用してファイルをテープにコピーします。

```
$ tar cvf /dev/rmt/n filename ...
```

| | |
|---------------------------|------------------------------------|
| <code>c</code> | アーカイブが作成されるように指定する。 |
| <code>v</code> | アーカイブされるたびに、各ファイルの名前を表示する。 |
| <code>f /dev/rmt/n</code> | アーカイブを指定したデバイスまたはファイルに書き込むように指定する。 |
| <code>filename ...</code> | コピーしたいファイルとディレクトリを指定する。 |

指定した名前のファイルがテープにコピーされ、テープ上の既存のファイルが上書きされます。

4. テープをドライブから取り出して、テープラベルにファイル名を記入します。
5. テープの内容を表示する `t` オプションを指定し、`tar` コマンドを使用して、コピーされたファイルがテープに入っているかどうかを確認します。**tar** テープ上のファイルを表示する方法についての詳細は、657ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

```
$ tar tvf /dev/rmt/n
```

例 — ファイルをテープにコピーする (tar)

次の例では、3つのファイルがテープドライブ0のテープにコピーされます。

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 2 tape blocks
a reports/reportB 5 tape blocks
a reports/reportC 6 tape blocks
$ tar tvf /dev/rmt/n
```

▼ テープ上のファイルのリストを表示する方法 (tar)

1. テープをテープドライブに挿入します。
2. `tar` コマンドを使用してテープの内容を表示します。

```
$ tar tvf /dev/rmt/n [filename]
```

| | |
|---------------------------|---|
| <code>t</code> | テープ上のファイルの内容一覧が表示される。 |
| <code>v</code> | <code>t</code> オプションと併用すると、テープ上のファイルに関する詳細情報が表示される。 |
| <code>f /dev/rmt/n</code> | テープデバイスを示す。 |
| <code>filename ...</code> | リスト表示したいファイルとディレクトリを指定する。 |

例 — テープ上のファイルのリストを表示する (tar)

次の例では、ドライブ 0 のテープに含まれているファイルのリストを表示します。

```
$ tar tvf /dev/rmt/0
drwxr-xr-x 101/10      0 Nov  6 16:31 1996 reports/
-rw-r--r-- 101/10      0 Nov  6 16:31 1996 reports/reportA
-rw-r--r-- 101/10      0 Nov  6 16:31 1996 reports/reportB
-rw-r--r-- 101/10      0 Nov  6 16:31 1996 reports/reportC
```

▼ テープからファイルを取り出す方法 (tar)

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. `tar` コマンドを使用してテープからファイルを取り出します。

```
$ tar xvf /dev/rmt/n [filename ...]
```

| | |
|---------------------------|---|
| <code>x</code> | 指定したアーカイブファイルからファイルを抽出するように指定する。指定したドライブのテープに含まれるすべてのファイルが現在のディレクトリにコピーされる。 |
| <code>v</code> | 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。 |
| <code>f /dev/rmt/n</code> | アーカイブが入っているテープデバイスを示す。 |
| <code>filename</code> | 取り出すファイルを指定する。 |

- 現在のディレクトリの内容をリストして、ファイルがコピーされていることを確認します。

```

$ ls -l

```

例 — テープ上のファイルを取り出す (tar)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出します。

```

$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l

```

注 - テープから抽出されるファイル名は、アーカイブに格納されるファイル名と同一でなければなりません。ファイルの名前やパス名が不明な場合は、まずテープ上のファイルのリストを表示します。手順については、657ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

詳細は、tar(1) のマニュアルページを参照してください。

pax を使用してファイルをテープにコピーする

この節では、pax コマンドでファイルをコピーする方法について説明します。

▼ ファイルをテープにコピーする方法 (pax)

1. コピーしたいファイルが入っているディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. pax コマンドを使用してファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/0 filename ...
```

| | |
|---------------|-------------------------|
| -w | 書き込みモードを有効にする。 |
| -f /dev/rmt/0 | テープドライブを識別する。 |
| filename ... | コピーしたいファイルとディレクトリを指定する。 |

4. ファイルがテープにコピーされていることを確認します。

```
$ pax -l -f /dev/rmt/0
```

5. ドライブからテープを取り出して、テープラベルにファイル名を記入します。

例 — ファイルをテープにコピーする (pax)

```
$ pax -w -f /dev/rmt/0 .  
$ pax -f /dev/rmt/0  
filea  
fileb  
filec
```

詳細については、pax(1) のマニュアルページを参照してください。

▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)

1. 書き込み保護されていないテープをテープドライブに挿入します。
2. `ls` コマンドと `cpio` コマンドを使用してファイルをテープにコピーします。

```
$ ls | cpio -oc > /dev/rmt/n
```

| | |
|------------------------------|---|
| <code>ls</code> | <code>cpio</code> コマンドにファイル名のリストを渡す。 |
| <code>cpio -oc</code> | <code>cpio</code> がコピーアウトモード (-o) で動作し、ASCII 文字形式 (-c) でヘッダー情報を書き込むように指定する。これにより他のベンダーのシステムと可搬性を保つ。 |
| <code>> /dev/rmt/n</code> | 出力ファイルを指定する。 |

ディレクトリ内のすべてのファイルは、指定したドライブ内のテープにコピーされ、テープ上の既存のファイルが上書きされます。コピーされた合計ブロック数が表示されます。

3. 次の `cpio` コマンドで、ファイルがテープにコピーされていることを確認します。

```
$ cpio -civt < /dev/rmt/0
```

4. テープをドライブから取り出して、テープラベルにファイル名を記入します。

例 — ディレクトリ内のすべてのファイルをテープにコピーする (cpio)

次の例では、ディレクトリ `/export/home/kryten` 内のすべてのファイルを、テープドライブ 0 のテープにコピーします。

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
```

(続く)

```

8 blocks
$ cpio -civt < /dev/rmt/0
drwxr-xr-x 2 kryten users 0 Jun 9 15:56 1998, letters
drwxr-xr-x 2 kryten users 0 Jun 9 15:56 1998, memos
drwxr-xr-x 2 kryten users 0 Jun 9 15:55 1998, reports
8 blocks
$

```

▼ テープ上のファイルのリストを表示する方法 (cpio)

注 - 内容一覧を表示すると、cpio コマンドはアーカイブ全体を処理しなければならないので、アーカイブファイルの読み込みに長時間かかります。

1. テープをテープドライブに挿入します。
2. cpio コマンドを使用してテープ上のファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/n
```

| | |
|--------------|--|
| -c | ファイルを ASCII 文字形式で読み込むように指定する。 |
| -i | cpio がコピーインモードで動作することを指定する (この時点ではファイルをリストするだけ)。 |
| -v | ls -l コマンドと同様の形式で出力を表示する。 |
| -t | 指定したテープドライブ内のテープ上にあるファイルの内容一覧が表示される。 |
| < /dev/rmt/n | 既存の cpio アーカイブの入力ファイルを指定する。 |

例 — テープ上のファイルのリストを表示する (cpio)

次の例では、ドライブ 0 のテープに含まれているファイルのリストを表示します


```
$ cpio -civt < /dev/rmt/0
drwxr-xr-x 2 rimmer users 0 Jun  9 16:07 1998, answers
drwxr-xr-x 2 rimmer users 0 Jun  9 16:07 1998, sc.directives
drwxr-xr-x 2 rimmer users 0 Jun  9 16:07 1998, tests
8 blocks
```

▼ テープからすべてのファイルを取り出す方法 (cpio)

相対パス名を使用してアーカイブを作成した場合、入力ファイルはそれを取り出すときに現在のディレクトリ内のディレクトリとして作成されます。ただし、絶対パス名を指定してアーカイブを作成した場合は、それと同じ絶対パス名を使用してシステム上でファイルが再び作成されます。



注意 - 絶対パス名を使用すると、自分のシステム上にある元のファイルを上書きすることになるので危険です。

1. ファイルを入れたいディレクトリに変更します。
2. テープをテープドライブに挿入します。
3. cpio コマンドを使用して、すべてのファイルをテープから現在のディレクトリにコピーします。

```
$ cpio -icvd < /dev/rmt/n
```

| | |
|--------------|---------------------------------|
| -i | テープの内容を読み込む。 |
| -c | ファイルを ASCII 文字形式で読み込むように指定する。 |
| -v | 取り出されたファイルを ls コマンドと同様の形式で表示する。 |
| -d | 必要に応じて、ディレクトリを作成する。 |
| < /dev/rmt/n | 入力ファイルを指定する。 |

- 現在のディレクトリの内容を表示して、ファイルがコピーされていることを確認します。

```
$ ls -l
```

例 — テープからすべてのファイルを取り出す (cpio)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出します。

```
$ cd /var/tmp
$ cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

▼ 指定したファイルをテープから取り出す方法 (cpio)

- ファイルを置きたいディレクトリに移動します。
- テープをテープドライブに挿入します。
- cpio コマンドを使用してテープからファイルのサブセットを取り出します。

```
$ cpio -icv "file" < /dev/rmt/n
```

| | |
|----|---------------------------------|
| -i | テープの内容を読み込む。 |
| -c | ファイルを ASCII 文字形式で読み込むように指定する。 |
| -v | 取り出されたファイルを ls コマンドと同様の形式で表示する。 |

`"*file"` パターンに一致するすべてのファイルを現在のディレクトリにコピーするように指定する。複数のパターンを指定できるが、個々のパターンを二重引用符で囲まなければならない。

`< /dev/rmt/n` 入力ファイルを指定する。

4. 現在のディレクトリの内容を表示して、ファイルがコピーされていることを確認します。

```
$ ls -l
```

例 — 指定したファイルをテープから取り出す (cpio)

次の例では、末尾に接尾辞 `chapter` が付いているすべてのファイルをドライブ 0 のテープから取り出します。

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

詳細は、`cpio(1)` のマニュアルページを参照してください。

▼ ファイルをリモートテープドライブにコピーする方法 (tar と dd)

1. リモートテープドライブを使用するには、次の前提条件を満たしている必要があります。
 - ローカルホスト名 (および、オプションで、コピーしているユーザーのユーザー名) が、リモートシステムの `/etc/hosts.equiv` ファイル内になければならない。あるいは、コピーしているユーザーは、リモートマシン上の自分のホームディレクトリをアクセス可能にしておかなければならない。さら

に、\$HOME/.rhosts 内にローカルマシン名がなければならない。詳細については、hosts.equiv(4) を参照。

- リモートシステムのエントリがローカルシステムの /etc/inet/hosts ファイル内またはネームサービスの hosts ファイル内になければならない。

2. リモートコマンドを実行するための適切なアクセス権を持っているかどうかをテストするには、次のように入力します。

```
$ rsh remotehost echo test
```

「test」と表示された場合、リモートコマンドを実行するためのアクセス権を持っています。「Permission denied」と表示された場合、上記の手順 1 の内容を確認してください。

3. tar コマンドと dd コマンドを使用して、ファイルをリモートテープドライブにコピーします。

```
$ tar cf - files | rsh remotehost dd of=/dev/rmt/n obs=blocksize
```

| | |
|------------------|---|
| tar cf | テープアーカイブを作成し、テープデバイスを指定する。 |
| - (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。 |
| files | コピーするファイル |
| rsh remotehost | tar コマンドの出力がパイプを通してリモートシェルに渡され、ファイルがコピーされる。 |
| dd of=/dev/rmt/n | 出力デバイスを指定する。 |
| obs=blocksize | ブロック係数を指定する。 |

4. テープをドライブから取り出して、テープラベルにファイル名を記入します。

例 — ファイルをリモートテープドライブにコピーする (tar と dd)

```
# tar cvf - * | rsh mercury dd of=/dev/rmt/0 bs=126b
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

▼ ファイルをリモートテープドライブから取り出す方法

1. 一時ディレクトリに変更します。

```
$ cd /var/tmp
```

2. tar コマンドと dd コマンドを使用して、ファイルをリモートテープドライブに抽出します。

```
$ rsh remotehost dd if=/dev/rmt/n | tar xvBpf -
```

rsh remotehost

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されるリモートシェル。

dd if=/dev/rmt/n

入力デバイスを指定する。

| tar xvBpf -

dd コマンドの出力は tar コマンドにパイプされ、復元されたファイルに使用される。

3. ファイルが抽出されたかどうかを確認します。

```
$ ls -l /var/tmp
```

例 — ファイルをリモートテープドライブから抽出する

```
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -  
x answers/, 0 bytes, 0 tape blocks  
x answers/test129, 48 bytes, 1 tape blocks  
20+0 records in  
20+0 records out  
x sc.directives/, 0 bytes, 0 tape blocks  
x sc.directives/sc.190089, 77 bytes, 1 tape blocks  
x tests/, 0 bytes, 0 tape blocks  
x tests/test131, 84 bytes, 1 tape blocks  
$ ls -l /var/tmp
```

ファイルとファイルシステムをフロッピーディスクにコピーする

ファイルやファイルシステムをフロッピーディスクにコピーする前に、フロッピーディスクをフォーマットしなければなりません。フロッピーディスクをフォーマットする方法については、第 13 章を参照してください。

tar コマンドを使用して、UFS ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーします。

UFS ファイルを複数のフォーマット済みフロッピーディスクにコピーする必要がある場合は、cpio コマンドを使用します。cpio は媒体の終りを認識し、次のボリュームの挿入を促すプロンプトを表示します。

注 - ボリューム管理の関係で、cpio コマンドを使用して UFS ファイルを複数のフォーマット済みフロッピーディスクにコピーする手順は単純ではありません。

両面高密度 3.5 インチフロッピーディスクを使用してください (フロッピーディスクには「DS、HD」マークが付いています)。

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ァイルをフォーマット済みフロッピーディスクにコピーすると、フロッピーディスク上の既存のファイルは破壊 (上書き) される。
- すでに tar イメージが入っているフロッピーディスクはマウントできない。

▼ ファイルを1枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)

1. コピーしたいファイルが入っているディレクトリに移動します。
2. 書き込み保護されていないフォーマット済みフロッピーディスクをドライブに挿入します。
3. `volcheck` コマンドを使用してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

4. フロッピーディスク上のファイルシステムをすべてマウント解除し、再度フォーマットします。

```
$ fdformat -U /vol/dev/aliases/floppy0
```

5. `tar` コマンドを使用してファイルをフロッピーディスクにコピーします。

```
$ tar cvf /vol/dev/rdiskette0/unlabeled filename ...
```

指定した名前のファイルがフロッピーディスクにコピーされ、フロッピーディスク上の既存のファイルが上書きされます

6. フロッピーディスクの内容を表示する `-t` オプションをつけて `tar` コマンドを使用し、コピーしたファイルがフロッピーディスクに入っているかどうかを検査します。ファイルのリストを表示する方法についての詳細は、670ページの「フロッピーディスク上のファイルのリストを表示する方法 (tar)」を参照してください。

```
$ tar tvf /vol/dev/rdiskette0/unlabeled
```

7. フロッピーディスクをドライブから取り出します。
8. ファイル名をフロッピーディスクラベルに記入します。

例 — ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする (tar)

次の例では、2つのファイルをフロッピーディスクにコピーします。

```
$ cd /home/smith
$ ls evaluation*
evaluation.doc  evaluation.doc.backup
$ tar cvf /vol/dev/rdiskette0/unlabeled evaluation*
a evaluation.doc 86 blocks
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/rdiskette0/unlabeled
```

▼ フロッピーディスク上のファイルのリストを表示する方法 (tar)

1. フロッピーディスクをドライブに挿入します。
2. `volcheck` を実行してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

3. `tar` コマンドを使用してフロッピーディスク上のファイルのリストを表示します。

```
$ tar tvf /vol/dev/rdiskette0/unlabeled
```

例 — フロッピーディスク上のファイルのリストを表示する (tar)

次の例では、フロッピーディスクの上のファイルを表示します。


```
$ tar tvf /vol/dev/rdiskette0/unlabeled
rw-rw-rw-6693/10  44032 Oct 23 14:54 1996 evaluation.doc
rw-rw-rw-6693/10  43008 Oct 23 14:47 1996 evaluation.doc.backup
$
```

詳細は tar(1) のマニュアルページを参照してください。

複数のボリュームを扱いたい場合は、cpio コマンドを使用してください。tar コマンドは1つのボリュームに対して使用できるユーティリティです。

▼ ファイルをフロッピーディスクから取り出す方法 (tar)

1. ファイルを置きたいディレクトリに移動します。
2. フロッピーディスクをドライブに挿入します。
3. volcheck を実行してフロッピーディスクを使用できるようにします。

```
$ volcheck
```

4. tar コマンドを使用してファイルをフロッピーディスクから取り出します。

```
$ tar xvf /vol/dev/rdiskette0/unlabeled
```

フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

5. 現在のディレクトリの内容を表示して、ファイルが取り出されたことを確認します。

```
$ ls -l
```

6. フロッピーディスクをドライブから取り出します。

例 — ファイルをフロッピーディスクから取り出す (tar)

次の例では、フロッピーディスクからすべてのファイルを取り出します。

```
$ /home/smith/Evaluations
$ tar xvf /vol/dev/rdiskette0/unlabeled
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

tar コマンドを使用してフロッピーディスクから個々のファイルを取り出します。

```
$ tar xvf /vol/dev/rdiskette0/unlabeled evaluation.doc
x evaluation.doc, 44032 bytes, 86 tape blocks
$ ls -l
```

指定した名前のファイルがフロッピーディスクから取り出され、現在の作業ディレクトリに格納されます。

▼ ファイルを複数のフロッピーディスクにアーカイブする方法

大量のファイルやファイルシステムをフロッピーディスクにコピーする場合は、一杯になったフロッピーディスクを別のフォーマット済みフロッピーディスクと交換するように促すプロンプトを表示させることができます。cpio コマンドにはこの機能があります。使用する cpio コマンドはファイルをテープにコピーする場合と同じですが、テープデバイス名ではなくデバイスとして /vol/dev/aliases/floppy0 を指定します。cpio の使用方法については、661ページの「ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)」を参照してください。

ファイルを別のヘッダー形式でコピーする

SunOS 5.7 の cpio コマンドを使用して作成したアーカイブには、SunOS の旧リリースとの互換性がない場合があります。cpio コマンドを使用すると、他の複数の形式で読み込めるアーカイブを作成できます。これらの形式は、-H オプションと次のいずれかの引数で指定します。

- crc または CRC - チェックサム付きの ASCII ヘッダー
- ustar または USTAR - IEEE/P1003 データ交換
- tar または TAR - tar のヘッダーと形式
- odc - 小型デバイス番号付きの ASCII ヘッダー
- bar - bar のヘッダーと形式

ヘッダーオプションを使用する場合の構文は次のとおりです。

```
cpio -o -H header-option < file-list > output-archive
```

▼ SunOS の旧リリース用のアーカイブを作成する方法

cpio コマンドを使用してアーカイブを作成します。

```
$ cpio -oH odc < file-list > /dev/rmt/n
```

-H オプションは、入力に対して出力の場合と同じ意味を持ちます。-H オプションを使用してアーカイブを作成した場合は、読み込むときにも同じオプションを使用しないと、次のように cpio コマンドが失敗に終わります。

例 — SunOS の旧リリース用にアーカイブを作成する

```
$ find . -print | cpio -oH tar > /tmp/test
113 blocks
$ cpio -iH bar < /tmp/test
cpio: Invalid header "bar" specified
USAGE:
    cpio -i[bcdfkmrstuvBSV6] [-C size] [-E file] [-H hdr]
[-I file [-M msg]] [-R id] [patterns]
    cpio -o[acvABLV] [-C size] [-H hdr] [-O file [-M msg]]
    cpio -p[adlmuvLV] [-R id] directory
```

各種オプションを使用してアーカイブを作成するときには、必ず媒体のラベルにアーカイブ上のファイル名やファイルシステム名といっしょにコマンド構文を記入してください。

アーカイブの作成時にどの `cpio` オプションを使用したかがわからない場合は、各種オプションをいろいろ組み合わせてみなければ、どの方法でアーカイブを読み込めるかがわかりません。

オプションのリストについては、`cpio(1)` のマニュアルページを参照してください。

bar コマンドで作成したファイルを取り出す

SunOS 4.0/4.1 の `bar` コマンドを使用してアーカイブしたファイルをフロッピーディスクから取り出すには、`cpio` の `-H bar` オプションを使用します。

注・ファイルを取り出すには、`-H bar` オプションと `-i` をいっしょに使用しなければなりません。`bar` ヘッダーオプションを使用してファイルを作成することはできません。

▼ bar ファイルをフロッピーディスクから取り出す方法

1. ファイルを置きたいディレクトリに移動します。
2. フロッピーディスクをドライブに挿入します。
3. `volcheck` を実行して、フロッピーディスクを使用できるようにします。

```
$ volcheck
```

4. `cpio` コマンドを使用してフロッピーディスクから `bar` ファイルを取り出します。

フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

```
$ cpio -ivH bar < /vol/dev/rdiskette/unlabeled
```

テープドライブの管理手順

この章では、テープドライブを管理する方法について説明します。

この章で説明する手順は次のとおりです。

- 678ページの「テープドライブの状態を表示する方法」
- 680ページの「磁気テープカートリッジのたるみを直す方法」
- 680ページの「磁気テープカートリッジを巻き戻す方法」

使用する媒体の選択

通常は、次の媒体を使用して Solaris システムのバックアップを作成します。

- 1/2 インチのリールテープ
- 1/4 インチのストリームカートリッジテープ
- 8 mm のカートリッジテープ
- 4 mm のカートリッジテープ (DAT)

フロッピーディスクを使用してバックアップを実行することもできますが、時間がかかり煩雑です。

どの媒体を選択するかは、媒体をサポートする機器とファイルの格納に使用する媒体 (通常はテープ) の可用性によって決まります。バックアップはローカルシステムから実行しなければなりません、ファイルはリモートデバイスに書き込めます。

表 38-1 に、ファイルシステムの典型的なバックアップ媒体と、その記憶容量を示します。

表 38-1 媒体の記憶容量

| 媒体 | 容量 ¹ |
|-----------------------------------|-----------------------|
| 1/2 インチ リールテープ | 140 Mbytes (6250 bpi) |
| 2.5G バイト 1/4 インチ カートリッジ (QIC) テープ | 2.5G バイト |
| DDS3 4-mm カートリッジ テープ (DAT) | 12 ~ 24G バイト |
| 14-Gbyte 8-mm カートリッジ テープ | 14G バイト |
| DLT™ 7000 1/2 インチ カートリッジ テープ | 35 ~ 70G バイト |

1. 容量は、ドライブのタイプとテープに書き込むデータによって異なります。

バックアップデバイス名

バックアップに使用するテープまたはフロッピーディスクドライブに論理デバイス名を与えて指定します。この名前は、「raw」のデバイスファイルが入っているサブディレクトリを指し、ドライブの論理デバイス番号が含まれています。テープドライブの命名規則では、物理デバイス名ではなく論理デバイス名を使用します。表 38-2 は、この命名方式を示しています。

表 38-2 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 | /dev/rmt/ <i>n</i> |
| フロッピーディスク | /vol/dev/rdiskette0/unlabeled |

通常は、図 38-1 のようにテープドライブデバイスを指定します。

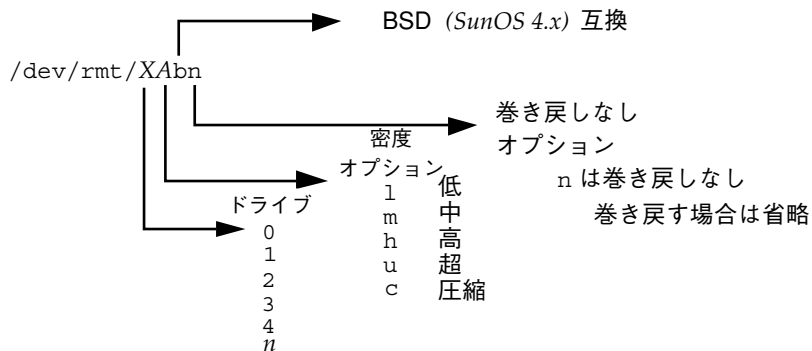


図 38-1 テープドライブデバイス名

密度を指定しないと、テープドライブは通常それがサポートする一番高い密度で書き込みます。ほとんどの SCSI ドライブはテープ上の密度やフォーマットを自動的に検出し、それに従って読み取りを実行します。ドライブでサポートされる密度を調べるには、/dev/rmt サブディレクトリを見てください。このディレクトリには、各テープで異なる出力密度をサポートするためのテープデバイスファイルのセットが含まれています。

SCSI コントローラは、最大7台の SCSI テープドライブを持つことができます。

テープドライブのデフォルト密度を指定する

通常は、テープドライブを 0 から n までの論理デバイス番号で指定します。表 38-3 に、デフォルトの密度設定を使用してテープデバイス名を指定する方法を示します。

表 38-3 テープドライブのデフォルト密度指定

| 指定するドライブ | 使用する番号 |
|------------------|-------------|
| 第 1 のドライブ、巻き戻し | /dev/rmt/0 |
| 第 1 のドライブ、巻き戻しなし | /dev/rmt/0n |
| 第 2 のドライブ、巻き戻し | /dev/rmt/1m |
| 第 2 のドライブ、巻き戻しなし | /dev/rmt/1n |

デフォルトでは、ドライブはその「推奨」密度で書き込みますが、これは一般にそのドライブでサポートされる最大密度です。テープデバイスを指定しなければ、コマンドはデバイスでサポートされるデフォルト密度でドライブ番号 0 に書き込みます。

テープドライブに別の密度を指定する

テープを特定の密度しかサポートされないテープドライブが付いたシステムに転送するには、目的の密度で書き込むデバイス名を指定します。表 38-4 は、テープドライブに別の密度を指定する方法を示しています。

表 38-4 テープドライブに別の密度を指定する

| 指定 | 番号 |
|----------------------|--------------|
| 第 1 のドライブ、巻き戻し、低密度 | /dev/rmt/0l |
| 第 1 のドライブ、巻き戻しなし、低密度 | /dev/rmt/0ln |
| 第 2 のドライブ、巻き戻し、中密度 | /dev/rmt/1m |
| 第 2 のドライブ、巻き戻しなし、中密度 | /dev/rmt/1mn |

デバイス名と密度文字については、図 38-1 を参照してください。

テープドライブの状態を表示する

mt コマンドの status オプションを使用すると、テープドライブに関する状態情報を表示できます。mt コマンドは、/kernel/drv/st.conf ファイルに記述されているすべてのテープドライブの情報を表示します。

▼ テープドライブの状態を表示する方法

1. 情報を表示したいドライブにテープをロードします。
2. mt コマンドを使用してテープドライブ情報を表示します。


```
# mt -f /dev/rmt/n status
```

3. テープドライブ番号を **1**、**2**、**3** というように置き換えて手順 **1** と **2** を繰り返し、使用可能なすべてのテープドライブに関する情報を表示します。

例 — テープドライブの状態を表示する

次の例は、QIC-150 テープドライブ (/dev/rmt/0) と Exabyte テープドライブ (/dev/rmt/1) の状態を示しています。

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0   retries= 0
file no= 0   block no= 0
```

次の方法を使用すると、システムを手早くポーリングしてすべてのテープドライブを検査できます。

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
  sense key(0x0)= No Additional Sense   residual= 0   retries= 0
  file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

磁気テープカートリッジの取り扱い

テープの読み込み中にエラーが発生した場合は、テープのたるみを直し、テープドライブを掃除してからやり直してください。

▼ 磁気テープカートリッジのたるみを直す方法

mt コマンドを使用して磁気テープカートリッジのたるみを直します。

```
$ mt -f /dev/rmt/n retension
```

例 — 磁気テープドライブのたるみを直す方法

次の例では、ドライブ /dev/rmt/1 内のテープのたるみを直します。

```
$ mt -f /dev/rmt/1 retension
$
```

注 - QIC 以外のテープドライブのたるみは直さないでください。

▼ 磁気テープカートリッジを巻き戻す方法

磁気テープカートリッジを巻き戻すには、mt コマンドを使用します。

```
$ mt -f /dev/rmt/n rewind
```

例 — 磁気テープカートリッジを巻き戻す

次の例では、ドライブ /dev/rmt/1 内のテープを巻き戻します。

```
$ mt -f /dev/rmt/1 rewind
```

ドライブの管理と媒体処理のガイドライン

バックアップテープは読み込めなければ役に立ちません。そこで、テープドライブが正常に動作するように定期的に掃除してチェックするとよいでしょう。テープド

ライブのクリーニング手順については、ハードウェアのマニュアルを参照してください。次の2つの方法のうち、どちらかを使ってテープハードウェアをチェックできます。

- テープにファイルをコピーし、読み込んで、コピーをオリジナルと比較する。
- `ufsdump` コマンドの `-v` オプションを使用すると、媒体の内容をソースファイルシステムと比較して検査できます。`-v` オプションを機能させるには、ファイルシステムをマウント解除するか、完全にアイドル状態にしなければなりません。

ハードウェアは、システムからレポートされないような障害を起こす可能性があるので注意してください。

バックアップ後は、必ずテープにラベルを付けてください。第33章で説明したようなバックアップ方法を採用した場合は、ラベルに「テープA」、「テープB」というようにしておく必要があります。このラベルは変更しないでください。バックアップを実行するたびに別のテープラベルを作成して、バックアップ日付、マシン名、バックアップを作成したファイルシステム、バックアップレベル、テープ番号(複数のボリュームにまたがる場合は n 本のうちの1本目)、サイト特有の情報を入力します。テープは、磁気を発生させる機器から離れた埃のない安全な場所に保管してください。サイトによっては、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ています。

各ジョブ(バックアップ)がどの媒体(テープボリューム)に格納されているかということと、各バックアップファイルがどこに保管されているかを記録したログを作成し、管理する必要があります。

64 ビット Solaris オペレーティング環境

この付録は、Solaris 7 で利用できる 32 ビットと 64 ビットの Solaris オペレーティング環境上のシステムを管理するシステム管理者を対象としています。

64 ビット Solaris オペレーティング環境の概要

Solaris 7 では、次に示す 2 種類の環境を SPARC または x86 のプラットフォームで利用できます。

- 次の作業を可能にする、32 ビット Solaris アプリケーションとオペレーティング環境
 - 64 ビットアプリケーションの開発 (SPARC プラットフォームのみ)
 - 既存の多くの 32 ビットアプリケーションの実行
- 次の作業を可能にする、64 ビット Solaris アプリケーションとオペレーティング環境 (SPARC プラットフォームのみ)
 - 64 ビットアプリケーションの開発
 - 新しい 64 ビットアプリケーションでの大規模アドレス空間の操作
 - 既存の多くの 32 ビットアプリケーションの実行

現時点では、64 ビット Solaris 7 オペレーティング環境をサポートできるプラットフォームは、UltraSPARC™ プロセッサを持つシステムだけです。その他のプラット

フォームや旧リリースの Solaris では、64 ビットアプリケーションは開発できません。

注 - この付録では、UltraSPARC プロセッサを搭載したシステムのことを「UltraSPARC システム」と呼びます。

64 ビット Solaris アプリケーション環境を使用する必要があるか

32 ビットまたは 64 ビットの Solaris 7 システム上で 64 ビットアプリケーションを開発するか、UltraSPARC システム上で 64 ビットアプリケーションを実行する必要がある限り、64 ビット Solaris アプリケーション環境を使用する必要はありません。

一部のアプリケーション、特にデータベースなどの大規模アドレス空間を操作するアプリケーションを 64 ビット環境で実行すると、32 ビット環境では得られない利点があります。ただし当面は、32 ビット Solaris アプリケーション環境がデフォルトのアプリケーション環境になります。

64 ビット Solaris オペレーティング環境を使用する必要があるか

64 ビット Solaris オペレーティング環境は、32 ビットアプリケーションのソースレベルとバイナリレベルの互換性を提供します。通常のエンドユーザーと開発者は、Solaris 64 ビット環境を意識する必要はありません。

どの SPARC システムが 64 ビットアプリケーション開発をサポートしているか

64 ビットアプリケーションは、sun4c、sun4m、sun4d、および sun4u システム上の Solaris 7 環境で開発できます。もちろん、Solaris 7 環境の UltraSPARC システムでは、64 ビットアプリケーションを実行できます。

どのようにして 64 ビット Solaris オペレーティング環境をインストールするか

すべてのシステムタイプにおいて、Solaris 7 をインストールするときに、ソフトウェアを選択する際に 64 ビットサポートオプションを選択することができます。

- UltraSPARC システムでは、デフォルトで 64 ビットサポートが選択されます。このデフォルトを変更するには、64 ビットサポートの選択を解除します。
- 他のすべてのシステムでは、デフォルトで 32 ビットサポートが選択されます。このデフォルトを変更するには、64 ビットサポートを選択します。このオプションは、32 ビットシステム上で 64 ビットアプリケーションを開発する開発者に 64 ビット Solaris ライブラリを提供する「Developer (開発者)」および「Entire Distribution (全体ディストリビューション)」のソフトウェアクラスターで使用できます。

Solaris 7 をインストールするときに、いろいろな画面で 64 ビットサポートオプションを選択できます。

- Solaris 対話式インストールプログラムを使用して、初めて Solaris 7 をインストールするときは、64 ビットサポートオプションは、インストールするシステムソフトウェアのタイプ (「End User (エンドユーザー)」、「Developer (開発者)」、または「Entire Distribution (全体ディストリビューション)」) を選択する「Software Selection (ソフトウェアの選択)」画面で表示されます。
- Solaris 対話式インストールプログラムを使用してアップグレードを行うときには、64 ビットサポートオプションは、ソフトウェアのアップグレードを開始する前に現れる「Customize Software (ソフトウェアをカスタマイズしますか?)」画面で表示されます。
- Solaris Web Start を使用して、初めて Solaris 7 をインストールするときは、64 ビットサポートオプションは、「Custom Install (カスタムインストール)」ボタンを押した後に現れる画面でさらに「Configure Solaris (Solaris を設定する)」ボタンを押した後に現れる画面で表示されます。
- カスタム JumpStart™ を使用して、初めて Solaris 7 をインストールするときまたはアップグレードするときには、プロファイルを使用して、32 ビットサポートまたは 64 ビットサポートを選択できます。次のキーワードと値の組み合わせの 1 つをプロファイルに追加します。

```
isa_bits 32
```

```
isa_bits 64
```

システムが 64 ビット Solaris オペレーティング環境を実行できるかどうかを調べる方法

現時点では、64 ビット Solaris オペレーティング環境をサポートできるプラットフォームは、UltraSPARC システムだけです。使用しているシステムが UltraSPARC システムであるかどうかを確認するには、次のコマンドを使用します。

```
$ uname -m
sun4u
```

uname -m コマンドの出力が sun4u であれば、そのマシンは UltraSPARC システムです。

Solaris 7 を実行している場合は、psrinfo コマンドでも確認することができます。

```
# psrinfo -v
Status of processor 0 as of: 02/10/98 14:25:55
Processor has been on-line since 01/30/98 15:10:29.
The sparcv9 processor operates at 168 MHz,
and has a sparcv9 floating point processor.
Status of processor 1 as of: 02/10/98 14:25:55
Processor has been on-line since 01/30/98 15:10:33.
The sparcv9 processor operates at 168 MHz,
and has a sparcv9 floating point processor.
```

プロセッサタイプが sparcv9 であれば、そのプラットフォームは 64 ビット Solaris オペレーティング環境を実行できます。旧リリースの Solaris の psrinfo コマンドでは、sparc プロセッサタイプの情報が上記のように詳細に出力されません。

▼ システムが 64 ビット Solaris の機能を有効にしているかどうかを調べる方法

基本の 64 ビット Solaris オペレーティング環境は、以前の 32 ビット環境と同じように見えます。新しいコマンドを使用すれば、システムが 64 ビット機能を有効にしているかどうか、つまりシステムが 64 ビットカーネルでブートされているかどうかを調べることができます。

isainfo(1) コマンド (「instruction set architecture information」の意味) には次の 2 つの機能があります。

- 実行中のシステムがサポートするアプリケーションを報告する (isainfo -v コマンド)。

- 実行中のシステムでネイティブなアプリケーションがサポートするビット数を報告する (`isainfo -b` コマンド)。

例 — 64 ビット Solaris の機能がシステムで有効であるかどうかを調べる

32 ビットカーネルを実行している UltraSPARC システムでは、次のように出力されます。

```
$ isainfo -v
32-bit sparc applications
```

この出力は、システムが 32 ビットアプリケーションだけをサポートできていることを示しています。

64 ビットカーネルを実行している UltraSPARC システムでは、次のように出力されます。

```
$ isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

この出力は、システムが 32 ビットと 64 ビットの両方のアプリケーションをサポートできていることを示しています。

`isainfo -b` コマンドを使用すると、実行中のシステムでネイティブなアプリケーションがサポートするビット数が表示されます。

32 ビット Solaris オペレーティング環境を実行している SPARC、x86、または UltraSPARC システムでは、次のように出力されます。

```
$ isainfo -b
32
```

64 ビット Solaris オペレーティング環境を実行している 64 ビット UltraSPARC システムでは、次のように出力されます。

```
$ isainfo -b  
64
```

コマンドは 64 だけを返します。64 ビット UltraSPARC システムは両方のタイプのアプリケーションを実行できますが、64 ビットシステムで実行するのに最適なアプリケーションは、64 ビットアプリケーションです。

uname -p の出力は sparc または i386 のままです。これは、既存の 32 ビットアプリケーションの実行に影響を与えないためです。

どのようにして **64** ビット **Solaris** オペレーティング環境でアプリケーションを実行するか

64 ビット Solaris 環境でも、32 ビット Solaris 環境と同じようにアプリケーションを実行できます。

64 ビット Solaris システムを使用するために、ユーザーの PATH 環境変数を変更する必要はありません。ほとんどのシステムユーティリティは 32 ビットアプリケーションのままですが、いくつかのシステムユーティリティは 64 ビットバージョンに拡張されています。64 ビットバージョンのコマンドユーティリティは自動的にコマンドラッパープログラムによって呼び出され、実行中のシステムに応じたバージョンのコマンドが透過的に起動されます。

この他にも、64 ビット Solaris オペレーティング環境でアプリケーションを実行するときに注意が必要なことがあります。

- カーネルメモリーを読み取る 32 ビットアプリケーションは、64 ビットプログラムとしてコンパイルしなければなりません。これは、カーネルが 64 ビットプログラムになっているためです。
- /proc ファイルシステムで他のプロセスを参照する 32 ビットプログラムは、64 ビットプログラムに変換しなければなりません。これは、64 ビットプロセスの拡張機能を扱えるようにするためです。
- 大規模仮想アドレス空間を必要とする 64 ビットアプリケーションを実行する場合は、より多くのスワップ空間をシステムに追加しなければなりません。

64 ビットデバイスドライバやサン以外のデバイスドライバについて

システム管理者は、次の手順でデバイスドライバが 64 ビット Solaris システムで使用できるかどうかを確認しておく必要があります。

1. 64 ビット Solaris カーネルをブートするために、UltraSPARC システムのファームウェアをアップグレードする必要があるかどうかを確認します。UltraSPARC システムのファームウェアをアップグレードする必要があるかどうかを確認するには、ハードウェアメーカーのマニュアルを参照してください。
2. Solaris のインストール時に、64 ビット Solaris パッケージを選択します。
3. 64 ビット Solaris オペレーティング環境をブートします。

また、Sun 以外の他社製デバイスドライバが 64 ビットバージョンで使用できることを確認する必要があります。これらのデバイスドライバが 64 ビットで使用できない場合は、64 ビット Solaris オペレーティング環境は利用できません。

64 ビット Solaris オペレーティング環境で LD_LIBRARY_PATH 変数を使用する

64 ビットアプリケーションを実行する場合は、次のように LD_LIBRARY_PATH 環境変数をカスタマイズしなければならないことがあります。

- 新しい環境変数 LD_LIBRARY_PATH_64 を設定します。LD_LIBRARY_PATH に似ていますが、この環境変数は 64 ビットアプリケーションだけが参照できます。
- アプリケーションが 32 ビットと 64 ビットの両方のライブラリを参照できるように、LD_LIBRARY_PATH 変数を設定します。

32 ビットと 64 ビットの Solaris 7 カーネルを実行するシステムを同じ方法でブートするのか

同じようにブートされるわけではありませんが、2つのカーネルをそれぞれ違う方法でブートする必要はありません。64 ビット Solaris 7 カーネルが、UltraSPARC システムにインストールされている場合、64 ビットカーネルがデフォルトでブートされます。64 ビットカーネルがインストールされていなければ、32 ビットカーネルがデフォルトでブートされます。

また、UltraSPARC システムでは、kadb などの多くのブートプログラムが 64 ビットプログラムになっており、32 ビットカーネルでも 64 ビットカーネルでもブートできます。

32 ビットまたは 64 ビットのシステムをブートするときに問題が発生した場合は、次の節を参照してください。

64 ビット Solaris ブート時の問題の解決

64 ビット Solaris を UltraSPARC システムにインストールした後、次の条件に該当しないかぎり、64 ビットカーネルが自動的にブートされます。

- 64 ビットカーネルを正常にブートするためには、UltraSPARC システムの FLASH PROM をアップグレードする必要があります。UltraSPARC システムのファームウェアをアップグレードする必要があるかどうかを確認するには、ハードウェアメーカーのマニュアルを参照してください。
- OpenBoot PROM の `boot-file` パラメータが `kernel/unix` に設定されており、64 ビットカーネルのブートに失敗した場合は、設定を解除して、システムをリブートします。
- 一部の UltraSPARC システムでは、システム上に 64 ビット Solaris のすべてのコンポーネントおよび適切なファームウェアが完全にインストールされていても、64 ビット Solaris のカーネルが自動的にブートされないことがあります。64 ビットアプリケーションを実行するには、必ず 64 ビット Solaris のカーネルをブートする必要があります。

64 ビット Solaris のカーネルが自動的にブートされるようにするには、などこの問題についての詳細は、`boot(1M)` のマニュアルページおよび『Solaris 7 ご使用にあたって (SPARC 版)』を参照してください。

以下のように `isainfo -kv` コマンドによって、システム上で現在動作している Solaris カーネルを確認することができます。

```
$ isainfo -kv
64-bit sparcv9 kernel modules
```

この例では、64 ビット Solaris のカーネルが動作しています。

64 ビット Solaris オペレーティング環境は、32 ビット Solaris システムではブートできません。

64 ビット Solaris パッケージの変更

一部の Solaris 7 システムパッケージは、32 ビットと 64 ビットの両方のバージョンを提供しています。次に例を示します。

- SUNWcs1 パッケージには、32 ビット Solaris コア共有ライブラリが含まれています。
- SUNWcs1x パッケージには、64 ビット Solaris コア共有ライブラリが含まれています。

64 ビットアプリケーションを 32 ビットシステムで開発する場合は、64 ビットアプリケーションツール (64 ビット共有ライブラリ) を 32 ビットプラットフォームに追加できます。

32 ビットまたは 64 ビットのアプリケーションを自動的にマウントする

システム管理者は、`isainfo -n` コマンドの出力に対応している新しい変数 `NATISA` をオートマウンタのマップに使用できます。この変数を設定すると、システムが 32 ビットまたは 64 ビットのネイティブアプリケーションをサポートするかどうかによって、異なるディレクトリ階層をシステムにマウントできます。

たとえば、次のマップエントリを使用すると、64 ビットアプリケーションがそのシステムでサポートされているかどうかによって、クライアントは `/export/sparc/bin` または `/export/sparcv9/bin` のどちらかをマウントできます。

```
/export/bin server:/export/$NATISA/bin -ro
```

詳細は、`automount (1m)` のマニュアルページを参照してください。

索引

数字

- 1.2M バイト フロッピーディスク, 206, 207
- 1.44M バイトフロッピーディスク, 183, 206, 207
- 2.88M バイトフロッピーディスク, 183, 206
フォーマットオプション, 206
- 4.3 Tahoe ファイルシステム, 433
- 360K バイトフロッピーディスク, 183, 206, 207
- 720K バイトフロッピーディスク, 183, 206, 207
- 9660 CD フォーマット, 191, 260

A

Admintool

- ソフトウェア管理, 299
- 概要, 267, 268, 299
- サードパーティのパッケージ, 271

Admintool: ユーザー, 265

- アカウントを無効にする, 17
- 「削除」オプション (編集メニュー), 17
- 初期設定ファイル, 17
- 説明, 16
- 長所, 16
- 「追加」オプション (「編集」メニュー), 42, 43
- パスワード管理, 17
- 「変更」オプション (編集メニュー), 16

aliases ディレクトリ, 259

aliases ファイル, 17

ARCH 環境変数, 27

AutoClient システム, 269

autofs, 448

auto_home ファイル, 17

awk コマンド, 13

B

banner PROM コマンド, 144

bar コマンド

- 作成したファイルを取り出す, 674

basedir キーワード (管理ファイル), 272, 273, 283, 290

bin グループ, 6

bootparams エントリ, 74

Bourne シェル

- 環境変数, 26, 31
- 基本機能, 24, 25
- シェル (ローカル) 変数, 26, 29
- ユーザー初期設定ファイル, 22 - 24, 33, 39

Break キー, 154, 156

BSD Fat Fast ファイルシステム, 433

C

CALENDAR 環境変数, 27

CD, 183

- CD が使用中かどうかを調べる, 193

ISO 9660 フォーマット, 191, 260

UFS CD

- SPARC と x86 フォーマット, 191, 211
- 混合フォーマット, 260, 261

アクセス

- raw キャラクタのアクセス, 259

アクセスポイント, 187

- ボリューム管理の使用, 253, 255, 256

アクセスするプロセスを終了させる, 193

- 音楽 CD
 - 演奏用にシステムを設定, 199
- コマンド, 201
- 混合フォーマット, 260, 261
- 情報のコピー
 - コマンド行での入力, 192
- 取り出す
 - コマンド行での入力, 193
- 内容の検査
 - コマンド行での入力, 191
- 名前
 - CD の名前の確認, 194
 - 呼び名, 190
- ボリューム管理下のディレクトリ, 253, 255, 256
- マウント
 - 手作業と自動の比較, 186
 - リモート CD, 194, 195
- マウントポイント, 194, 256, 258
- リモートアクセス
 - 他のシステム上の CD にアクセスする, 194, 195
 - ローカルの CD を他のシステムで使
用できるようにする, 196
- ロード
 - コマンド行での入力, 191
- CD-ROM デバイス (命名), 332
- CD-ROM ドライブ, 183
 - アクセスポイント, 187
 - 新しいドライブ用の準備, 200
 - 共有, 196
 - ソフトウェアのインストール
 - マウントされた CD からサーバー
へ, 292
 - マウントされた CD からスタンドア
ロンシステムへ, 278
 - マウントされた CD からスプール
ディレクトリへ, 281
 - マウントされた CD からディスクレ
スクライアントへ, 289
 - ボリューム管理下のディレクトリ, 255,
256
- CD-ROM ドライブの共有, 196
- CDPATH 環境変数, 27
- /cdrom マウントポイント, 256, 258
- cfsadmin コマンド, 486, 496
- clri コマンド, 437
- cpio コマンド, 665, 652, 661

- 指定したファイルをテープから取り出
す, 664
- 使用する場合, 656
- ディレクトリ内のすべてのファイルを
テープにコピー, 661
- テープからすべてのファイルを取り出
す, 663
- テープ上のファイルのリスト, 662
- ファイルシステム間のディレクトリのコ
ピー, 652
- 別のヘッダー形式でファイルをコ
ピー, 672

cp コマンド

- CD 情報のコピー, 192
- フロッピーディスクからコピー, 219
- フロッピーディスクへコピー, 221

cred ファイル, 17

.cshrc ファイル

- カスタマイズ, 11, 24, 33
- 説明, 22

C シェル

- 環境変数, 26, 31
- 基本機能, 24, 25
- シェル (ローカル) 変数, 26, 29
- ユーザー初期設定ファイル, 22 - 24, 33, 39

D

- daemon グループ, 6
- dd コマンド, 648, 651
 - ディスクのクローン化, 649
 - ファイルをリモートテープから取り出す
(tar), 667
 - ファイルをリモートテープにコピー
(tar), 665
- DD (中密度)フロッピーディスク, 183
 - フォーマットオプション, 206, 207
- DESKSET 環境変数, 27
- /dev/dsk ディレクトリ, 327
- /dev/rdisk ディレクトリ, 327
- dfstab ファイル
 - 共有 CD の構成, 197
 - ユーザーのホームディレクトリ, 45
- df コマンド, 328, 437
- diskette ディレクトリ, 254
- dmesg コマンド, 322
 - SPARC の例, 322

x86 の例, 323
DOS
ファイルシステム, 434
フロッピーディスクのフォーマット, 213,
215
完全な形式, 206
サイズと密度のオプション, 206, 207
プラットフォームの互換性, 205
driver not attached メッセージ, 319

E

eject コマンド
CD, 193
フロッピーディスク, 222
env コマンド, 25
/etc/dfs/dfstab ファイル
共有 CD の構成, 197
ユーザーのホームディレクトリ, 45
/etc/dumpdates ファイル, 630
/etc/init.d ディレクトリ, 125
/etc/inittab ファイル, 119, 122
エントリの説明, 120
デフォルトの例, 120
/etc/passwd ファイル, 18, 154, 167
説明, 18
フィールド, 18, 19
ユーザー ID 番号の割り当て, 6
ユーザーアカウントの削除, 17
/etc/passwd ファイルの修復
SPARC, 154
x86, 167
/etc/rmmount.conf ファイル
CD の共有, 197
音楽 CD の演奏, 199
フロッピーディスクの共有, 226, 227
/etc/shadow ファイル
説明, 18
/etc/skel ディレクトリ, 22
/etc/vfstab ファイル, 46
/etc ファイル
ユーザーアカウント情報, 5, 18
/export/home ディレクトリ, 440
/export/home ファイルシステム, 10
/export/root ファイルシステム, 286
export コマンド, 26

F

fdformat コマンド, 232
DOS フロッピーディスクのフォーマット, 213, 215
UFS フロッピーディスクのフォーマット, 208
オプション, 232
確認メッセージ, 209
構文, 208, 209, 213, 215
サイズと密度のオプション, 206, 207, 213
正しいフォーマットの検査, 208
便利なオプション, 208, 213
FDFS ファイルシステム, 437
ff コマンド, 437
FIFOFS ファイルシステム, 437
FIFO i ノード, 539
/floppy マウントポイント, 256, 258
format.dat ファイル
エントリの作成, 370, 371
キーワード, 421, 424
構文規則, 420
内容, 419
format ユーティリティ
analyze メニュー, 416, 418
defect メニュー, 418
fdisk メニュー, 415
SCSI ディスクドライブの自動構成, 371,
374
Solaris fdisk パーティションの作成, 398,
402
概要, 342
関連するマニュアルページ, 427
機能と利点, 343
コマンド名の入力方法, 426
システム上のディスクの確認, 355, 356
使用条件, 411
使用上のガイドライン, 344
使用する場合, 344
情報の保存についての推奨事項, 412
ディスクスライス情報の表示, 360, 362
ディスクスライスとディスクラベルの作成
SPARC, 386
x86, 405, 407
ディスクのフォーマット, 358, 360
ディスクラベルの作成, 363, 365
入力, 425, 426
パーティションメニュー, 414, 415

破損したディスクラベルの復元, 367, 369
フォーマットされているかを調べる, 358
ブロック番号の指定方法, 425
ヘルプ機能の使用方法, 427
メインメニュー, 412
fsck pass フィールド (vfstab), 544
fsck コマンド, 328, 437
FSACTIVE 状態フラグ, 534
FSBAD 状態フラグ, 534
FSCLEAN 状態フラグ, 534
FSSTABLE 状態フラグ, 534
i ノードリストサイズのチェック, 538
空き i ノードのチェック, 538
空きブロックのチェック, 538
構文とオプション, 552, 555
修正する条件, 536
修復, 548
状態フラグ, 534
スーパーブロックのチェック, 537
対話式, 546
fsdb コマンド, 437
fstypes ファイル, 450
fstyp コマンド, 437
fs ファイル, 450
fuser コマンド, 245
CD が使用中かどうかを調べる, 193
CD にアクセスするプロセスを終了させる, 193
フロッピーディスクが使用中かどうかを調べる, 221

G

GECOS フィールド (passwd ファイル), 19
GID, 6
大きな値, 7
グループファイル, 21
定義, 12
割り当て, 12
grep コマンド, 13, 451
groupadd コマンド, 13
groupdel コマンド, 13
groupmod コマンド, 13
groups コマンド, 12
group* コマンド, 13
group ファイル
説明, 18

フィールド, 20
ユーザーアカウントの削除, 17

H

halt コマンド, 133
HD (High Density)フロッピーディスク, 183
HD (高密度) フロッピーディスク
フォーマットオプション, 206, 207
High Sierra ファイルシステム, 434
\$HOME/\$ENV ファイル, 22
HOME 環境変数, 27
home シェル変数, 25, 27
/home (自動マウント), 448
\$HOME ディレクトリ, 11, 24
/home の自動マウント, 448
/home ファイルシステム
ユーザーのホームディレクトリ, 10

I

ID 番号
グループ, 6, 12, 21
ユーザー, 5, 6, 16
init コマンド
スタンドアロンシステムのシャットダウン, 139
説明, 132
install4x, 90
installboot コマンド, 392, 409
ISO 9660 ファイルシステム, 434
iso_8895_1, 31
ISO 標準
9660 CD フォーマット, 191
aab9660 CD フォーマット, 260
i ノード, 567
FIFO, 539
キャラクタ型特殊ファイル, 539
形式とタイプのチェック, 539
サイズ, 541
シンボリックリンク, 539
通常, 539
ディレクトリ, 539
バイト数, 573
不良番号, 542
ブロック型特殊ファイル, 539
リンク数, 540

i ノードの形式, 539
i ノードの状態, 539
i ノードのタイプ, 539
i ノードのリンク数, 540
i ノードリストサイズ, 538

K

/kernel/drv ディレクトリ, 315
Korn シェル
環境変数, 26, 31
基本機能, 24, 25
シェル (ローカル) 変数, 26, 29
ユーザー初期設定ファイル, 22 - 24, 33, 39

L

L1-A キー, 154, 156
labelit コマンド, 437
LANG 環境変数, 27, 30, 31
LC 環境変数, 30, 31
"Do you want to continue with this
installation?" メッセージ, 291
"WARNING: " メッセージ, 288
LK パスワード, 17, 20
local.cshrc ファイル, 23
local.login ファイル, 23
local.profile ファイル, 23
locale 環境変数, 27
.login ファイル
カスタマイズ, 11, 24, 33
説明, 22
LOGNAME 環境変数, 27
loopback ファイルシステム
作成, 459
マウント, 469
lost+found ディレクトリ, 534
LPDEST 環境変数, 27
ls コマンド
CD の内容の検査, 191
フロッピーディスクの内容の検査, 218,
219

M

MAIL 環境変数, 26, 27
make コマンド, 13
MANPATH 環境変数, 27

MANSECT 環境変数, 27
media was found メッセージ, 217
mkfile コマンド, 528, 529
mkfs コマンド, 438, 454
mnttab ファイル, 446
mount コマンド, 328
mountall コマンド, 438
mount コマンド, 438
リモート CD, 195
リモートのフロッピーディスク, 224
mt コマンド, 680

N

NAMEFS ファイルシステム, 437
ncheck コマンド, 438
nec オプション, 207, 213
newfs コマンド, 211, 328, 454, 573
newgrp コマンド, 12, 21
NFS, 30, 447
nfsd デーモン
起動, 197, 226
実行中かの確認
CD を他のシステムで使用できるよう
にする, 196, 197
実行中かの検査
他のシステム上でフロッピーディス
クを使用できるようにす
る, 225, 226
NFS サーバー, 447
NFS ファイルシステム, 469
NIS
グループ, 13
ユーザーアカウント, 5, 13, 18
NIS+, 265
グループ, 12, 13
ユーザーアカウント, 5, 13, 18
nistbladm コマンド, 13
nis* コマンド, 13
noaccess グループ, 6, 21
noask_pkgadd 管理ファイル, 273, 280
nobody グループ, 6, 21
no media was found メッセージ, 217
NP パスワード, 20

O

OPENWINHOME 環境変数, 27
/opt ディレクトリ, 440
ソフトウェアパッケージのインストール, 270
ソフトウェアパッケージのインストール in, 285

OS サーバー

Solairs 7 OS サービスの追加, 94
SunOS 4.0 およびその互換バージョンの OS サービスの追加, 89
スタンドアロンシステムからの変換, 87
説明, 74

P

passwd コマンド, 13
passwd ファイル, 18
フィールド, 18, 19
ユーザー ID 番号の割り当て, 6
ユーザーアカウントの削除, 17
patchadd コマンド, 304, 307
patchrm コマンド, 304
pathchrn コマンド, 308
PATH 環境変数
設定, 29, 30
説明, 28, 29
path シェル変数, 25, 28
PC BIOS (とブート), 173
PCMCIA メモリーカード
DOS 用のフォーマット, 237
UFS ファイルシステム用のフォーマット, 232
識別, 230
情報のコピーまたは移動, 243
情報をカードへコピーまたは移動, 244
他のシステム上で使用できるようにする, 249
他のシステム上でのアクセス, 247
デフォルト名, 231
取り出す, 246
内容の表示, 242
マウント, 248
ロード, 240
PC ファイルシステム, 434
pkgadd コマンド

-a オプション (管理ファイル), 273, 277, 291
-d オプション (デバイス名), 277, 280, 291
-s オプション (スプールディレクトリ), 280
a オプション (管理ファイル), 279, 283
d オプション (デバイス名), 278, 279, 281, 282
R オプション (ルートファイルシステム), 283
s オプション (スプールディレクトリ), 282
概要, 266, 267, 276
サードパーティパッケージ, 271
サーバー, 289
使用の前提条件, 268
スタンドアロンシステム, 277, 280, 296
スプールディレクトリ, 280, 282
代替 ベースディレクトリ, 273, 283
ディスクレスクライアント, 286, 289
ユーザーの対話操作の省略, 273
pkgchk コマンド
オプション, 293, 295
概要, 276, 293
使用方法, 293, 295
pkginfo コマンド
-l オプション (詳細情報), 295
-R オプション (ディスクレスと AutoClient システム), 294
インストール済みのすべてのパッケージ, 293
概要, 268, 276, 293
使用, 280
使用方法, 293
pkginfo ファイル, 270
pkgmap ファイル, 271, 285
pkgparam コマンド, 276, 285
pkgrm コマンド, 296
-R オプション (ディスクレスと AutoClient システム), 297
-s オプション (スプールされたパッケージ), 297
rm コマンドとの比較, 272, 296
概要, 266, 267, 276
基本手順, 296
使用の前提条件, 268
注意, 272, 296

/pkg ディレクトリ, 281, 282
 PROCFS ファイルシステム, 436
 /proc ディレクトリ, 436, 440
 .profile ファイル
 カスタマイズ, 11, 24, 33
 説明, 22
 PROM
 ok プロンプトへの切り替え, 144
 ROM Rev 番号の確認, 144
 ブート設定の変更, 145
 モニター, 171
 リリースレベルの確認, 144
 prompt シェル変数, 28
 PROM のリリースレベル, 144
 prtconf コマンド, 320
 prtvtoc コマンド, 328, 365
 PS1 環境変数, 28

R
 raw ディスクデバイスインタフェース, 327, 328
 rdiskette ディレクトリ, 254
 reboot コマンド, 133
 removef コマンド, 272
 reset コマンド, 146
 rmmount.conf ファイル
 CD の共有, 197
 音楽 CD の演奏, 199
 フロッピーディスクの共有, 226, 227
 rm コマンド, 272, 296
 Rock Ridge 拡張 (HSFS ファイルシステム), 434
 root グループ, 6

S
 /sbin/rc0 スクリプト, 127
 /sbin/rc1 スクリプト, 128
 /sbin/rc2 スクリプト, 128
 /sbin/rc3 スクリプト, 129
 /sbin/rc5, 129
 /sbin/rc6 スクリプト, 129
 /sbin/rcS スクリプト, 129
 SCSI ディスクドライブ, 371
 SCSI テープドライブ, 677
 setenv コマンド, 26
 set コマンド, 26
 shadow ファイル
 説明, 18
 フィールド, 19, 20
 shareall コマンド, 448
 share コマンド, 448
 他のシステム上でディスクを使用できるようにする
 フロッピーディスク, 226
 ディスクを他のシステムで使用できるようにする
 CD, 197
 SHELL 環境変数, 28
 shutdown コマンド
 サーバーのシャットダウン, 111, 135
 説明, 132
 ユーザーへの通知, 133
 /skel ディレクトリ, 22
 Solaris fdisk パーティション, 395
 Solaris アップグレードオプション, 272
 Solaris のフロッピーディスクのフォーマット, 205
 Solaris フロッピーディスクのフォーマット, 207
 Solstice AdminSuite, 13
 Solstice AutoClient システム, 269
 sort コマンド, 13
 SPARC システム
 UFS フォーマット, 191, 260
 フロッピーディスクのフォーマット, 205, 207
 SPARC システムの再設定, 146
 SPECFS ファイルシステム, 437
 staff グループ, 12
 start コマンド, 201
 Stop-A キー, 154, 156
 stop コマンド, 201
 stty コマンド, 30
 SunOS のデフォルトファイルシステム, 439
 SUNW_PKGTYPE パラメータ, 269 - 271, 286
 SUNW接頭辞, 268
 Sun ソフトウェアパッケージ
 インストール, 270, 278, 280 - 283
 インストール位置の表示, 285
 Sun のソフトウェアパッケージ
 インストール, 270
 swapadd コマンド, 525
 SWAPFS ファイルシステム, 437

swap コマンド, 528
swmtool コマンド, 265, 267
sync コマンド, 155, 156
sync コマンドによるディスクの同期, 156, 157
sysdef コマンド, 320

T

tar コマンド, 656, 659
1 枚のフロッピーディスクにコピー, 669
テープからファイルを取り出す, 658
テープ上のファイルのリスト, 657
ファイルをテープへコピー, 656
ファイルをリモートテープから取り出す
(dd), 667
ファイルをリモートテープにコピー
(dd), 666
フロッピーディスクからファイルを取り
出す, 671
フロッピーディスク上のファイルのリス
ト, 670
TERMINFO 環境変数, 28
TERM 環境変数, 29
term シェル変数, 25, 29
TMPFS ファイルシステム, 435
/tmp ディレクトリ, 435, 440
tty (疑似), 6
tty タイプの疑似ユーザーログイン, 6
TZ 環境変数, 29

U

UFS CD
SPARC と x86 フォーマット, 191, 260
混合フォーマット, 260, 261
ufsdump コマンド, 601, 608
オプションと引数, 635
機能, 629
制限, 634
データのコピー方法, 630
テープへファイルをバックアップ, 602
媒体の終りの検出, 630
ufsrestore コマンド, 612, 627, 639
使用する準備, 612
テープからのファイルシステム全体の復
元, 622
UFS ファイルシステム, 433, 441
PCMCIA メモリーカード上に作成, 234

拡張基礎タイプ, 441
状態フラグ, 441
大規模ファイルシステム, 441
フロッピーディスクへ追加, 211, 213
マウント, 468

UFS フロッピーディスク, 183
SPARC と x86 フォーマット, 205, 260
UFS ファイルシステムの追加, 211, 213
フォーマット, 208, 211
SPARC と x86 フォーマット, 205
完全な形式, 206
サイズと密度のオプション, 206, 207
取り消し, 209

UID, 16
大きな値, 7
定義, 5
割り当て, 6

umask コマンド, 31
umountall コマンド, 438
umount コマンド, 438
UNIX グループ, 11, 265
UNIX ファイルシステム, 433
useradd コマンド, 13
userdel コマンド, 13
usermod コマンド, 13
user シェル変数, 25
/usr ファイルシステム, 439
ソフトウェアのインストール, 270, 285,
291, 292
ディスクレスクライアント, 269, 270, 287
データレスクライアント, 270
uucp グループ, 6

V

var/sadm/install/admin ディレクトリ, 273
/var/sadm/patch, 307
/var/spool/pkg ディレクトリ, 280 - 282
/var ディレクトリ, 440
vfstab ファイル, 450, 525
fsck pass の変更, 544
一時ファイルシステム用エントリ, 460
エントリの作成, 468
すべてのファイルのマウント, 469
スワップの追加, 525
デフォルト, 446
ファイルシステム名の検索, 600

vipw コマンド, 13
/vol/dev/dsk ディレクトリ, 255
/vol/dev/rdisk ディレクトリ, 255, 256
/vol/dev ディレクトリ, 253, 254
 cdrom マウントポイント, 256, 258
 CD サブディレクトリ, 253, 255, 256
 floppy マウントポイント, 256, 258
 シンボリックリンク
 raw デバイスのアクセス, 259
 ファイルシステムのアクセス, 258
/vol/dsk ディレクトリ, 256
volcopy コマンド, 438

W

who コマンド, 119, 134

X

x86 システム
 UFS フォーマット, 191, 260
 フロッピーディスクのフォーマット, 205,
 207

Y

yp* コマンド, 13

あ

アーカイブ (cpio), 673
空き i ノード, 538
空きブロック, 538, 568
空き容量 (最小), 571
アクセス, 183
 CD アクセスポイント, 187
 他のシステム上の PCMCIA メモリーカード, 247
 ディスクデバイス, 327, 331
 テープデバイス, 331
 フロッピーディスクのアクセスポイント, 187
アクセス権, 31, 39, 265
 CD からファイルをコピー, 192
アップグレードオプション (Solaris), 272
暗号化, 18

い

一次グループ, 12, 265
一時ファイルシステム, 435, 457
移動, 183
 フロッピーディスク情報
 フロッピーディスクから, 219, 220
 フロッピーディスクへ, 220, 221
インストールサーバー, 76

お

遅れ (回転), 571
オプション
 mkfile, 529
 ufsdump コマンド, 635
 ufsrestore コマンド, 639
音楽 CD, 199
音楽 CD の演奏, 199

か

カートリッジテープ
 たるみを直す, 680
回転の遅れ, 571
書き込み権
 説明, 192
書き込み保護 (PCMCIA メモリーカード), 232
拡張基礎タイプ (UFS ファイルシステム), 441
拡張密度フロッピーディスク, 183
 フォーマットオプション, 206
確認
 PROM リリースレベル, 144
 システム上のディスク, 355
 システムにログインしているユーザー, 134
 ファイルシステムのタイプ, 451
 nfsd デーモンが実行中かの
 CD を他のシステムで使用できるようにする, 197
カスタマイズ, 265
仮想ファイルシステム, 435
仮想ファイルシステムテーブル, 446
仮想メモリー記憶域, 440, 522
環境
 シェル, 25, 29
環境変数, 265
 説明, 25, 31

間接ブロック, 541
完全バックアップ
 ufsdump コマンドによる, 602
 定義, 587
 必要なテープの本数, 600
管理コマンド, 437
管理ファイル, 273, 277, 283, 290

き

記憶域 (仮想メモリー), 440, 522
記憶ブロック, 568
記憶容量 (媒体), 581, 676
疑似 tty, 6
疑似ユーザーのログイン, 6
疑似ユーザーログイン, 6
起動, 183
 nfsd デーモン, 197, 226
 ボリューム管理, 201
キャッシュされたファイルシステム, 484
 削除, 496
 作成, 486
 情報の表示, 494
 設定, 486
 チェック (fsck), 497
 パラメータ, 519
 パラメータの設定, 488
キャラクタ型特殊 i ノード, 539
共有
 クライアントとサーバーによるソフトウェアの共有, 268, 298
 ファイル, 447
 ユーザーのホームディレクトリ, 44, 46
記録
 増分バックアップ, 631
 ダンプ, 630

く

空白 (ユーザーログイン名における), 5
クライアント
 サポート, 104
 ソフトウェア管理, 269, 271

ソフトウェア管理
 Solstice AutoClient システム, 269
 Sun のパッケージ, 270, 283
 異機種環境, 271, 285, 286
 インストール済みソフトウェア情報の表示, 293, 295
 サードパーティパッケージ, 271, 284
 サーバーとのソフトウェアの共有, 268, 298
 同機種環境, 282
 パッケージの削除, 297

グループ

 ID 番号, 6, 12, 21
 UNIX, 11
 アクセス権の設定, 31
 一次, 12
 一次グループの変更, 12
 管理ツール, 13
 管理のガイドライン, 11, 12
 コマンド, 13
 情報の格納, 18, 20
 説明, 4, 11
 デフォルト, 12
 名前の変更, 49
 二次, 12, 21
 ネームサービス, 12, 13
 パスワード, 21
 ユーザーが所属するグループの表示, 12
 削除
 コマンド, 13
 追加
 Admintool: グループ, 41
 コマンド, 13
 名前
 説明, 12
 変更, 49
 ユーザーの変更
 Admintool: ユーザー, 16
 コマンド, 13
グループ ID 番号, 6, 12, 21
グループファイル
 フィールド, 21
グループ名
 説明, 12
 変更, 49

け

検査

- CD が使用中かどうか, 193
- フロッピーディスクが使用中か, 221
- nfsd デーモンが実行中かについて
CD を他のシステムで使用できるようにする, 196
- nfsd デーモンの実行
他のシステム上でフロッピーディスクを使用できるようにする, 225, 226
- ソフトウェアパッケージのインストール
 - pkgchk コマンド, 276, 293, 295
 - pkginfo コマンド, 276, 280, 296

検索

- ユーザーアカウント, 13

検出、媒体の終わり

- ufsdump コマンド, 634

検出、媒体の終わりの

- cpio コマンド, 652
- ufsdump コマンド, 630

こ

構造、シリンダグループの, 565

構文

- fsck コマンド, 552, 554
- newfs, 573

高密度 (HD) フロッピーディスク, 183

- フォーマットオプション, 206, 207

互換性のあるアーカイブ, 673

コピー, 183

- PCMCIA メモリーカードから情報をコピー, 243
- PCMCIA メモリーカードへ情報をコピー, 244
- 個別のファイル (cpio), 652
- ディレクトリ内のすべてのファイルをテープにコピー (cpio), 661
- データ (ufsdump), 630
- テープからファイルをコピー (pax), 660
- ファイルシステム間でディレクトリをコピー (cpio), 652
- ファイルシステム間のディレクトリのコピー (cpio), 652
- ファイルシステム全体 (dd), 648
- ファイルのグループ (cpio), 652

ファイルをテープへコピー (tar), 656

ファイルをフロッピーディスクにコピー, 668

ファイルをリモートテープにコピー (tar と dd), 665

フロッピーディスクにファイルをコピー (tar), 669

別のヘッダー形式でファイルをコピー (cpio), 672

CD 情報

コマンド行での入力, 192

フロッピーディスク情報

フロッピーディスクから, 219, 220

フロッピーディスクへ, 220, 221

ユーザーアカウント

コマンド, 13

さ

サードパーティソフトウェア パッケージ

インストール位置の表示, 285

サードパーティのソフトウェアパッケージ

サーバーへのインストール, 271

サードパーティのソフトウェアパッケージ

クライアントへのインストール, 271, 284

サーバー, 265

OS サーバー, 74

インストールサーバー, 76

サポート, 104

説明, 68

ブートサーバー, 76

プロファイルサーバー, 76

サポート

管理, 104

ソフトウェア管理

Sun のパッケージ, 270

異機種環境, 271, 285, 289

クライアントとのソフトウェアの共有, 268, 298

サードパーティパッケージ, 271

同機種環境, 282

パッケージの削除, 296

パッケージの追加, 289, 292

サーバーとクライアントサポートの管理, 103, 104

サーバーとクライアントのサポート

説明, 66

- デフォルトの設定, 84
- サービスの起動と停止, 124
- 再構成ブート, 372
 - SPARC の例, 384
 - x86 の例, 397
- 最小空き容量, 571
- 最小の長さ
 - ユーザーパスワードの長さ, 9
- 最小文字数
 - ユーザーログイン名の長さ, 5
- サイズ
 - i ノード, 541
 - ファイルシステムのチェック, 538
 - フラグメント, 570
 - フロッピーディスクのフォーマット, 206, 207
 - フロッピーディスクのフォーマット
 - DOS フロッピーディスク, 213
 - UFS フロッピーディスク, 208, 209
- 最大数
 - ユーザーが所属できる二次グループ数, 12
- 最大値
 - ユーザー ID 番号, 6
- 最大の長さ
 - ユーザーパスワードの長さ, 9
- 最長文字数
 - ユーザーログイン名の長さ, 5
- 最適化のタイプ, 572
- サイトの初期設定ファイル, 23
- 削除
 - キャッシュされたファイルシステム, 496
 - メール別名, 17
 - ユーザーのホームディレクトリ, 17
 - ユーザーのメールボックス, 17
 - グループ
 - コマンド, 13
 - ユーザーアカウント
 - Admintool: ユーザー, 17
 - コマンド, 13
 - 「削除」オプション (編集メニュー)
 - Admintool: ユーザー, 17
- 作成, 265
 - Solaris fdisk パーティション, 395, 398
 - UFS ファイルシステム, 455
 - format.dat エントリ, 370, 371
 - loopback ファイルシステム, 459
 - 一時ファイルシステム, 457
 - 互換性のあるアーカイブ (cpio), 673

- スワップファイル, 528
- ファイルシステム, 454
- マウントポイント, 444
- ディスクスライスとディスクラベル
 - SPARC, 385
 - x86, 405
- サポートされていないデバイス, 315

し

シェル

- 環境, 25, 29
- 環境変数, 25, 26, 31
- 基本機能, 24, 25
- ユーザー初期設定ファイル, 22 - 24, 33, 39
- ローカル変数, 25, 26, 29
- シェル変数, 26, 29
- 時間 (最適化のタイプ), 572
- 時間帯の環境変数, 29
- 磁気テープカートリッジ
 - たるみを直す, 680
 - 巻き戻し, 680
- 磁気テープカートリッジのたるみを直す, 680
- 磁気テープカートリッジを巻き戻す, 680
- 識別
 - PCMCIA メモリーカード, 230
 - デバイス, 320
- システムアーキテクチャ, 27
- システムアカウント, 6
- システムシャットダウンコマンド, 132
- システム初期設定ファイル, 11
- システムタイプ
 - generic, 73
 - 概要, 67
 - 更新, 83
 - サーバー, 68
 - スタンドアロンシステム, 69
 - 選択のガイドライン, 70
 - 変換, 74
- システムタイプの更新, 83
- システムタイプの変換, 87
- システムディスク
 - 説明, 341
 - 接続
 - SPARC, 383
 - x86, 395

- ブートブロックのインストール
 - SPARC, 391
 - x86, 409
- システムのシャットダウン, 111
- システムのブート (ガイドライン), 112
- 実行制御スクリプト, 123
 - サービスの起動と停止, 124
 - 追加, 125
 - 無効にする, 126
- 実行レベル
 - 0 (電源切断状態), 118
 - 1 (シングルユーザー状態), 118
 - 2 (マルチユーザー状態), 118
 - 3 (NFS が使用できるマルチユーザー状態), 118
 - 6 (リポート状態), 118
 - s または r S (シングルユーザー状態), 118
 - 調べる, 119
 - 定義, 117
 - デフォルトの実行レベル, 117
 - 3 (NFS を使用できるマルチユーザー)
 - 実行されるプロセス, 122
 - 実行レベル 3 のシステムの動作, 121
 - ブート, 148, 161
 - s または S (シングルユーザー状態)
 - ブート, 162
 - s または S (シングルユーザー状態)
 - ブート, 149
- 自動構成プロセス, 314
- 自動ブートチェック機能の変更, 544
- 自動ブートチェックの変更, 545
- 自動マウント
 - ユーザーのホームディレクトリ, 11
- シャットダウン
 - shutdown と init コマンドによる正常なシャットダウン, 132
 - サーバー, 134
 - スタンドアロンシステム, 138
- シャットダウン時間のユーザーへの通知, 134
- 重複ブロック, 541
- 終了, 183
 - CD にアクセスするプロセス, 193
 - フロッピーディスクにアクセスするプロセス, 221
 - ボリューム管理, 201
- 準備
 - バックアップ, 599
 - ファイルの復元, 611

- 状態フラグ
 - UFS ファイルシステム, 441
 - fsck, 534
- 初期設定ファイル
 - システム, 11
- 調べる
 - 完全バックアップに必要なテープの本数, 600
 - ディスクデバイス名, 612
 - テープデバイス名, 612
 - テープドライブのタイプ, 612, 678
 - ファイルシステムのタイプ, 451
- シリンドラグループ, 565
- シンボリックリンク, 539
 - CD ディレクトリの表示, 191
 - raw デバイスのアクセス, 259
 - ファイルシステムのアクセス, 258
 - フロッピーディスクのディレクトリの表示, 218

す

- スーパーブロック, 537, 549, 566
- スケルトンディレクトリ, 11, 17
- スタンドアロンシステム, 69
 - OS サーバーへの変換, 87
 - ソフトウェアパッケージの削除, 296
 - ソフトウェアパッケージの追加, 277, 280
- スプールディレクトリ
 - ソフトウェアパッケージのインストール, 280, 282, 295
 - ソフトウェアパッケージの削除, 297
- すべてのデバイスの電源を落とす方法, 140
- すべてのプロセスの停止, 479
- スライス (定義), 337
- スワップパーティション, 440, 522
- スワップファイル
 - vfstab へ追加, 525
 - 削除, 530
 - 作成, 528
 - 表示, 527
- スワップファイルの削除, 530

せ

- 正常なシャットダウン, 132
- セキュリティ, 265

ユーザー ID 番号の再利用, 6

そ

増分バックアップ, 587, 631

その他 (アクセス権の設定), 31

ソフトウェア管理, 265

Solaris アップグレードオプション, 272
インストール済みソフトウェア情報の表示, 268, 270, 271, 276, 285, 293, 295

概要, 265, 271

クライアント, 269, 271

ツール, 265 - 267

定義, 266

パッケージの削除, 266, 269, 272, 296

パッケージの追加, 265, 266, 271, 273, 276

パッケージの定義, 266

パッケージの命名規則, 268

インストールの検査

pkgchk コマンド, 276, 293, 295

pkginfo コマンド, 276, 280, 295

クライアント

Solstice AutoClient システム, 269

Sun のパッケージ, 270, 283

異機種環境, 271, 285

サードパーティパッケージ, 271, 284

サーバーとのソフトウェアの共有, 268, 298

同機種環境, 282

サーバー

Sun のパッケージ, 270

異機種環境, 271, 285, 290

クライアントとのソフトウェアの共有, 268, 298

サードパーティパッケージ, 271

同機種環境, 282

パッケージの削除, 296

パッケージの追加, 289

ツール

コマンド, 266, 267, 276

パッケージの削除

ガイドライン, 272

管理ファイル, 273

スタンドアロンシステム, 296

スプールディレクトリ, 297

ツール, 266, 267

ディスクレスクライアント, 297, 298

パッケージの追加

Sun のパッケージ, 270, 283

Sun パッケージ, 278, 280 - 282

異機種環境, 271, 285

ガイドライン, 269, 271

管理ファイル, 273, 277

サードパーティパッケージ, 271, 284

サーバー, 289

スタンドアロンシステム, 277, 280

スプールディレクトリへ, 280, 282, 295

前提条件, 268

ツール, 266, 267

ディスクレスクライアント, 269 - 271, 282, 286, 289

同機種環境, 282

パッケージの複数のバージョン, 272, 273

ベースディレクトリ, 272, 273

マウントされた CD から, 278

ユーザーの対話操作の省略, 273

リモートパッケージサーバーから, 279, 280

ソフトウェア管理、パッケージの削除, 265

ソフトウェアパッケージの複数のバージョン, 272, 273

ソフトウェアマネージャ, 265, 267

た

大規模ファイルシステムオプション, 464

対話式

ファイルシステムのチェック, 546, 547

復元, 615

対話式ブート

SPARC, 150

x86, 163

ダンプレベル

定義, 588

日単位個別バックアップ, 589

日単位累積バックアップ, 588

ち

チェック

i ノードの形式とタイプ, 539

i ノードリストの整合性, 538

空き i ノード, 538

- 空きブロック, 538
- 自動ブートの変更, 544
- ファイルシステム, 546
- ファイルシステムのサイズ, 538
- ファイルシステムの対話式のチェック, 547
- 中密度 (DD) フロッピーディスク, 183
 - フォーマットオプション, 206, 207
- 直接アクセスするディスクコントローラ, 329

- つ
- 追加
 - Solaris 7 OS サービスを OS サーバーに, 94
 - SunOS 4.0 およびその互換バージョンの OS サービスを OS サーバーに追加, 89
 - 実行制御スクリプト, 125
 - 周辺デバイス, 316
 - スワップを `vfstab` へ追加, 525
 - デバイスドライバ, 317
 - ユーザー初期設定ファイル, 17
 - グループ
 - Admintool: グループ, 41
 - コマンド, 13
 - サーバーとクライアントサポート
 - OS サーバー, 84
 - サーバーとクライアントのサポート
 - スタンドアロンシステム, 84
 - 説明, 66
 - ディスクレスクライアント, 100
 - デフォルトの設定, 84
 - ホストマネージャ, 71
 - ディスク
 - x86, 394
 - ユーザーアカウント
 - Admintool: ユーザー, 42, 43
 - コマンド, 13
 - ユーザーをグループに
 - Admintool: グループ, 49
 - Admintool: ユーザー, 16
 - コマンド, 13
 - 「追加」オプション (「編集」メニュー)
 - Admintool: グループ, 41
 - Admintool: ユーザー, 42, 43
- 通常の i ノード, 539

- て
- 提供側スライスの説明, 351
- 停止
 - CD にアクセスするプロセスを終了させる, 193
 - ファイルシステムのすべてのプロセス, 479
 - フロッピーディスクにアクセスするプロセスの終了, 221
 - CD の取り出し
 - コマンド行での入力, 194
- ディスク
 - SCSI ドライブの自動構成, 371
 - 欠陥セクターの修復, 374, 377
 - システム上のディスクの確認, 355
 - ディスクラベルの検査, 365
 - 破損したディスクラベルの復元, 366
 - フォーマット, 346
 - フォーマットされているかを調べる, 358
 - フォーマットする場合, 357
 - ラベル, 363
 - 新しいディスク上でのファイルシステムの作成
 - SPARC, 390
 - x86, 408
 - システムディスクの接続
 - SPARC, 383
 - x86, 395
 - 追加
 - x86, 394
 - ディスクスライスとディスクラベルの作成
 - SPARC, 386
 - x86, 405
 - 二次ディスクの接続
 - SPARC, 384
 - x86, 396
- ディスクコントローラ, 328
- ディスクスライス
 - システム構成の要件, 342
 - 使用するスライスの決定, 341
 - 情報の表示, 360
 - 定義, 337
- ディスクスライスの指定, 328, 331
- ディスクデバイス名, 612
- ディスクのクローン化, 649
- ディスクのフォーマットの概要, 346

- ディスクベースのファイルシステム, 433
- ディスクラベル
 - prvtoc コマンドによる検査, 365
 - 作成, 363
 - 説明, 347
- ディスクレスクライアント
 - ソフトウェア管理
 - インストール済みソフトウェア情報の表示, 293
 - サポートの追加, 100
 - パッケージの削除, 297, 298
 - パッケージの追加, 269 - 271, 282, 286
 - ソフトウェア管理 for
 - パッケージの追加, 289
- 低密度フロッピーディスク, 183
 - フォーマットオプション, 206, 207
- ディレクトリ, 183
 - PATH 環境変数, 28 - 30
 - i ノード, 539
 - path シェル変数, 25, 28
 - /proc, 436
 - /tmp, 435
 - アクセス制御, 31
 - アクセスの制御, 4
 - スケルトン, 11, 17
 - ファイルシステム間のコピー (cpio), 652
 - ベースディレクトリ (basedir), 272, 273, 283, 290
 - ホーム, 10
 - 未割り当てブロック, 542
 - ユーザーアカウント用の所有権の変更, 16
- CD
 - /cdrom マウントポイント, 256, 258
 - ボリューム管理による管理, 253, 255, 256
- コピー
 - CD から, 192
 - フロッピーディスクから, 219
 - フロッピーディスクへ, 220, 221
 - フロッピーディスク
 - /floppy マウントポイント, 256, 258
 - ボリューム管理による管理, 253
- データディレクトリブロック, 542
- データブロック, 543
- テープ, 680
 - tar コマンドによるファイルのリスト, 657
 - 記憶容量, 581, 676
 - サイズ, 581, 676
 - 指定したファイルを取り出す (cpio), 664
 - ディレクトリ内のすべてのファイルをテープにコピー (cpio), 661
 - 特性, 634
 - ファイルを取り出す (cpio), 663
 - ファイルを取り出す (tar), 658
 - 容量, 634
 - テープ (磁気カートリッジ)
 - たるみを直す, 680
 - テープデバイス (命名), 331
 - テープドライブ
 - 管理, 680
 - 最大の SCSI テープドライブ数, 677
 - タイプを調べる, 678
 - デフォルトの密度, 677
 - 復元用にタイプを調べる, 612
 - リモートからの復元, 620
 - テープドライブの管理, 680
 - テープドライブのタイプ, 678
 - テープの容量, 634
 - テーブル
 - マウント, 446
 - 手作業によるマウント, 183
 - 自動マウントとの比較, 186
 - リモート CD, 194, 195
 - リモートのフロッピーディスク, 224
 - 手作業のマウント
 - リモートのフロッピーディスク, 223
 - デバイス
 - アクセス, 325
 - 電源を落とす場合, 133
 - デバイスドライバ
 - 追加, 318
 - 定義, 313
 - デバイスのインスタンス名, 326
 - デバイスの電源を落とす場合, 133
 - デバイス名
 - ディスクデバイス名を調べる, 612
 - テープデバイス名を調べる, 612
 - バックアップ, 676
 - デフォルト
 - LOGNAME 環境変数, 27
 - PATH 環境変数, 28, 30
 - SHELL 環境変数, 28
 - SunOS ファイルシステム, 439
 - TZ (時間帯) 環境変数, 29
 - /tmp (TMPFS) 用ファイルシステム, 435

- グループ, 12
 - スプールディレクトリ, 280
 - テープドライブの密度, 677
 - ファイルのアクセス権, 31
 - プリンタ, 27
 - マウントオプション, 472, 475
 - ユーザー初期設定 ファイル, 22
 - ユーザーアカウントのデフォルト設定
 コマンド, 13
- と
- 取り消し, 183
 - UFS フロッピーディスクのフォーマット, 209
 - 取り出し
 - bar コマンドで作成したファイルを取り出す, 674
 - 指定したファイルをテープから取り出す (cpio), 664
 - テープからすべてのファイルを取り出す (cpio), 663
 - テープからファイルを取り出す (tar), 658
 - ファイルをフロッピーディスクから取り出す (tar), 671
 - ファイルをリモートテープから取り出す (tar と dd), 667
 - 取り出す
 - CD
 - コマンド行での入力, 193
 - PCMCIA メモリーカード
 - コマンド行での入力, 246
 - フロッピーディスク
 - コマンド行での入力, 222
- な
- 名前
 - SUNW 接頭辞, 268
 - ソフトウェアパッケージの命名規則, 268
 - グループ
 - 説明, 12
 - ユーザーログイン
 - 説明, 4, 5
 - 変更, 16
 - 名前/命名
 - フロッピーディスク名, 204
- CD 名
 - CD の名前の確認, 194
 - 呼び名, 190
- に
- 二次グループ, 12, 21, 265
 - 二次ディスク
 - 説明, 341
 - システムへの接続
 - SPARC, 385
 - x86, 397
- ね
- ネームサービス, 265
 - グループ, 12, 13
 - ユーザーアカウント, 5, 13, 18
 - ネットワーク, 265
 - ネットワークベースのファイルシステム, 434
- は
- パーティション (スワップ), 440, 522
 - 媒体の終わりの検出
 - cpio コマンド, 652
 - 媒体の終りの検出
 - ufsdump コマンド, 630, 634
 - バイト数 (i ノードごとの), 573
 - バス指向コントローラ, 330
 - バス指向ディスクコントローラ, 330
 - バス名, 187
 - パスワード (グループ), 21
 - パスワードの有効期限, 17, 18
 - パスワードファイル
 - テープからの復元, 620
 - パスワード (ユーザー)
 - Admintool: ユーザー, 17
 - *LK* パスワード, 17, 20
 - NP パスワード, 20
 - 暗号化, 18
 - 期限切れ, 20
 - 設定, 9, 17
 - 説明, 4, 9, 10
 - 選択, 9
 - 注意点, 9
 - 有効期限, 10, 17, 18

ユーザーアカウントを無効に、または
ロックする, 17, 20

変更

- Admintool: ユーザー, 17
- コマンド, 13
- 頻度, 9, 20
- ユーザーによる, 9
- ユーザーによる変更, 9

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, 586
- 準備, 599
- 増分バックアップの記録, 631
- タイプ, 586
- テープへの完全 (レベル 0) バックアッ
プ, 602
- デバイス名, 676
- ファイルシステムの選択, 582
- 理由, 581

バックアップスケジュール, 587

- ガイドライン, 587
- サーバー用, 593
- 推奨事項, 597
- ダンプレベル, 588
- 日単位増分、週単位累積バックアッ
プ, 592
- 日単位累積、週単位増分バックアッ
プ, 591
- 日単位累積、週単位累積バックアッ
プ, 590

例, 590, 597

パッチ

- Sun Service カスタマが利用できるも
の, 305
- WWW によるアクセス, 306
- ftp によるアクセス, 306
- 一般的な利用可能性, 305
- インストール, 307
- インストール済みのものを調べる, 304
- インストール用 README, 304
- 削除, 308
- 参照先, 306
- 定義, 303
- パッチ番号, 307
- ユーティリティ, 304

パラメータ (ファイルシステム), 569

ひ

日単位個別バックアップ, 589

日単位累積バックアップ, 588

ビッグエンディアン方式のビットコード
化, 203

表

mkfile のオプション, 529

ufsdump コマンドのオプション, 636

ufsrestore コマンドのオプション, 639

ufsrestore の対話式コマンド, 642

テープ容量, 634

デフォルトの Solaris ファイルシステ
ム, 439

媒体の記憶容量, 581, 676

ファイルシステム管理コマンド, 437

表記上の規則, xxxii

表示, 183

- CD ユーザー, 193
- PCMCIA 内容, 242
- 環境変数, 25
- システム設定情報, 319, 321
- スワップ空間, 527
- ディスクスライス情報, 360
- デバイス情報, 322
- ユーザーマスク, 31

CD の内容

- コマンド行での入力, 191

フロッピーディスクの内容

- コマンド行での入力, 218, 219

ふ

ファイル, 265

- /etc/default/fs, 450
- /etc/dfs/fstypes, 451
- /proc ディレクトリの中, 436
- アクセス制御, 31
- アクセスの制御, 4
- 数, 573
- 共有, 447
- コピー (cpio), 652
- 新規インストールパッケージの属性の検
査, 295
- 対話式の復元, 615
- テープから取り出す (cpio), 663, 664
- テープから取り出す (tar), 658

- テープからファイルをコピー (pax), 660
- テープ上のファイルのリスト (tar), 657
- テープ上のファイルのリストを表示 (cpio), 662
- テープへコピー (tar), 656
- 特定のファイルの復元, 618
- 複数のフロッピーディスクにアーカイブする (cpio), 672
- フロッピーディスクから取り出す (tar), 671
- フロッピーディスク上のファイルのリスト (tar), 670
- フロッピーディスクにコピー, 668
- フロッピーディスクにコピー (tar), 669
- ユーザーアカウント用の所有権の変更, 16
- ファイルシステム
 - /, 439
 - 4.3 Tahoe, 433
 - BSD Fat Fast, 433
 - DOS, 434
 - FDFS, 437
 - FIFOFS, 437
 - High Sierra, 434
 - ISO 9660, 434
 - NAMEFS, 437
 - NFS のマウント, 474
 - PC, 434
 - PROCFS, 436
 - SPECFS, 437
 - SWAPFS, 437
 - SunOS のデフォルト, 439
 - TMPFS, 435
 - UFS, 433
 - UFS について, 441
 - UFS の作成, 455
 - UNIX, 433
 - /export/home, 440
 - loopback の作成, 459
 - /opt, 440
 - /proc, 440
 - /usr, 439
 - /var, 440
 - カスタムパラメータ, 569
 - 仮想, 435
 - 管理コマンド, 437
 - 管理コマンドの表, 437
 - キャッシュされた, 484
 - 共有, 447
 - サイズのチェック, 538
 - 修正, 551
 - 修復, 548
 - 使用可能にする, 462
 - シリンダグループの構造, 565
 - すべてのプロセスの停止, 479
 - 整合性のなくなる原因, 537
 - 全体のコピー (dd), 648
 - 全体の復元, 613, 621
 - 大規模, 464
 - タイプ, 433
 - タイプを調べる, 451
 - 対話方式のチェック, 546, 547
 - チェック, 546
 - 定義, 433
 - ディスクベース, 433
 - テーブル (vfstab), 446
 - どれにバックアップするか, 582
 - ネットワークベース, 434
 - 破損, 534
 - バックアップする理由, 581
 - プロセス, 436
 - マウント解除, 480
 - マウントテーブル, 446
 - マニュアルページ, 438
 - ファイルシステムテーブル
 - 仮想, 446
 - ファイルシステムの修復, 548
 - ファイルシステムのすべてのプロセスの停止, 479
 - ファイルシステムの整合性がなくなる, 537
 - ファイルシステムのタイプ, 433
 - ファイルシステムの破損, 534
 - ファイルシステムのバックアップと復元
 - コマンド, 580
 - コマンドの概要, 585
 - 定義, 580
 - ファイルシステムの復元
 - 全体, 621, 623
 - ルートと /usr, 624, 627
 - ファイルシステムのマウント解除, 443, 480, 481
 - ファイルシステム破損の原因, 534
 - ファイルシステム用のカスタムパラメータ, 569
 - ファイルとディレクトリアクセスの制御, 4

ファイルとディレクトリのアクセス制御, 31
ファイルとファイルシステム, 183
 UFS ファイルシステム, 211, 213
 ファイルのアクセス
 CD ファイル, 187
 フロッピーディスクのファイル, 187
ファイルマネージャ
 フロッピーディスクのフォーマット, 208, 213
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), 672
フィールド
 passwd ファイル, 19
 shadow ファイル, 20
 グループファイル, 21
フィールド (passwd ファイル), 19
フィールド (shadow ファイル), 20
フィールド (グループファイル), 21
フィールド交換ユニット (FRU), 70
ブート
 PC BIOS, 173
 自動チェックの変更, 545
 自動ブートチェック機能の変更, 544
 カーネルデバッグ (kadb) の使用
 SPARC, 156
 クラッシュダンプの強制とリポート
 SPARC, 155
 x86, 170
 実行レベル 3
 x86, 161
 実行レベル 3 に
 SPARC, 148
 実行レベル S
 SPARC, 149
 x86, 162
 対話式
 SPARC, 150
 ブートプロセス
 SPARC, 172
 x86, 180
 復元のため
 SPARC, 152
 x86, 165
 ブート PROM の設定, 145
 ブートサーバー, 76
 ブートタイプの説明, 111
 ブートブロック, 566
 ブートブロックのインストール

 SPARC, 391
 x86, 409
フォーマット
 フロッピーディスク, 215
 PCMCIA メモリーカード
 DOS 用, 237
 UFS ファイルシステム, 232
 フロッピーディスク
 DOS フロッピーディスク, 213, 215
 UFS フロッピーディスク, 208, 210
 完全な形式, 206
 サイズと密度のオプション, 206 - 209, 213
 注意, 208, 213
 ハードウェアの考慮点, 205, 207
 ファイルマネージャの問題点, 208, 213
 フォーマットの取り消し, 209
 フロッピーディスク名, 204
復元
 準備, 611
 対話式, 615
 テープドライブのタイプ, 612
 特定のファイル, 613, 618
 ファイルシステム全体, 621
 リモートドライブからの復元, 620
不正な . と .. エントリ, 543
物理デバイス名, 326
フラグメントサイズ, 570
不良 i ノード番号, 542
不良スーパーブロック, 549
不良スーパーブロックの復元, 549
不良ファイルシステムの修正, 551
不良ブロック数, 541
プリンタ (デフォルト), 27
プロセスファイルシステム, 436
ブロック
 空き, 568
 間接, 541
 記憶, 568
 重複, 541
 通常ファイル, 543
 ディレクトリデータ, 542
 特殊 i ノード, 539
 ブート, 566
 論理サイズ, 570
 ブロックサイズ, 454

- ブロック数
 - 不良, 541
- ブロックディスクデバイスインタフェース
 - 使用する場合, 328
 - 定義, 327
- フロッピーディスク, 183
 - コマンド, 203
 - 名前, 204
 - ファイルのコピー, 668
 - ファイルのリスト表示 (tar), 670
 - ファイルをコピー (tar), 669
 - ファイルを取り出す (tar), 671
 - 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), 672
 - フォーマット, 215
 - フロッピーディスクが使用中かどうかを調べる, 221
 - フロッピーディスクにアクセスするプロセスの終了, 221
 - ボリューム管理下のディレクトリ, 254
 - マウントポイント, 256, 258
 - UFS フロッピーディスク
 - SPARC と x86 フォーマット, 205, 260
 - UFS ファイルシステムの追加, 211, 213
 - フォーマット, 208, 210
 - フォーマットの取り消し, 209
 - アクセス
 - raw キャラクタのアクセス, 259
 - アクセスポイント, 187
 - ボリューム管理の使用, 253
 - 情報の移動
 - コマンド行での入力, 219, 221
 - 情報のコピー
 - フロッピーディスクから, 219
 - フロッピーディスクへ, 220, 221
 - 取り出す
 - コマンド行での入力, 222
 - 内容の検査
 - コマンド行での入力, 218, 219
- フォーマット
 - DOS フロッピーディスク, 213, 215
 - UFS フロッピーディスク, 208, 210
 - 完全な形式, 206
 - サイズと密度のオプション, 206 - 209, 213
 - 注意, 208, 213
 - ハードウェアの考慮点, 205, 207
 - ファイルマネージャの問題点, 208, 213
 - フォーマットの取り消し, 209
 - フロッピーディスク名, 204
- マウント
 - 手作業と自動の比較, 186
 - リモートのフロッピーディスク, 223, 224
 - リモートアクセス
 - 他のシステム上でフロッピーディスクを使用できるようにする, 225
 - 他のシステム上のフロッピーディスクにアクセスする, 223, 224
 - ロード
 - ボリューム管理の使用, 216, 218
- フロッピーディスクドライブ, 183
 - アクセスポイント, 187
 - ボリューム管理下のディレクトリ, 254
 - プロファイルサーバー, 76
- へ
- ベースディレクトリ (basedir), 272, 273, 283, 290
- ヘッダー形式
 - 別のヘッダー形式でファイルをコピー (cpio), 672
- 別名
 - 削除, 17
 - ユーザーログイン名, 5
- 変更, 265
 - ユーザー ID 番号, 16
 - ユーザーアカウント用のディレクトリ所有権の変更, 16
 - ユーザーアカウント用のファイル所有権の変更, 16
 - ユーザーログイン名, 16

- ユーザーアカウント
 - Admintool: ユーザー, 16
 - コマンド, 13
- ユーザーパスワード
 - Admintool: ユーザー, 17
 - コマンド, 13
 - 頻度, 9, 20
 - ユーザーによる変更, 9
- 「変更」オプション (「編集」メニュー)
 - Admintool: グループ, 49
 - Admintool: ユーザー, 16
- 「編集」メニュー
 - Admintool: グループ
 - 「追加」オプション, 41
 - 「変更」オプション, 49
 - Admintool: ユーザー
 - 「追加」オプション, 42, 43
- 変数, 265
 - 環境, 25, 31
 - シェル (ローカル), 25
 - シェル (ローカル) 変数, 29
- ほ
- ホストマネージャ, 71
 - OS サービスの追加, 74
 - SunOS 4.0/4.1 の制限, 73, 75
 - 起動, 82
 - サポートするサーバーとクライアントの構成, 72
 - サポートの追加と管理, 72
 - システムタイプの更新, 73
 - 「システムタイプの更新」メニュー, 83
 - 実行する条件, 81
 - リモートインストールサービスの設定, 76
- ボリューム管理, 183, 253, 261
 - UFS の制限, 260
 - 再起動, 201
 - 終了, 201
 - 設定, 201
 - 手作業のマウントとの比較, 186
 - 取り外し可能媒体へのアクセス, 253
 - 利点, 186
 - CD
 - /cdrom マウントポイント, 256, 258
 - アクセス位置, 187, 253, 259
 - ディレクトリ, 253, 255, 256
 - シンボリックリンク
 - raw デバイスのアクセス, 259
 - ファイルシステムのアクセス, 258
 - フロッピーディスク
 - /floppy マウントポイント, 256, 258
 - アクセス位置, 187, 253, 255, 256
 - ディレクトリ, 254
 - ロード, 218
- ボリューム管理の使用
 - フロッピーディスク
 - ロード, 216
- ま
- マウント, 183
 - LOFS ファイルシステム, 469
 - NFS ファイルシステム, 469, 474
 - PCMCIA メモリーカード, 247
 - UFS ファイルシステム, 468
 - s5fs ファイルシステム, 475
 - vfstab ファイル内のすべてのファイル, 469
 - 手作業のマウントとの比較, 186
 - デフォルトオプションの使用, 472, 475
 - ファイルシステム, 443
 - ファイルシステムの自動マウント, 448
 - ユーザーのホームディレクトリ, 46
 - リモート CD, 194, 195
 - リモートのフロッピーディスク, 223, 224
 - マウントポイント
 - CD, 194, 256, 258
 - フロッピーディスク, 256, 258
 - ユーザーのホームディレクトリ
 - 自動マウント, 11
 - リモート, 44, 46
- マウントテーブル, 446
- マウントポイント, 444
- マスク, 31
- マニュアルページ, 438
- み
- 密度
 - フロッピーディスクのフォーマット, 206, 207

フロッピーディスクのフォーマット
DOS フロッピーディスク, 213
UFS フロッピーディスク, 208, 209
未割り当て i ノード, 539
未割り当てディレクトリブロック, 542

む

無効にする
実行制御スクリプト, 126
ユーザーアカウント
Admintool: ユーザー, 17
コマンド, 13
パスワード, 17, 20

め

メール別名
削除, 17
ユーザーログイン名, 5
メールボックス (ユーザー), 17
メモリー記憶域 (仮想), 440, 522

も

モニター (PROM), 171

ゆ

ユーザー ID 番号, 5, 6, 16
ユーザーアカウント, 3
ID 番号, 5, 6, 16
ガイドライン, 4, 11
管理ツール, 13
検索 (コマンド), 13
コマンド, 13
情報の格納, 5, 18, 21
説明, 4
ソート (コマンド), 13
ネームサービス, 5, 13, 18
ログイン名, 4, 5, 16
コピー
コマンド, 13
削除
Admintool: ユーザー, 17
コマンド, 13
設定
記録シート, 37

追加

Admintool: ユーザー, 42, 43
コマンド, 13

デフォルト設定
コマンド, 13

変更

Admintool: ユーザー, 16, 17
コマンド, 13

無効に、またはロックする
Admintool: ユーザー, 17
コマンド, 13

パスワード, 17, 20

ユーザーアカウントのソート, 13

ユーザーアクセス, 4

ユーザー初期設定 ファイル

カスタマイズ, 22, 33

シェル, 22 - 24, 33

説明, 4, 11, 22

デフォルト, 22

例, 32

カスタマイズ

概要, 11, 22, 23

カスタマイズファイルの追加, 17

環境変数, 26, 31

サイトの初期設定ファイル, 23

シェル変数, 26, 29

手順, 39

ユーザーマスクの設定, 31

ローカルシステムへの参照を避ける, 24

ユーザー登録

solregis コマンド, 59

説明, 57

無効にする, 60

問題, 59

ユーザーのホームディレクトリ

カスタマイズした初期設定ファイル, 11, 17

共有, 44, 46

サイズ, 10

削除, 17

所有権の変更, 16

説明, 4, 10

マウント, 46

ローカル以外からの \$HOME の参照, 11

ローカル以外からの \$HOME への参照, 24

マウント
自動マウント, 11
リモート, 44, 46
ユーザーのメールボックス, 17
ユーザーパスワードの有効期限, 10
ユーザーマスク, 31
ユーザーマネージャ、Solstice AdminSuite, 13
ユーザーログイン (疑似), 6
ユーザーログイン名
説明, 4, 5
変更, 16
ユーザーログイン名における下線 (_), 5

よ

呼び名

CD, 190
フロッピーディスク, 204

り

リスト表示

テープ上のファイル (cpio), 662
テープ上のファイル (tar), 657
フロッピーディスク上のファイル
(tar), 670

リセットボタン, 169

リトルエンディアン方式のビットコード
化, 203

リモートアクセス

CD

CD を他のシステムで使用できるように
にする, 196
他のシステム上の CD にアクセスす
る, 193, 195

フロッピーディスク

他のシステム上でフロッピーディスク
を使用できるようにする, 225
他のシステム上のフロッピーディス
クにアクセスする, 223, 224

リモートドライブ (からの復元), 620

リモートパッケージサーバー

スプールディレクトリへのソフ
トウェアインストール, 282
ソフトウェアのインストール, 279, 280
リモートマウント, 44, 46
領域最適化タイプ, 572
履歴環境変数, 27

る

ルート (/) ファイルシステム, 439
ソフトウェアパッケージのインストー
ル, 270, 283
ソフトウェアパッケージの削除, 297

れ

レベル 0 バックアップ, 588

ろ

ロード, 183

PCMCIA メモリーカード, 240

CD

コマンド行での入力, 191
フロッピーディスク
ボリューム管理の使用, 216, 218

ログイン名 (ユーザー)

説明, 4, 5
変更, 16

ログ (ダンプの記録), 630

論理デバイス名

CD-ROM, 332
定義, 327
ディスク, 327
テープ, 331

論理ブロックサイズ, 570

わ

ワールド (アクセス権), 31
割り当て済み i ノード, 539