Solaris のシステム管理 (デバイスとファイルシステム)

Part No: 819–0386–18
2010 年 9 月

ORACLE

Copyright © 2004, 2010, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

Oracle と Java は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。UNIX は X/Open Company, Ltd. からライセンスされている登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

100901@24661

目次

はじめに 21

1 リムーバブルメディアの管理 (概要) 27
リムーバブルメディアの新機能 27
サービス管理機能 (SMF) によって管理される vold 27
ボリューム管理の改善 (vold) 28
リムーバブルメディアの管理についての参照先 29
リムーバブルメディアの機能と利点 29
手動によるマウントと自動マウントの比較 30
リムーバブルメディアへのアクセスの概要 31

2 リムーバブルメディアの管理 (手順) 33
リムーバブルメディアの管理 (作業マップ) 33
フロッピーディスクをフォーマットする 34
リムーバブルメディアのハードウェア面での考慮事項 34
▼ リムーバブルメディアを読み込む方法 36
▼ フロッピーディスクをフォーマットする方法 (rmformat) 37
▼ リムーバブルメディア上にファイルシステムを作成する方法 38
▼ DVD-RAM 上にファイルシステムを作成する方法 40
▼ リムーバブルメディア上のファイルシステムを検査する方法 41
▼ リムーバブルメディア上の不良ブロックを修復する方法 41
リムーバブルメディアに読み取り/書き込み保護とパスワードによる保護を適用する 42
▼ リムーバブルメディアの書き込み保護を有効または無効にする方法 42
▼ リムーバブルメディアの読み取り/書き込み保護を有効または無効にする方法とパスワードを設定する方法 43

3 リムーバブルメディアへのアクセス (手順) 45
リムーバブルメディアへのアクセス (作業マップ) 45
リムーバブルメディアへのアクセス 46
リムーバブルメディア名の使用 46
リムーバブルメディア上のデータにアクセスするためのガイドライン 47
▼新しいリムーバブルメディアドライブを追加する方法 47
▼リムーバブルメディアサービスを無効または有効にする方法 48
▼リムーバブルメディア上の情報にアクセスする方法 49
▼リムーバブルメディア上の情報をコピーする方法 49
▼リムーバブルメディアが使用中かどうかを調べる方法 50
▼リムーバブルメディアを取り出す方法 51
リモートシステム上のリムーバブルメディアへのアクセス (作業マップ) 51
▼ローカルのメディアをほかのシステムで使用可能にする方法 52
▼リモートシステム上のリムーバブルメディアにアクセスする方法 53

4 CD および DVD への書き込み (手順) 55
オーディオ CD、データ CD、データ DVD の取り扱い 55
CD/DVD メディアに関するよく使われる用語 56
データ CD、データ DVD、オーディオ CD への書き込み 58
RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する 59
▼RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する方法 59
▼CD または DVD ライターを確認する方法 60
▼CD または DVD メディアを検査する方法 60
データ CD またはデータ DVD を作成する 61
▼データ CD またはデータ DVD 用の ISO 9660 ファイルシステムを作成する方法 . 61
▼マルチセッションのデータ CD を作成する方法 62
オーディオ CD を作成する 64
▼オーディオ CD を作成する方法 64
▼オーディオトラックを CD から抽出する方法 65
▼CD をコピーする方法 66
▼CD-RW メディアを消去する方法 67

5 デバイスの管理 (概要と手順) 69
デバイス管理の新機能 69

新しいホットプラグ機能 70
x86: デバイス検出ツール 70
PCI Express (PCIe) のサポート 70
USB と 1394 (FireWire) サポートの機能拡張 71
使用中デバイスのエラーチェック機能の向上 71
デバイス管理作業についての参照先 73
Solaris OS でのデバイス管理 73
デバイスの電力管理 73
デバイスドライバについて 75
デバイスの自動構成 75
デバイス構成情報の表示 77
システムへ周辺デバイスを追加する 81
▼ 周辺デバイスを追加する方法 81
▼ デバイスドライバを追加する方法 82
デバイスへのアクセス 83
デバイス情報が作成される方法 83
デバイスの管理方法 84
デバイス名の命名規則 84
論理ディスクデバイス名 85
論理テープデバイス名 88
論理リムーバブルメディアデバイス名 88

6 デバイスの動的構成 (手順) 89
動的再構成とホットプラグ機能 89
接続点 91
PCI または PCIe アダプタカードの取り外し 92
PCI または PCIe アダプタカードの取り付け 93
`hotplug` コマンドによる PCIe ホットプラグ 93
`cfgadm` コマンドによる SCSI ホットプラグ (作業マップ) 94
`cfgadm` コマンドによる SCSI ホットプラグ 95
▼ SCSI デバイスに関する情報を表示する方法 95
▼ SCSI コントローラの構成を解除する方法 96
▼ SCSI コントローラを構成する方法 97
▼ SCSI デバイスを構成する方法 97
▼ SCSI コントローラを切り離す方法 98

▼ SPARC: SCSI コントローラを接続する方法 99
▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法 100
▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法 101
▼ SPARC: SCSI デバイスを取り外す方法 102
SCSI 構成に関する問題の障害追跡 103
▼ 失敗した SCSI 構成解除操作の解決方法 104
`cfgadm` コマンドによる PCI または PCIe ホットプラグ (作業マップ) 105
`cfgadm` コマンドによる PCI または PCIe ホットプラグ 105
PCIe LED インジケータの動作 106
▼ PCI スロット構成情報を表示する方法 106
▼ PCI アダプタカードを取り外す方法 107
▼ PCI アダプタカードを取り付ける方法 109
PCI 構成に関する問題の障害追跡 110
Reconfiguration Coordination Manager (RCM) スクリプトの概要 111
RCM スクリプトについて 111
RCM スクリプトで実行できること 112
RCM スクリプト処理の動作方法 112
RCM スクリプトでの作業 113
アプリケーション開発者 RCM スクリプト (作業マップ) 113
システム管理者 RCM スクリプト (作業マップ) 114
RCM スクリプトに名前を付ける 115
RCM スクリプトのインストールまたは削除 115
▼ RCM スクリプトのインストール方法 115
▼ RCM スクリプトの削除方法 116
▼ RCM スクリプトのテスト方法 116
テープバックアップ用の RCM スクリプトの例 117

7 USB デバイスの使用 (概要) 121
USB デバイスの新機能 122
EHCI アイソクロナス転送のサポート 122
CDC ACM デバイスのサポート 122
USB デバイスのホットプラグ対応動作の変更 122
USB デバイスでの Oracle Solaris ZFS のサポート 123
Prolific および Keyspan シリアルアダプタのサポート 123
USB の電源割り当て 123

x86: GRUB ブートにおける USB CD および DVD のサポート 123
USB 仮想キーボードとマウスのサポート 124
vold によるホットプラグ対応 USB デバイスの認識 124
USB デバイスの Solaris サポート 125
USB デバイスの概要 126
よく使用される USB 関連の略語 126
USB バスの説明 127
Oracle Solaris OS における USB について 133
USB 2.0 機能 133
USB キーボードとマウス 135
USB ホストコントローラとハブ 136
USB ケーブルに関するガイドライン 138

8 USB デバイスの使用 (手順) 139
Oracle Solaris OS での USB デバイスの管理 (作業マップ) 139
USB 大容量ストレージデバイスの使用 (作業マップ) 140
USB 大容量ストレージデバイスの使用 142
USB フロッピーディスクデバイスの使用 143
非準拠 USB 大容量ストレージデバイスの使用 144
USB 大容量ストレージデバイスのホットプラグ 145
vold を使用した USB 大容量ストレージデバイスの使用の準備 148
▼ USB デバイス情報を表示する方法 149
▼ USB 大容量ストレージデバイス上にファイルシステムを作成する方法 150
▼ vold を使用しないで USB 大容量ストレージデバイス上にファイルシステムを作成する方法 152
▼ vold を使用しないで USB 大容量ファイルシステム上でパーティションを変更し、PCFS ファイルシステムを作成する方法 154
▼ USB 大容量ストレージデバイス上で vold を使用しないで Solaris パーティションの作成およびスライスの変更を行う方法 157
▼ vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法 159
USB 大容量ストレージデバイスの問題の障害追跡 161
特定の USB ドライバを無効にする 161
▼ 特定の USB ドライバを無効にする方法 162
▼ 使用されていない USB デバイスのリンクを削除する方法 162
USB オーディオデバイスの使用 (作業マップ) 163

USB オーディオデバイスの使用 163
複数の USB オーディオデバイスのホットプラグ 165
▼ USB オーディオデバイスを追加する方法 165
▼ システムの主オーディオデバイスを識別する方法 165
▼ 主 USB オーディオデバイスを変更する方法 166
USB オーディオデバイスに関する問題の障害追跡 166
cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ) 167
cfgadm コマンドを使った USB デバイスのホットプラグ 168
▼ USB バス情報を表示する方法 (cfgadm) 169
▼ USB デバイスの構成を解除する方法 170
▼ USB デバイスの構成方法 170
▼ 論理的に USB デバイスを接続解除する方法 171
▼ 論理的に USB デバイスを接続する方法 171
▼ 論理的に USB デバイスのサブツリーを接続解除する方法 172
▼ USB デバイスのリセット方法 172
▼ 複数の構成を持つ USB デバイスのデフォルト構成を変更する方法 172

9 InfiniBand デバイスの使用 (概要/手順) 175
InfiniBand デバイスの概要 175
IB デバイスの動的再構成 (作業マップ) 177
IB デバイスの動的再構成 (cfgadm) 178
▼ IB デバイス情報を表示する方法 179
▼ IOC デバイスの構成を解除する方法 180
▼ IOC デバイスを構成する方法 181
▼ IB Port、HCA_SVC、VPPA デバイスの構成を解除する方法 181
▼ IB Port、HCA_SVC、VPPA デバイスを構成する方法 182
▼ IB 擬似デバイスの構成を解除する方法 183
▼ IB 擬似デバイスを構成する方法 183
▼ HCA のカーネル IB クライアントを表示する方法 183
▼ HCA に接続されている IB デバイスの構成を解除する方法 184
IB HCA を構成する方法 185
▼ IB p_key テーブルを更新する方法 185
▼ IB 通信サービスを表示する方法 185
▼ VPPA 通信サービスを追加する方法 186
▼ 既存の IB Port、HCA_SVC、VPPA 通信サービスを削除する方法 186

▼IOC の構成を更新する方法 187
InfiniBand デバイスでの uDAPL アプリケーションインタフェースの使用 187
▼uDAPL を使用可能にする方法 188
DAT 静的レジストリの更新 189

10 ディスクの管理 (概要) 191
ディスク管理の新機能 191
Solaris OS のインストール時およびブート時における 2T バイトディスクのサポート 192
Solaris iSCSI ターゲットおよびイニシエータでの iSNS のサポート 193
Solaris iSCSI ターゲットのサポート 193
Solaris iSCSI イニシエータのサポート 194
x86: GRUB ブート環境でのディスク管理 195
2T バイトを超える SCSI ディスクのサポート 196
ディスク管理作業についての参照先 196
ディスク管理の概要 196
ディスク関連の用語 197
ディスクラベルについて 197
EFI ディスクラベル 198
ディスクスライスについて 202
`format` ユーティリティー 207
ディスクをパーティションに分割する 210
パーティションテーブル関連の用語 210
パーティションテーブル情報の表示 211
free hog スライスの使用方法 213

11 ディスクの管理 (手順) 215
ディスクの管理 (作業マップ) 215
システム上のディスクの確認 216
▼システム上のディスクを確認する方法 216
ディスクのフォーマット 219
▼ディスクがフォーマット済みかを調べる方法 219
▼ディスクをフォーマットする方法 220
ディスクスライスの表示 222
▼ディスクスライス情報を表示する方法 222

ディスクラベルの作成と検査 224
▼ディスクラベルを作成する方法 224
▼ディスクラベルを検査する方法 226
破損したディスクラベルの復元 228
▼破損したディスクラベルを復元する方法 228
他社製のディスクの追加 230
format.dat のエントリの作成 231
▼format.dat のエントリを作成する方法 231
SCSI ディスクドライブの自動構成 232
▼SCSI ドライブを自動構成する方法 233
欠陥セクターの修復 234
▼表面解析を使用して欠陥セクターを調べる方法 235
▼欠陥セクターを修復する方法 236
ディスク管理のヒント 236
format セッションのデバッグ 236
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 237

12 SPARC: ディスクの追加 (手順) 239
SPARC: システムディスクまたは二次ディスクの追加 (作業マップ) 239
SPARC: システムディスクまたは二次ディスクの追加 240
▼SPARC: システムディスクを接続してブートする方法 241
▼SPARC: 二次ディスクを接続してブートする方法 242
▼SPARC: ディスクスライスとディスクラベルを作成する方法 243
▼SPARC: UFS ファイルシステムを作成する方法 248
▼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 249

13 x86: ディスクの追加 (手順) 251
x86: システムディスクまたは二次ディスクの追加 (作業マップ) 251
x86: システムディスクまたは二次ディスクの追加 253
▼x86: システムディスクの接続方法 253
fdisk パーティション識別子の変更 254
▼Solaris fdisk 識別子を変更する方法 254
▼x86: 二次ディスクを接続してブートする方法 255
x86: fdisk パーティションの作成上のガイドライン 257
▼x86: Solaris fdisk パーティションを作成する方法 258

▼x86: ディスクスライスとディスクラベルを作成する方法 263
▼x86: ファイルシステムを作成する方法 264
▼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 265

14 Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順) 267
iSCSI 技術 (概要) 268
Solaris iSCSI のソフトウェア要件およびハードウェア要件 269
Solaris iSCSI ターゲットおよびイニシエータの設定 (作業マップ) 269
Solaris iSCSI ターゲットおよびイニシエータの構成 271
▼Solaris iSCSI 構成の準備を整える方法 273
Solaris iSCSI ターゲットデバイスの設定 273
iSCSI ベースのストレージネットワークにおける認証の構成 275
他社製の RADIUS サーバーを使用して iSCSI 構成内の CHAP 管理を単純化する 278
▼iSCSI ターゲット発見を構成する方法 279
▼発見された iSCSI ターゲットを削除する方法 280
iSCSI ディスクへのアクセス 282
▼リブート時に iSCSI ディスクにアクセスする方法 282
▼iSCSI 構成の監視 284
iSCSI イニシエータと iSCSI ターゲットのパラメータの変更 286
Solaris iSCSI マルチパスデバイスの設定 290
▼ターゲットの複数の iSCSI セッションを有効にする方法 291
iSCSI 構成に関する問題の障害追跡 294
ローカルシステムから iSCSI ターゲットに接続できない 294
ローカルシステム上で iSCSI デバイスまたは iSCSI ディスクが利用できない 295
iSNS 発見方式を使用する場合の LUN マスクの使用 296
iSCSI の一般的なエラーメッセージ 296

15 `format` ユーティリティー (参照情報) 303
`format` ユーティリティーを使用する上での推奨事項および要件 303
`format` のメニューとコマンドの説明 304
`partition` メニュー 306
x86: `fdisk` メニュー 307
`analyze` メニュー 308
`defect` メニュー 309

format.dat ファイル 310
format.dat ファイルの内容 311
format.dat ファイルの構文 311
format.dat ファイル中のキーワード 311
パーティションテーブル (format.dat) 314
format ユーティリティーの代替データファイルを指定する 315
format コマンドへの入力規則 315
format コマンドへ番号を指定する 315
format コマンドへブロック番号を指定する 316
format のコマンド名を指定する 316
format コマンドへディスク名を指定する 317
format ユーティリティーのヘルプを利用する 317

16 ファイルシステムの管理 (概要) 319
ファイルシステムの新機能 319
ファイルシステム監視ツール (fsstat) 320
Oracle Solaris ZFS ファイルシステム 320
UFS ファイルシステムユーティリティー (fsck、mkfs、および newfs) の拡張機能 321
ファイルシステム管理作業についての参照先 327
ファイルシステムの概要 328
ファイルシステムのタイプ 328
UFS ファイルシステム管理用のコマンド 336
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 337
汎用ファイルシステムコマンドと専用ファイルシステムコマンドのマニュアルページ 337
デフォルトの Solaris ファイルシステム 338
UFS ファイルシステム 339
UFS ファイルシステムの機能 339
UFS ファイルシステムの計画 340
64 ビット: マルチテラバイトの UFS ファイルシステムのサポート 341
UFS ロギング 344
UFS スナップショット 345
UFS 直接入出力 346
ファイルシステムのマウントとマウント解除 346
マウントされたファイルシステムテーブル 348

仮想ファイルシステムテーブル 349
NFS 環境 350
自動マウント (autofs) 351
ファイルシステムのタイプを調べる 352
ファイルシステムのタイプを調べる方法 352

17 ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順) 355
Oracle Solaris ZFS ファイルシステムの作成 355
▼ Oracle Solaris ZFS ファイルシステムの作成方法 356
UFS ファイルシステムの作成 356
▼ UFS ファイルシステムを作成する方法 357
▼ マルチテラバイトの UFS ファイルシステムを作成する方法 359
▼ マルチテラバイトの UFS ファイルシステムを拡張する方法 360
▼ UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法 361
マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡 362
一時ファイルシステム (TMPFS) の作成 363
▼ TMPFS ファイルシステムを作成およびマウントする方法 363
ループバックファイルシステム (LOFS) の作成およびマウント 365
▼ LOFS ファイルシステムを作成およびマウントする方法 365

18 ファイルシステムのマウントとマウント解除 (手順) 367
ファイルシステムのマウントの概要 367
UFS 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 368
UFS の汎用マウントオプション 369
`/etc/vfstab` ファイルのフィールドの説明 371
ファイルシステムのマウント 373
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 373
▼ `/etc/vfstab` ファイルにエントリを追加する方法 374
▼ 1 つのファイルシステムをマウントする方法 (`/etc/vfstab` ファイル) 375
▼ UFS ファイルシステムのマウント方法 (`mount` コマンド) 377
▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (`mount` コマンド) 378
▼ NFS ファイルシステムのマウント方法 (`mount` コマンド) 379
▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法

(mount コマンド) 380
UFS ファイルシステムをマウント解除する 381
ファイルシステムをマウント解除する場合の前提条件 382
ファイルシステムのマウント解除を確認する方法 382
▼ 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 382
▼ 1 つのファイルシステムをマウント解除する方法 383

19 CacheFS ファイルシステムの使用 (手順) 385
CacheFS ファイルシステムの概観 (作業マップ) 385
CacheFS ファイルシステムの概要 386
CacheFS ファイルシステムの機能 386
CacheFS ファイルシステムの構造と動作 387
CacheFS ファイルシステムの作成とマウント (作業マップ) 388
▼ キャッシュを作成する方法 389
ファイルシステムをキャッシュにマウントする 389
▼ CacheFS ファイルシステムのマウント方法 (mount) 390
▼ CacheFS ファイルシステムのマウント方法 (/etc/vfstab) 392
▼ CacheFS ファイルシステムのマウント方法 (autofs) 393
CacheFS ファイルシステムの保守 (作業マップ) 394
CacheFS ファイルシステムの保守 394
CacheFS ファイルシステムの変更 395
▼ CacheFS ファイルシステムに関する情報を表示する方法 395
CacheFS ファイルシステムの整合性検査 396
▼ 必要に応じてキャッシュの整合性検査を指定する方法 396
▼ CacheFS ファイルシステムを削除する方法 397
▼ CacheFS ファイルシステムの整合性を検査する方法 398
CacheFS ファイルシステムのパッキング (作業マップ) 399
CacheFS ファイルシステムのパッキング 400
▼ キャッシュにファイルをパックする方法 400
▼ パックされたファイルの情報を表示する方法 401
パッキングリストの使用 402
▼ パッキングリストを作成する方法 403
▼ パッキングリストを使ってファイルをキャッシュにパックする方法 403
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 404
▼ 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 404
cachefspack エラーの障害追跡 405
CacheFS の統計情報の収集 (作業マップ) 409
CacheFS の統計情報の収集 409
▼ CacheFS ロギングを設定する方法 410
▼ CacheFS ログファイルの場所を調べる方法 411
CacheFS ロギングを停止する方法 412
▼ 作業セット (キャッシュ) のサイズを表示する方法 412
CacheFS の統計情報の表示 413
▼ CacheFS 統計情報を表示する方法 413

20 追加スワップ空間の構成 (手順) 415
スワップ空間について 415
スワップ空間と仮想メモリー 416
スワップ空間と TMPFS ファイルシステム 416
ダンプデバイスとしてのスワップ空間 417
スワップ空間と動的再構成 418
SAN 環境でのスワップ空間の構成 418
スワップ空間の追加が必要かどうかを調べる方法 419
スワップ関連のエラーメッセージ 419
TMPFS 関連のエラーメッセージ 419
スワップ空間の割り当て方法 420
スワップ空間と /etc/vfstab ファイル 420
スワップ空間の計画 420
UFS ベースのシステムでのスワップ空間の割り当て 421
ZFS ベースのシステムでのスワップ空間の割り当て 422
スワップリソースの監視 422
スワップ空間の追加 424
UFS ルート環境でのスワップファイルの作成 424
▼ UFS ルート環境でスワップファイルを作成して使用可能にする方法 425
Oracle Solaris ZFS ルート環境でのスワップ空間の追加または変更 426
▼ Oracle Solaris ZFS ルート環境でスワップ空間を追加する方法 427
スワップファイルの削除 428
▼ 不要になったスワップ空間を削除する方法 428

21 UFS ファイルシステムの整合性検査 (手順) 429
ファイルシステムの整合性 430
ファイルシステムの状態はどのように記録されるか 430
fsck コマンドで検査して修復される内容 431
UFS ファイルシステムの不整合が発生する理由 431
整合性が検査される UFS 構成要素 432
fsck 要約メッセージ 437
UFS ファイルシステムを対話式で検査して修復する 438
▼ 代替ブートデバイスからルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムを検査する方法 439
▼ ルート (/)、/usr、または /var 以外のファイルシステムを検査する方法 441
UFS ファイルシステムの修復 443
▼ UFS ファイルシステムを修復する方法 444
fsck コマンドで修復できない UFS ファイルシステムの修正 444
不正なスーパーブロックの復元 445
▼ 不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース) 446
▼ 不正なスーパーブロックを復元する方法 (Solaris 8、9、および 10 リリース) 449
fsck コマンドの構文とオプション 450

22 UFS ファイルシステム (参照情報) 453
UFS ファイルシステムのシリンダグループの構造 453
ブートブロック 454
スーパーブロック 454
i ノード 454
データブロック 456
空きブロック 456
UFS ファイルシステムパラメータのカスタマイズ 457
論理ブロックサイズ 457
フラグメントサイズ 458
最小空き容量 458
回転待ち 459
最適化のタイプ 459
i ノード数 (ファイルの数) 459
UFS ファイルとファイルシステムの最大サイズ 460
UFS サブディレクトリの最大数 460

23 UFS ファイルシステムのバックアップと復元 (概要) 461
UFS バックアップと復元についての参照先 461
UFS ファイルシステムのバックアップと復元の概要 462
ファイルシステムをバックアップする理由 463
どの UFS ファイルシステムをバックアップするかの計画 463
バックアップタイプの選択 466
テープデバイスの選択 467
UFS ファイルシステムのバックアップおよび復元の概要 (作業マップ) 467
バックアップスケジュールに関するその他の注意事項 468
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 469
ダンプレベルを使用して増分バックアップを作成する 471
バックアップスケジュールの例 472

24 UFS ファイルとファイルシステムのバックアップ (手順) 479
UFS ファイルとファイルシステムのバックアップ (作業マップ) 479
ファイルシステムバックアップの実行準備 480
▼ UFS ファイルシステム名を検索する方法 480
▼ 完全バックアップに必要なテープ数を決定する 481
UFS ファイルシステムのバックアップ 481
▼ UFS ファイルシステムのバックアップをテープに作成する方法 482

25 UFS スナップショットの使用 (手順) 489
UFS スナップショットの使用 (作業マップ) 489
UFS スナップショットの概要 490
なぜ UFS スナップショットを使用するか 491
UFS スナップショットのパフォーマンス上の問題 492
UFS スナップショットの作成と削除 492
マルチテラバイト UFS のスナップショットの作成 493
▼ UFS スナップショットを作成する方法 494
▼ UFS スナップショットの情報を表示する方法 495
UFS スナップショットの削除 496
▼ UFS スナップショットを削除する方法 496
UFS スナップショットのバックアップ 497
▼ UFS スナップショットの完全バックアップの作成方法 (`ufsdump`) 497
▼ UFS スナップショットの増分バックアップの作成方法 (`ufsdump`) 498

▼ UFS スナップショットのバックアップ方法 (tar) 498
UFS スナップショットのバックアップからのデータの復元 499

26 UFS ファイルとファイルシステムの復元 (手順) 501
UFS ファイルおよびファイルシステムのバックアップを復元する (作業マップ) 501
UFS ファイルとファイルシステムを復元するための準備 502
UFS ファイルシステム名の確認 503
必要なテープデバイスのタイプの決定 503
テープデバイス名の決定 503
UFS ファイルとファイルシステムの復元 503
▼ 使用するテープを決定する方法 504
▼ 対話式で UFS ファイルを復元する方法 505
▼ 特定の UFS ファイルを復元する方法 507
▼ UFS ファイルシステム全体を復元する方法 509
▼ UFS ルート (/) および /usr ファイルシステムを復元する方法 513

27 UFS バックアップおよび復元コマンド (参照情報) 517
ufsdump コマンドの機能 517
デバイス特性の判断 517
メディアの終わりの検出 518
ufsdump コマンドを使用したデータのコピー 518
/etc/dumpdates ファイルの目的 518
バックアップデバイス (*dump-file*) 引数 519
バックアップを作成するファイルを指定する 521
テープの性質を指定する 521
ufsdump コマンドの制限 521
ufsdump コマンドオプションおよび引数の指定 522
ufsdump のデフォルトオプション 522
ufsdump とセキュリティーに関する注意事項 523
ufsrestore オプションおよび引数の指定 523

28 UFS ファイルとファイルシステムのコピー (手順) 525
ファイルシステムをコピーするためのコマンド 526
ファイルシステムをディスク間でコピーする 528

ファイルシステムのリテラルコピーを作成する 528
▼ディスクをコピーする方法 (dd) 529
cpio を使用してファイルシステム間でディレクトリをコピーする 532
▼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) 533
ファイルとファイルシステムをテープにコピーする 534
tar を使用してファイルをテープにコピーする 534
▼ファイルをテープにコピーする方法 (tar) 535
▼テープ上のファイルのリストを表示する方法 (tar) 536
▼テープからファイルを取り出す方法 (tar) 536
pax コマンドを使用してファイルをテープにコピーする 537
▼ファイルをテープにコピーする方法 (pax) 537
cpio コマンドを使用してファイルをテープにコピーする 538
▼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) 538
▼テープ上のファイルのリストを表示する方法 (cpio) 539
▼テープからすべてのファイルを取り出す方法 (cpio) 540
▼テープから特定のファイルを取り出す方法 (cpio) 541
ファイルをリモートテープデバイスにコピーする 542
▼ファイルをリモートテープデバイスにコピーする方法 (tar と dd) 542
▼ファイルをリモートテープデバイスから抽出する方法 543
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 544
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 544
▼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) 545
▼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) 546
▼ファイルをフロッピーディスクから取り出す方法 (tar) 546
複数のフロッピーディスクへファイルをアーカイブする 547

29 テープドライブの管理 (手順) 549
使用するメディアの選択 549
バックアップデバイス名 550
テープドライブの巻き戻しオプションを指定する 551
テープドライブに別の密度を指定する 552
テープドライブの状態を表示する 552
▼テープドライブの状態を表示する方法 552
磁気テープカートリッジの取り扱い 553

磁気テープカートリッジのたるみを直す 553
磁気テープカートリッジを巻き戻す 554
ドライブの管理とメディア処理のガイドライン 554

索引 555

はじめに

本書『Solaris のシステム管理 (デバイスとファイルシステム)』は、Oracle Solaris システム管理に関する重要な情報を提供するマニュアルの一部です。本書には、SPARC および x86 の両方のシステムに関する情報が含まれています。

本書では、次のことを前提としています。

- SunOS 5.10 オペレーティングシステムがインストールされていること
- 使用するネットワークソフトウェアの設定が完了していること

システム管理者にとって重要な Oracle Solaris 10 リリースの新機能については、各章のはじめにある新機能に関する節を参照してください。

注 – この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーをサポートしています。サポートされるシステムについては、Solaris 10 Hardware Compatibility List (http://www.sun.com/bigadmin/hcl) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- 「x64」は、特に 64 ビットの x86 互換 CPU を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、Solaris 10 Hardware Compatibility List を参照してください。

注 – このマニュアルで説明する Sun 以外の Web サイトの利用については、Sun は責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスのご利用あるいは信頼によって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

対象読者

このマニュアルは、Solaris Oracle Solaris 10 リリースが動作しているシステムの管理者を対象としています。このマニュアルを読むには、 UNIX のシステム管理について 1 - 2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Solaris システム管理マニュアルセットの構成

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

マニュアルのタイトル	内容
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、およびサービスの管理
『Solaris のシステム管理 (上級編)』	端末とモデムの設定、システムリソースの管理 (ディスク割り当て、アカウンティング、および crontab ファイルの管理)、システムプロセスの管理、および Oracle Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク管理、IPv4 と IPv6 アドレス管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチのパス化 (IPMP)、および IPQoS
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、および LDAP のネーミングとディレクトリサービス (NIS から LDAP への移行、および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』	NIS+ のネーミングとディレクトリサービス

マニュアルのタイトル	内容
『Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と autofs)、メール、SLP、および PPP
『Solaris のシステム管理 (印刷)』	印刷に関するトピックや、サービス、ツール、プロトコル、およびテクノロジを使って印刷サービスおよびプリンタを設定および管理する方法
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティー、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、および Solaris Secure Shell
『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』	リソース管理に関連する計画と作業、拡張アカウンティング、リソース制御、フェアシェアスケジューラ (FSS)、資源上限デーモン (rcapd) による物理メモリーの制御、および資源プール (Solaris Zones ソフトウェア区分技術と lx ブランドゾーンによる仮想化)
『Oracle Solaris ZFS 管理ガイド』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Oracle Solaris Trusted Extensions 管理の手順』	Oracle Solaris Trusted Extensions 機能固有のシステム管理
『Oracle Solaris Trusted Extensions 構成ガイド』	Solaris 10 5/08 リリース以降での、Oracle Solaris Trusted Extensions 機能の計画、有効化、および初期設定の方法

マニュアル、サポート、およびトレーニング

追加リソースについては、次の Web サイトを参照してください。

- マニュアル (`http://docs.sun.com`)
- サポート (`http://www.oracle.com/us/support/systems/index.html`)
- トレーニング (`http://education.oracle.com`) – 左のナビゲーションバーで「Sun」のリンクをクリックします。

Oracle へのご意見

Oracle はドキュメントの品質向上のために、お客様のご意見やご提案をお待ちしています。誤りを見つけたり、改善に向けた提案などがある場合は、http://docs.sun.com で「Feedback」をクリックしてください。可能な場合には、ドキュメントのタイトルやパート番号に加えて、章、節、およびページ番号を含めてください。返信を希望するかどうかもお知らせください。

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) では、Oracle ソフトウェアに関する広範なリソースが提供されています。

- ディスカッションフォーラム (http://forums.oracle.com) で技術的な問題や解決策を話し合う。
- Oracle By Example (http://www.oracle.com/technology/obe/start/index.html) のチュートリアルで、手順に従って操作を体験する。
- サンプルコード (http://www.oracle.com/technology/sample_code/index.html) をダウンロードする。

表記上の規則

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

表 P–1　表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	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 *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。

表 P–1 表記上の規則 (続き)

字体または記号	意味	例
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	`sun% grep '^#define \` `XV_VERSION_STRING'`

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- C シェル

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

- C シェルのスーパーユーザー

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

- Bash シェル、Korn シェル、および Bourne シェル

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

- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー

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

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

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

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

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

一般規則

このマニュアルでは次の規則が使用されています。

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

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

◆ ◆ ◆ 第 1 章

1 リムーバブルメディアの管理 (概要)

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

この章の内容は以下のとおりです。

- 27 ページの「リムーバブルメディアの新機能」
- 29 ページの「リムーバブルメディアの管理についての参照先」
- 29 ページの「リムーバブルメディアの機能と利点」
- 30 ページの「手動によるマウントと自動マウントの比較」
- 31 ページの「リムーバブルメディアへのアクセスの概要」

リムーバブルメディアの新機能

次の節では、Solaris リリースにおけるリムーバブルメディアの新機能について説明します。

- 27 ページの「サービス管理機能 (SMF) によって管理される vold」
- 28 ページの「ボリューム管理の改善 (vold)」

Solaris の新機能の一覧および Solaris リリースについての説明は、『Oracle Solaris 10 9/10 の新機能』を参照してください。

サービス管理機能 (SMF) によって管理される vold

Solaris 10 1/06: ボリューム管理デーモンである `vold` は、サービス管理機能 (SMF) によって管理されるようになりました。つまり、該当する場合は、`svcadm disable` コマンドを使用して、次の新しい volfs サービスを無効にすることができます。

```
# svcadm disable volfs
```

このコマンドを使用して、`volfs` サービスのステータスを識別できます。

```
$ svcs volfs
STATE          STIME    FMRI
online         Sep_29   svc:/system/filesystem/volfs:default
```

詳細は、smf(5)を参照してください。

svccfg コマンドを使用して、追加の vold プロパティーを表示して設定できます。たとえば、vold のロギングを一時的に有効にして、問題のトラブルシューティングに役立てることができます。次に例を示します。

```
# svccfg
svc:> select system/filesystem/volfs
svc:/system/filesystem/volfs> setprop vold/log_debuglevel=3
svc:/system/filesystem/volfs> exit
# svcadm disable volfs
# svcadm enable volfs
```

svccfg コマンドを使用して、設定可能な vold プロパティーの一覧を表示することもできます。

```
# svccfg
svc:> select volfs
svc:/system/filesystem/volfs> listprop vold/*
vold/config_file            astring
vold/log_debuglevel         count    3
vold/log_file               astring
vold/log_nfs_trace          boolean  false
vold/log_verbose            boolean  false
vold/root_dir               astring
vold/never_writeback_label  boolean  false
svc:/system/filesystem/volfs> exit
```

これらのプロパティーの詳細は、vold(1M) のマニュアルページを参照してください。

ボリューム管理の改善 (vold)

Solaris 10 1/06: リムーバブルメディアの管理が改善されました。以前は、vold は、メディアを含んでいないリムーバブルデバイスのデバイスリンクを作成しませんでした。現在は、メディアを含んでいないデバイスのリンクが次のように適切に作成されます。

```
lrwxrwxrwx   1 root     root     28 Jun 13 13:09 /vol/dev/aliases/cdrom0 ->
 /vol/dev/rdsk/c2t2d0/nomedia
```

現在は、cdrw コマンドおよび rmformat コマンドを使用して、vold の実行時にメディアを含んでいないデバイスを一覧表示できます。

次のように /etc/vold.conf ファイル内の support nomedia エントリを変更することによって、以前の vold の動作に戻すことができます。

```
support media
```

次に、vold を再起動します。詳細は、vold.conf(4) のマニュアルページを参照してください。

さらに、vold はホットプラグ対応になりました。この結果、リムーバブルメディアを挿入しただけで、vold によってメディアが自動的に検出およびマウントされるようになりました。vold を手動で再起動しなくても、すべてのリムーバブルメディアデバイスのファイルシステムが認識されてデバイスからマウントされます。

USB ではない古いフロッピーディスクデバイスを使用している場合は、volcheck コマンドを実行しないと vold がメディアを認識しないことがあります。

メディアが検出されてもなんらかの理由でマウント解除された場合は、次のコマンドを実行する必要があります。

```
# volrmmount -i rmdisk0
```

リムーバブルメディアデバイスをホットリムーブする場合は、その前にメディアを取り出してください。次に例を示します。

```
# eject rmdisk0
```

リムーバブルメディアの管理についての参照先

リムーバブルメディアを管理する手順については、次を参照してください。

リムーバブルメディアの管理作業	参照先
リムーバブルメディアへのアクセス	第 3 章「リムーバブルメディアへのアクセス (手順)」
リムーバブルメディアのフォーマット	第 2 章「リムーバブルメディアの管理 (手順)」
データ CD、データ DVD、および音楽 CD の書き込み	第 4 章「CD および DVD への書き込み (手順)」

リムーバブルメディアの機能と利点

Solaris リリースには、ユーザーとソフトウェア開発者用に、リムーバブルメディアを扱うための標準インタフェースが用意されています。リムーバブルメディアサービスには、次の利点があります。

- リムーバブルメディアを自動的にマウントします。手動によるマウントと自動マウントの比較については、次の節を参照してください。

- スーパーユーザーでなくても、リムーバブルメディアにアクセスできるようになります。
- ネットワーク上のほかのシステムがローカルシステム上のリムーバブルメディアに自動的にアクセスできるようになります。詳細は、第 3 章「リムーバブルメディアへのアクセス (手順)」を参照してください。

手動によるマウントと自動マウントの比較

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

表 1–1 リムーバブルメディアの手動によるマウントと自動マウントの比較

手順	手動によるマウント	自動マウント
1	メディアを挿入します。	メディアを挿入します。
2	スーパーユーザーになります。	フロッピーディスクの場合は、`volcheck` コマンドを使用します。
3	メディアデバイスの位置を確認します。	リムーバブルメディアサービスは、リムーバブルメディアを手動でマウントして使用するのに必要な多くの作業を自動的に実行します。
4	マウントポイントを作成します。	
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します。	
6	`mount` に適切なオプションを付けて、デバイスをマウントします。	
7	スーパーユーザーを終了します。	
8	メディア上のファイルを操作します。	メディア上のファイルを操作します。
9	スーパーユーザーになります。	
10	メディアデバイスのマウントを解除します。	
11	メディアを取り出します。	メディアを取り出します。
12	スーパーユーザーを終了します。	

リムーバブルメディアへのアクセスの概要

リムーバブルメディアサービスを使用すると、手動によるマウントの場合と同様にリムーバブルメディアにアクセスできますが、その操作ははるかに容易になり、スーパーユーザーになる必要もありません。

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

表 1–2 リムーバブルメディア上のデータにアクセスする方法

アクセス	操作	ファイルを検索する場所
最初のフロッピーディスク上のファイル	フロッピーディスクを挿入して、コマンド行で `volcheck` と入力します	`/floppy`
リムーバブルハードディスク上のファイル	リムーバブルハードディスクを挿入して、コマンド行で `volcheck` と入力します	`/rmdisk/rmdisk0` または `/rmdisk/rmdisk1`
最初の CD 上のファイル	CD を挿入して、数秒間待ちます	`/cdrom/`*volume-name*
最初の DVD 上のファイル	DVD を挿入して、数秒間待ちます	`/cdrom/`*volume-name*

表 1–3 リムーバブルメディアにアクセスする場所

ファイルシステムにアクセスするためのパス	**raw** データにアクセスするための場所
`/floppy/floppy0`	`/vol/dev/aliases/floppy0`
`/floppy/floppy1`	`/vol/dev/aliases/floppy1`
`/cdrom/cdrom0`	`/vol/dev/aliases/cdrom0`
`/cdrom/cdrom1`	`/vol/dev/aliases/cdrom1`
`/rmdisk/rmdisk0` または `/rmdisk/rmdisk1`	`/vol/dev/aliases/rmdisk0` または `/vol/dev/aliases/rmdisk1`
`/pcmem/pcmem0`	`/vol/dev/aliases/pcmem0`

◆ ◆ ◆ 第 2 章

2 リムーバブルメディアの管理 (手順)

この章では、Solaris OS でコマンド行からリムーバブルメディアを管理する方法について説明します。

リムーバブルメディアの管理に関連する手順については、33 ページの「リムーバブルメディアの管理 (作業マップ)」を参照してください。リムーバブルメディアの概要については、第 1 章「リムーバブルメディアの管理 (概要)」を参照してください。

リムーバブルメディアの管理 (作業マップ)

次の作業マップでは、リムーバブルメディアを管理するための作業について説明します。

作業	説明	参照先
1. メディアをロードします。	フロッピーディスクをドライブに挿入して、`volcheck` コマンドを実行します。	36 ページの「リムーバブルメディアを読み込む方法」
2. (省略可能) フロッピーディスクをフォーマットします。	フロッピーディスクをフォーマットします。	37 ページの「フロッピーディスクをフォーマットする方法 (`rmformat`)」
3. (省略可能) UFS または PCFS ファイルシステムを追加します。	ファイル転送にメディアを使用する場合は、UFS または PCFS ファイルシステムを追加します。	38 ページの「リムーバブルメディア上にファイルシステムを作成する方法」
	UFS または UDFS ファイルシステムを DVD-RAM デバイスに追加します。	40 ページの「DVD-RAM 上にファイルシステムを作成する方法」

作業	説明	参照先
4. (省略可能) メディアを検査します。	メディア上のファイルシステムの整合性を検査します。	41 ページの「リムーバブルメディア上のファイルシステムを検査する方法」
5. (省略可能) メディア上の不良ブロックを修復します。	必要に応じて、メディア上に不良ブロックがあれば修復します。	41 ページの「リムーバブルメディア上の不良ブロックを修復する方法」
6. (省略可能) 読み取り/書き込み保護とパスワードによる保護を適用します。	必要に応じて、読み取り/書き込み保護やパスワードによる保護をメディアに適用します。	42 ページの「リムーバブルメディアの書き込み保護を有効または無効にする方法」

フロッピーディスクをフォーマットする

書き換え可能なフロッピーディスクをフォーマットしたり保護したりするには、`rmformat` コマンドを使用します。このユーティリティーには、vold が実行中の場合を除き、スーパーユーザー特権は必要ありません。ファイルシステムは自動的にマウントされます。そのため、メディアに既存のファイルシステムが含まれている場合は、メディアをフォーマットする前にマウント解除する必要があります。

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

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

リムーバブルメディアのハードウェア面での考慮事項

フロッピーディスクを使用するときは、次の制限事項に注意してください。

- SPARC と x86 とでは UFS フォーマットが異なります。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。UFS 用にフォーマットされたメディアは、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 UFS 用にフォーマットされたフロッピーディスクは、x86 システム上の UFS には使用できません。同様に、x86 システムで UFS 用にフォーマットされたフロッピーディスクは SPARC システムでは使用できません。

- SunOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、SunOS ファイルシステムをサポートするための構造からなります。DOS ファイルシステム用の完全な形式は、基本的な「ビット」形式と、MS-DOS または NEC-DOS のどちらかのファイルシステムをサポートする構造からなります。メディアを準備するために必要な手順は、ファイルシステムによって異なります。したがって、フロッピーディスクをフォーマットする前には、どの作業が必要かを決めてください。詳細は、33 ページの「リムーバブルメディアの管理 (作業マップ)」を参照してください。

フロッピーディスクのハードウェア面での考慮事項

フロッピーディスクをフォーマットするときは、次のことに注意してください。

- フロッピーディスク名については、表 3–1 を参照してください。
- 名前の付いていない (「ラベル」がない) フロッピーディスクには、`unnamed_floppy` というデフォルト名が割り当てられます。
- 名前の付いていない (「ラベル」がない) フロッピーディスクには、`floppy` というデフォルト名が割り当てられます。

Solaris システムは、次の種類のフロッピーディスクをフォーマットできます。

- UFS
- MS-DOS または NEC-DOS (PCFS) 用
- UDFS

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

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

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

▼ リムーバブルメディアを読み込む方法

リムーバブルメディアのハードウェア面での考慮事項については、34 ページの「リムーバブルメディアのハードウェア面での考慮事項」を参照してください。

1. メディアを挿入します。

2. メディアがフォーマットされていることを確認します。

   フォーマットされているかどうかがわからない場合は、メディアを挿入し、手順 3 の説明に従って、システムコンソールウィンドウの状態メッセージを確認してください。メディアをフォーマットする必要がある場合は、37 ページの「フロッピーディスクをフォーマットする方法 (rmformat)」を参照してください。

3. (省略可能) USB でない古いフロッピーディスクデバイスを使用している場合は、ボリューム管理に通知します。

   ```
   $ volcheck -v
   ```

   次の 2 つの状態メッセージのいずれかが表示されます。

`media was found`	ボリューム管理がメディアを検出し、表 3–1 に記述されたディレクトリにそのメディアをマウントしようとします。
	
	メディアが正しくフォーマットされている場合は、コンソールにエラーメッセージが表示されません。
	
	メディアがフォーマットされていない場合でも「`media was found`」メッセージは表示されます。ただし、次のようなエラーメッセージがシステムコンソールウィンドウに表示されます。
	
	`fd0: unformatted diskette or no diskette in the drive`
	
	`fd0: read failed (40 1 0)`
	
	`fd0: bad format`
	
	メディアをフォーマットしてからでないと、ボリューム管理はそれをマウントできません。詳細については、第 2 章「リムーバブルメディアの管理 (手順)」を参照してください。

`no media was found`	ボリューム管理は、メディアを検出しませんでした。メディアが正しく挿入されていることを確認して、`volcheck` をもう一度実行してください。うまくいかない場合は、メディアを検査してください。損傷の可能性があります。メディアを手動でマウントしてみることもできます。

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

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

ヒント – floppy0 はフロッピーディスクの実際の名前へのシンボリックリンクです。この場合、myfiles が実際の名前です。名前がないが正しくフォーマットされている場合は、unnamed_floppy と呼ばれます。

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

/floppy/*name* on /vol/dev/diskette0/ *name*

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

▼ フロッピーディスクをフォーマットする方法 (rmformat)

rmformat コマンドを使用すると、フロッピーディスクをフォーマットできます。デフォルトでは、このコマンドによってメディア上にパーティション 0 とパーティション 2 (メディア全体) という 2 つのパーティションが作成されます。

1 リムーバブルメディアサービスが動作していることを確認します。動作していれば、デバイス名のニックネームを使用できます。

```
$ svcs volfs
STATE          STIME    FMRI
online         10:39:12 svc:/system/filesystem/volfs:default
```

リムーバブルメディアサービスを再起動する方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。メディアのデバイス名の確認方法については、46 ページの「リムーバブルメディア名の使用」を参照してください。

2 フロッピーディスクをフォーマットします。

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

rmformat のフォーマットオプションについては、34 ページの「フロッピーディスクをフォーマットする」を参照してください。

rmformat コマンドの出力によって不良ブロックが見つかった場合は、41 ページの「リムーバブルメディア上の不良ブロックを修復する方法」を参照してください。

3 (省略可能) フロッピーディスクに 8 文字のラベルを付けます。

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

DOS ラベルの作成の詳細は、mkfs_pcfs(1M) のマニュアルページを参照してください。

例 2–1 フロッピーディスクをフォーマットする

次の例は、フロッピーディスクのフォーマット方法を示しています。

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

▼ リムーバブルメディア上にファイルシステムを作成する方法

1 (省略可能) フロッピーディスクをフォーマットします。

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

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

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

スライスファイルの例は次のようになります。

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

3 スーパーユーザーになります。

4 適切なファイルシステムの種類を決定し、次のいずれかの作業を選択します。

- UFS ファイルシステムを作成します。次に例を示します。

  ```
  # newfs /vol/dev/aliases/floppy0
  ```

- PCFS ファイルシステムを作成します。次に例を示します。

  ```
  # mkfs -F pcfs /dev/rdsk/c0t4d0s2:c
  ```

- UDFS ファイルシステムを作成します。次に例を示します。

```
# mkfs -F udfs /dev/rdsk/c0t1d0s1
```

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

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

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

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

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

次の例は、代替 `fdisk` パーティションを作成して、PCFS ファイルシステムを作成する方法を示しています。これらの例では、`vold` は動作していません。

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

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

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

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

▼ DVD-RAM 上にファイルシステムを作成する方法

現在のところ、vold は DVD-RAM デバイスをサポートしていません。このため、vold を無効にして DVD-RAM デバイスを使用すると、vold が無効で使用できないために、CD-R、CD-RW、DVD-R、DVD-RW、DVD+R、DVD+RW デバイスを使用できなくなります。

1. スーパーユーザーになります。
2. vold を停止します。

   ```
   # svcadm disable volfs
   ```

3. **DVD-RAM** デバイス上にファイルシステムを作成します。
   - UFS ファイルシステムを作成します。次に例を示します。

     ```
     # newfs /dev/rdsk/c0t0d0s2
     ```
   - UDFS ファイルシステムを作成します。次に例を示します。

     ```
     # mkfs -F udfs /dev/rdsk/c0t0d0s2
     ```
4. ファイルシステムをマウントします。
   - UFS ファイルシステムをマウントします。次に例を示します。

     ```
     # mount -F ufs /dev/dsk/c0t0d0s2 /mnt
     ```
   - UDFS ファイルシステムをマウントします。次に例を示します。

     ```
     # mount -F udfs /dev/dsk/c0t0d0s2 /mnt
     ```
5. ファイルシステムの読み取りや書き込みができることを確認します。
6. 作業が完了したら、**DVD-RAM** を取り出します。
7. vold を再起動します。

   ```
   # svcadm enable volfs
   ```

▼ リムーバブルメディア上のファイルシステムを検査する方法

1. スーパーユーザーになります。
2. ファイルシステムの種類を確認し、次のいずれかの作業を選択します。
   - UFS ファイルシステムを検査します。

     # fsck -F ufs *device-name*
   - UDFS ファイルシステムを検査します。

     # fsck -F udfs *device-name*
   - PCFS ファイルシステムを検査します。

     # fsck -F pcfs *device-name*

例 2–4 リムーバブルメディア上の PCFS ファイルシステムを検査する

次の例は、メディア上の PCFS ファイルシステムの整合性を検査する方法を示しています。この例では、`vold` は動作していません。

```
# fsck -F pcfs /dev/rdsk/c0t4d0s2
** /dev/rdsk/c0t4d0s2
** Scanning file system meta-data
** Correcting any meta-data discrepancies
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

▼ リムーバブルメディア上の不良ブロックを修復する方法

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

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

1 リムーバブルメディア上の不良ブロックを修復します。

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

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

2 リムーバブルメディアを検証します。

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

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

この機能をサポートしているリムーバブルメディアには、読み取り保護または書き込み保護を適用し、パスワードを設定できます。

▼ リムーバブルメディアの書き込み保護を有効または無効にする方法

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

- 書き込み保護を有効にします。

  ```
  $ rmformat -w enable device-name
  ```
- 書き込み保護を無効にします。

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

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

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

▼ リムーバブルメディアの読み取り/書き込み保護を有効または無効にする方法とパスワードを設定する方法

パスワードによる保護機能をサポートしているリムーバブルメディアには、最大 32 文字のパスワードを適用できます。

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

1. 読み取り保護または書き込み保護を有効または無効のどちらにするかを決定し、パスワードを設定します。次のいずれかの手順に従います。

   - 読み取り保護または書き込み保護を有効にします。

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

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

   - 読み取り保護または書き込み保護を無効にし、パスワードを削除します。

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

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

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

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

例 2–5 読み取り/書き込み保護とパスワードによる保護を有効または無効にする

次の例は、フロッピーディスクに対して書き込み保護を有効にし、パスワードを設定する方法を示しています。

```
$ rmformat -W enable /vol/dev/aliases/floppy0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

次の例は、フロッピーディスクに対して読み取り保護を無効にし、パスワードを削除する方法を示しています。

```
$ rmformat -R disable /vol/dev/aliases/floppy0
Please enter password (32 chars maximum): xxx
```

◆ ◆ ◆ 第 3 章

3 リムーバブルメディアへのアクセス (手順)

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

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

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

リムーバブルメディアの概要については、第 1 章「リムーバブルメディアの管理 (概要)」を参照してください。

リムーバブルメディアへのアクセス (作業マップ)

次の作業マップでは、リムーバブルメディアにアクセスするための作業について説明します。

作業	説明	参照先
1. (省略可能) リムーバブルメディアドライブを追加します。	必要に応じて、リムーバブルメディアドライブをシステムに追加します。	47 ページの「新しいリムーバブルメディアドライブを追加する方法」
2. (省略可能) リムーバブルメディアにアクセスするときにボリューム管理 (vold) を使用するかどうかを決めます。	デフォルトでは、ボリューム管理 (vold) は実行されます。リムーバブルメディアにアクセスするときにボリューム管理を使用するかどうかを決めます。	48 ページの「リムーバブルメディアサービスを無効または有効にする方法」

作業	説明	参照先
3. リムーバブルメディアにアクセスします。	ボリューム管理を使用して/使用しないで、各種のリムーバブルメディアにアクセスします。	49 ページの「リムーバブルメディア上の情報にアクセスする方法」
4. (省略可能) ファイルまたはディレクトリをコピーします。	ファイルまたはディレクトリを、ファイルシステムの別の場所からコピーするように、リムーバブルメディアからコピーします。	49 ページの「リムーバブルメディア上の情報をコピーする方法」
5. メディアが使用中であるかどうかを調べます。	メディアを取り出す前に、それが使用中であるかどうかを調べます。	50 ページの「リムーバブルメディアが使用中かどうかを調べる方法」
6. メディアを取り出します。	作業が終了したら、ドライブからメディアを取り出します。	51 ページの「リムーバブルメディアを取り出す方法」

リムーバブルメディアへのアクセス

リムーバブルメディア上の情報にアクセスするには、ボリューム管理を使用する方法と使用しない方法があります。GNOME のファイルマネージャーを使用してリムーバブルメディア上の情報にアクセスする方法については、GNOME デスクトップのマニュアルを参照してください。

ボリューム管理 (`vold`) を使用すると、すべてのリムーバブルメディアデバイスを動的に管理できます。このため、`/dev/rdsk/c`*n*`t`*n*`d`*n*`s`*n* や `/dev/dsk/c`*n*`t`*n*`d`*n*`s`*n* などのデバイス名でのリムーバブルメディアへのアクセスはできません。

リムーバブルメディア名の使用

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

表 3–1 リムーバブルメディア名

メディア	ボリューム管理デバイスの名前	ボリューム管理デバイスの別名	デバイス名
最初のフロッピーディスクドライブ	`/floppy`	`/vol/dev/aliases/floppy0`	`/dev/rdiskette` `/vol/dev/rdiskette0/`*volume-name*

表 3–1 リムーバブルメディア名 (続き)

メディア	ボリューム管理デバイスの名前	ボリューム管理デバイスの別名	デバイス名

最初、2 番目、3 番目の CD-ROM または DVD-ROM ドライブ	`/cdrom0`
	`/cdrom1`
	`/cdrom2`
		`/vol/dev/aliases/cdrom0`
		`/vol/dev/aliases/cdrom1`
		`/vol/dev/aliases/cdrom2`
			`/vol/dev/rdsk/c`*n*`t` *n*`[d`*n*`] /`
			volume-name

USB メモリースティック	`/rmdisk/noname`	`/vol/dev/aliases/rmdisk0`	`/vol/dev/dsk/c`*n*`t` *n*`d`*n*`/`*volume-name*`:c`

リムーバブルメディア上のデータにアクセスするためのガイドライン

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

たとえば、SPARC システム用にフォーマットされた UFS ファイルシステムを持つ CD と DVD は x86 システムでは認識されません。同様に、x86 システム用にフォーマットされた UFS CD は、ボリューム管理によって SPARC システム上にマウントすることはできません。一般に、フロッピーディスクにも同じ制限があてはまります。ただし、同じビット構造を共有するアーキテクチャーもあります。このため、場合によっては、あるアーキテクチャーに固有の UFS フォーマットが別のアーキテクチャーによって認識されることもあります。しかし、UFS ファイルシステム構造はこのような互換性を保証するように設計されたものではありません。

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

▼ 新しいリムーバブルメディアドライブを追加する方法

一般に、最近のバスタイプのほとんどで、ホットプラグ機能がサポートされています。システムのバスタイプでホットプラグ機能がサポートされている場合は、次の手順 5 だけを行う必要があります。システムのバスタイプでホットプラグ機能がサポートされていない場合は、次の作業を手順 1 - 6 に従って行う必要があります。

- `/reconfigure` ファイルを作成します。
- システムをリブートして、ボリューム管理に新しいメディアドライブを認識させます。

ホットプラグ対応デバイスの詳細は、第 6 章「デバイスの動的構成 (手順)」を参照してください。

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

   ```
   # touch /reconfigure
   ```

3. システムを実行レベル **0** にします。

   ```
   # init 0
   ```

4. システムの電源を切ります。
5. 新しいリムーバブルメディアドライブを接続します。

   詳細な手順については、該当するハードウェアのマニュアルを参照してください。
6. システムの電源を入れます。

   システムが自動的にマルチユーザーモードになります。

▼ リムーバブルメディアサービスを無効または有効にする方法

場合によっては、リムーバブルメディアサービスを使用しないで、メディアを管理した方がよいことがあります。この節では、リムーバブルメディアサービスを無効または有効にする方法について説明します。

これらのサービスを無効にした場合は、`mount` コマンドを使ってすべてのメディアを手動でマウントする必要があります。

1. メディアが使用中でないことを確認します。

   メディアを使用中のすべてのユーザーを確認できた保証がない場合は、50 ページの「リムーバブルメディアが使用中かどうかを調べる方法」の手順に従って `fuser` コマンドを使用してください。
2. スーパーユーザーになります。
3. 次のいずれかの手順に従います。
   - リムーバブルメディアサービスを無効にします。

```
# svcadm disable volfs
```

- リムーバブルメディアサービスを有効にします。

```
# svcadm enable volfs
volume management starting.
```

▼ リムーバブルメディア上の情報にアクセスする方法

1. メディアを挿入します。
   数秒後にメディアがマウントされます。
2. メディアの内容をリスト表示します。

   % **ls** */media*

   コマンド行インタフェースを使用し、適切なデバイス名を指定して情報にアクセスします。デバイス名については、表 3–1 を参照してください。

例 3–1 リムーバブルメディア上の情報にアクセスする

次の例は、フロッピーディスク上の情報にアクセスする方法を示しています。

```
$ ls /floppy
myfile
```

次の例は、USB メモリースティック上の情報にアクセスする方法を示しています。

```
$ ls /rmdisk
rmdisk0/        rmdisk1/
```

次の例は、DVD 上の情報にアクセスする方法を示しています。

```
$ ls /cdrom
cdrom0              sol_10_910_sparc
```

▼ リムーバブルメディア上の情報をコピーする方法

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

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

1. メディアがマウントされていることを確認します。

   ```
   $ ls /media
   ```

   ls コマンドは、マウントされたメディアの内容を表示します。内容が表示されない場合は、49 ページの「リムーバブルメディア上の情報にアクセスする方法」を参照してください。

2. (省略可能) ファイルまたはディレクトリをコピーします。

   たとえば、DVD の場合は次のように入力します。

   ```
   $ cp /cdrom/sol_10_910_sparc/Solaris_10/Tools/add_install_client .
   $ ls -l
   -rwxr-xr-x   1 pmorph      gelfs   64065 Jul 12  2010 add_install_client
   ```

▼ リムーバブルメディアが使用中かどうかを調べる方法

1. スーパーユーザーになります。

2. メディアにアクセスしているプロセスを特定します。

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

   -u は、メディアを使用しているユーザーなどを表示します。

   詳細は、fuser(1M) のマニュアルページを参照してください。

3. (省略可能) メディアにアクセスしているプロセスを強制終了します。

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

   -k は、メディアにアクセスしているプロセスを強制終了します。

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

4. プロセスが終了していることを確認します。

   ```
   # pgrep process-ID
   ```

例 3–2 メディアが使用中かどうかを調べる

次の例は、ユーザー pmorph が /cdrom/sol_10_910_sparc/Solaris_10/Tools ディレクトリにアクセスしていることを示しています。

```
# fuser -u /cdrom/sol_10_910_sparc/Solaris_10/Tools
/cdrom/sol_910_sparc/Solaris_10/Tools:      902c(pmorph)      339c(pmorph)
```

▼ リムーバブルメディアを取り出す方法

1 メディアが使用中でないことを確認します。

シェルまたはアプリケーションがメディア上のファイルまたはディレクトリのいずれかにアクセスしている場合、メディアは「使用中」であることを忘れないでください。DVD を使用しているシェルやアプリケーションなどをすべて検出したかどうかわからない (デスクトップツールの背後に隠れているシェルがアクセスしている可能性がある) 場合は、`fuser` コマンドを使用してください。50 ページの「リムーバブルメディアが使用中かどうかを調べる方法」を参照してください。

2 メディアを取り出します。

```
# eject media
```

たとえば、DVD の場合は次のように入力します。

```
# eject cdrom
```

たとえば、USB メモリースティックの場合は次のように入力します。

```
# eject rmdisk0
```

ヒント – リムーバブルデバイスの名前は `eject -l` コマンドで表示できます。

リモートシステム上のリムーバブルメディアへのアクセス (作業マップ)

次の作業マップでは、リモートシステム上のリムーバブルメディアにアクセスするための作業について説明します。

作業	説明	参照先
1. ローカルのメディアをリモートシステムで使用できるようにします。	システムのメディアドライブを共有するようにシステムを設定して、そのドライブ内のメディアをほかのシステムでも使用できるようにします。	52 ページの「ローカルのメディアをほかのシステムで使用可能にする方法」
2. リモートシステム上のリムーバブルメディアにアクセスします。	ローカルシステムでリモートメディアにアクセスします。	49 ページの「リムーバブルメディア上の情報にアクセスする方法」

▼ ローカルのメディアをほかのシステムで使用可能にする方法

システムのメディアドライブを共有するようにシステムを設定すると、そのドライブに読み込まれているメディアがほかのシステムでも使用できるようになります。ただし、音楽用 CD は例外です。メディアドライブを共有すると、そのドライブをマウントするだけで、ドライブにロードされているメディアをほかのシステムでも使用できます。手順については、53 ページの「リモートシステム上のリムーバブルメディアにアクセスする方法」を参照してください。

1. スーパーユーザーになります。
2. メディアがロードされていることを確認します。
3. 次のエントリを /etc/dfs/dfstab ファイルに追加します。

   次に例を示します。

   ```
   share -F nfs -o ro /cdrom/sol_10_910_sparc
   ```

4. **NFS サーバーサービスが実行中かどうかを確認します。**

   ```
   # svcs *nfs*
   ```

   NFS サーバーサービスが実行中であれば、svcs コマンドの出力は次のようになります。

   ```
   online         14:28:43 svc:/network/nfs/server:default
   ```

5. **NFS サーバーの状態を確認し、次のいずれかの手順を選択します。**
   - NFS サーバーサービスが実行されている場合は、手順 7 に進みます。
   - NFS サーバーサービスが実行されていない場合は、次の手順に進みます。
6. **NFS サーバーサービスを起動します。**

   ```
   # svcadm enable network/nfs/server
   ```

   NFS デーモンが実行されていることを確認します。

   次に例を示します。

   ```
   # svcs -p svc:/network/nfs/server:default
   STATE          STIME    FMRI
   online         Aug_30   svc:/network/nfs/server:default
                  Aug_30        319 mountd
                  Aug_30        323 nfsd
   ```

7 メディアが実際にほかのシステムで使用できるかどうかを確認します。

メディアが使用可能な場合は、その共有設定が表示されます。

```
# share
-               /cdrom/sol_10_910_sparc   ro   ""
```

例 3–3 ローカルの DVD または CD をほかのシステムで使用可能にする

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

```
# vi /etc/dfs/dfstab
(Add the following line:)
# share -F nfs -o ro /media
# svcs *nfs*
# svcadm enable network/nfs/server
# svcs -p svc:/network/nfs/server:default
# share
-               /cdrom/sol_10_910_sparc   ro   ""
```

例 3–4 ローカルのフロッピーディスクをほかのシステムで使用可能にする

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

```
# vi /etc/dfs/dfstab
(Add the following line, for example)
share -F nfs -o ro /floppy/myfiles
# svcs *nfs*
# svcadm enable network/nfs/server
# svcs -p svc:/network/nfs/server:default
# volcheck -v
media was found
# share
-               /floppy/myfiles   rw   ""
```

▼ リモートシステム上のリムーバブルメディアにアクセスする方法

リモートシステム上のメディアを手動でファイルシステムにマウントすることにより、そのメディアにアクセスできるようになります。ただし、リモートシステムが52 ページの「ローカルのメディアをほかのシステムで使用可能にする方法」の手順に従って、そのメディアを共有していることが必要です。

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します。または、マウントポイントを作成します。

```
$ mkdir /directory
```

ここで */directory* は、リモートシステムの DVD のマウントポイントとして作成するディレクトリの名前です。

2 マウントするメディアの名前を確認します。

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

3 スーパーユーザー権限で、メディアをマウントします。

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

system-name: マウントするメディアを持つシステムの名前です。

media-name マウントするメディアの名前です。

local-mount-point リモートメディアのマウント先となるローカルディレクトリです。

4 スーパーユーザーをログアウトします。

5 メディアがマウントされていることを確認します。

```
$ ls /media
```

例 3–5 リモートシステム上の DVD または CD にアクセスする

次の例は、リモートシステム starbug の sol_10_910_sparc という名前のリモート DVD に、autofs を使って自動的にアクセスする方法を示しています。

```
% showmount -e starbug
export list for starbug:
/cdrom/sol_10_910_sparc (everyone)
$ /net/starbug/cdrom/sol_10_910_sparc
```

例 3–6 ほかのシステム上のフロッピーディスクにアクセスする

次の例は、リモートシステム mars にある myfiles というフロッピーディスクに、autofs を使用して自動的にアクセスする方法を示しています。

```
$ showmount -e mars
$ cd /net/mars
$ ls /floppy
floppy0     myfiles
```

◆ ◆ ◆ 第 4 章

CD および DVD への書き込み (手順)

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

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

オーディオ CD、データ CD、データ DVD の取り扱い

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

cdrw コマンドを使用して、次の作業を実行できます。

- データ CD およびデータ DVD を作成します。
- オーディオ CD を作成します。
- オーディオ CD からオーディオデータを抽出します。
- CD および DVD をコピーします。
- CD-RW メディアを消去します。

cdrw コマンドは、次のリリース以降で利用可能です。

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

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

推奨される CD-R または CD-RW デバイスについては、次の Web サイトを参照してください。

http://www.sun.com/io_technologies/ihvindex.html

CD/DVD メディアに関するよく使われる用語

ここでは、CD/DVD メディアに関連してよく使われる用語を定義します。

用語	説明
CD-R	CD 読み取りメディア。1 度だけ書き込みができ、その後は読み取り専用となります。
CD-RW	書き換え可能な CD メディア。書き込みと消去が可能です。CD-RW デバイスだけが CD-RW メディアを読み取れます。
DVD-R	デジタルビデオディスク (記録可能)。1 度だけ書き込みができ、その後は読み取り専用となります。このデバイスは、CD-R メディアよりも大容量です。
DVD+R	デジタルビデオディスク (記録可能)。1 度だけ書き込みができ、その後は読み取り専用となります。DVD+R デバイスは、DVD-R よりも総合的なエラー管理システムを備えているため、メディアの品質に関係なくより正確な書き込みを行えます。
DVD-RW	DVD-R と同じ記憶容量のデジタルビデオディスク (書き換え可能)。最初にディスク全体を消去したあとで再度記録できます。
DVD+RW	DVD+R と同じ記憶容量のデジタルビデオディスク (ランダムアクセス書き換え可能)。ディスク全体を消去せずに個々のブロックを上書きできます。
DVD-RAM	デジタルビデオディスク (ランダムアクセスメモリー、書き換え可能)。このメディアでは、トラックおよびハードセクタがらせん状ではなく、円状になっています。

用語	説明
ISO 9660	ISO (Industry Standards Organization の略)。コンピュータの標準記憶フォーマットを設定する組織です。 ISO 9660 ファイルシステムは、CD や DVD の標準ファイルシステムであり、主なコンピュータプラットフォームで同じ CD や DVD を読み取れます。この標準は 1988 年に発行され、High Sierra (ネバダ州の High Sierra Hotel にちなんで名づけられた) という業界団体によって作成されました。CD ドライブや DVD ドライブを備えたほぼすべてのコンピュータが ISO 9660 ファイルシステムからファイルを読み取れます。
Joliet 拡張	Windows ファイルシステム情報を追加します。
Rock Ridge 拡張	UNIX ファイルシステム情報を追加します。(Rock Ridge は映画「Blazing Saddles」に出てくる町にちなんで名づけられました。) 注 – これらの拡張は排他的ではありません。したがって、`mkisofs` コマンドに `-R` と `-j` の両オプションを指定して両方のシステムとの互換性を確保できます。(詳細は mkisofs(1M) のマニュアルページを参照してください。)
MMC 準拠のレコーダ	Multi Media Command の略。これらのレコーダが共通のコマンドセットに準拠していること意味します。ある MMC 準拠レコーダに書き込めるプログラムは、ほかのすべての MMC 準拠レコーダにも書き込むことができます。
Red Book CDDA	Compact Disc Digital Audio の略。デジタルオーディオをコンパクトディスクに格納するための業界標準方式です。「Red Book」形式とも呼ばれます。この公式の業界仕様では、1 つまたは複数のオーディオファイルが 44.1 kHz のサンプリングレートで 16 ビットのステレオサウンドにサンプリングされることが要求されます。

CD メディアへの書き込みでよく使われる用語は、次のとおりです。

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

データ CD、データ DVD、オーディオ CD への書き込み

CD や DVD への書き込み処理は、途中で中断されることなく、データストリームが一定に保たれている必要があります。CD または DVD に十分な速度でデータを書き込めることを確認するために、`cdrw -S` オプションを使ってメディアへの書き込みをシミュレートすることを検討してください。

次のような問題がある場合には、書き込みエラーが発生することがあります。

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

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

たとえば、次のコマンドは、4x の速度での書き込みをシミュレートします。

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

注 – CD-R、CD-RW (MRW フォーマット以外)、DVD-R、および DVD-RW メディアでは、シミュレーションモード (`-S`) がサポートされます が、DVD-RAM、DVD+R、DVD+RW、MRW フォーマットされたメディア、およびその他一部のメディアではシミュレーションモードはサポートされません。シミュレーションモードがサポートされない場合、次のメッセージが表示されます。

```
Media does not support simulated writing
```

メディアタイプの詳細は、56 ページの「CD/DVD メディアに関するよく使われる用語」を参照してください。

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

詳細は、`cdrw(1)` のマニュアルページを参照してください。

RBAC を使用してリムーバブルメディアへのユーザーアクセスを制限する

デフォルトでは、すべてのユーザーが Solaris 9 リリースで起動するリムーバブルメディアを利用できます。ただし、役割によるアクセス制御 (RBAC) で役割を設定して、リムーバブルメディアへのユーザーアクセスを制限できます。リムーバブルメディアへのアクセスを制限するには、その役割を一部のユーザーだけに割り当てます。

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

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

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

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

   コンソールの起動方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。
3. デバイス管理の権利が含まれている役割を設定します。

   詳細は、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。
4. cdrw コマンドを使用する必要があるユーザーを、新しく作成した役割に追加します。
5. **/etc/security/policy.conf** ファイルの次の行をコメントにします。

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

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

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

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

▼ CD または DVD ライターを確認する方法

1 システム上の **CD** または **DVD** ライターを確認します。
次に例を示します。

```
$ cdrw -l
Looking for CD devices...
    Node               |    Connected Device            |  Device type
-----------------------+--------------------------------+----------------
 cdrom0                | YAMAHA   CRW8824S         1.0d | CD Reader/Writer
```

2 特定の **CD** または **DVD** ライターを確認します。
次に例を示します。

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

3 メディアが空であるか、または既存の目次があるかどうかを確認します。
次に例を示します。

```
$ cdrw -M

Device : YAMAHA   CRW8824S
Firmware : Rev. 1.00 (26/04/00)
Media is blank
%
```

▼ CD または DVD メディアを検査する方法

cdrw コマンドは、リムーバブルメディアサービスが動作している状態でも動作していない状態でも機能します。リムーバブルメディアサービスを無効または有効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。

1 **CD** または **DVD** をドライブに挿入します。
そのドライブで読み取ることができる CD または DVD であれば、どのような種類でもかまいません。

2 デバイスのリストを表示して、そのドライブが正しく接続されていることを確認します。

```
$ cdrw -l
 Looking for CD devices...
    Node                  Connected Device                Device type
-----------------------+--------------------------------+----------------
 cdrom1                | YAMAHA   CRW8824S         1.0d | CD Reader/Writer
```

3 (省略可能) そのドライブがリスト内にない場合、システムにそのデバイスを認識させるために、次のいずれかの作業を選択します。

- 再構成用ブートを実行します。

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

- システムをリブートせずにドライブを追加します。

```
# drvconfig
# disks
```

次に、リムーバブルメディアサービスを再起動します。

```
# svcadm disable volfs
# svcadm enable volfs
```

データ CD またはデータ DVD を作成する

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

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

1. 空の **CD** または **DVD** をドライブに挿入します。
2. その新しい **CD** または **DVD** 上に **ISO 9660** ファイルシステムを作成します。

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

`-r`	Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットします。
/pathname	ISO 9660 ファイルシステムの作成に使われるパス名を指定します。
`>` *cd-file-system*	CD または DVD に書き込むファイルシステムの名前を指定します。

3. ファイルシステムを **CD** または **DVD** にコピーします。

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

   `-i` *cd-file-system* は、CD または DVD を作成するためのイメージファイルを指定します。

例 4–1 データ CD またはデータ DVD 用の ISO 9660 ファイルシステムを作成する

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

```
$ mkisofs -r /home/dubs/ufs_dir > ufs_cd
Total extents actually written = 56
Total translation table size: 0
```

```
Total rockridge attributes bytes: 329
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 8000
56 extents written (0 Mb)
```

次に、ファイルシステムを CD または DVD にコピーします。

```
$ cdrw -i ufs_cd
Initializing device...done.
Writing track 1...done.
Finalizing (Can take several minutes)...done.
```

▼ マルチセッションのデータ CD を作成する方法

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

1. 最初の **CD** セッション用のファイルシステムを作成します。

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

`-o infoA`	ISO ファイルシステムの名前を指定します。
`-r`	Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットします。
`-V my_infoA`	リムーバブルメディアサービスがマウントポイントとして使用するボリュームラベルを指定します。
`/data/infoA`	作成する ISO イメージディレクトリを指定します。

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

```
$ cdrw -iO infoA
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

`-i` *infoA*	CD に書き込むイメージファイルの名前を指定します。
`-O`	書き込むために CD を開いたままにしておきます。

3. **CD** が排出されたあとで、再度挿入します。

4 次の書き込みセッションに含める **CD** メディアのパス名を確認します。

```
$ eject -n
.
.
.
cdrom0 -> /vol/dev/rdsk/c2t4d0/my_infoA
```

`/vol/dev/...` パス名を書き留めてください。

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

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

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

Last session start address: 75287
Next writable address: 91118
```

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

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

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

`-o` *infoB*	ISO ファイルシステムの名前を指定します。
`-r`	Rock Ridge 情報を作成し、ファイル所有権を 0 にリセットします。
`-C` *0,91118*	最初のセッションの開始アドレスと、次に書き込み可能なアドレスを指定します。
`-M /vol/dev/rdsk/c2t4d0/my_infoA`	マージする既存の ISO イメージのパスを示します。
`/data/infoB`	作成する ISO イメージディレクトリを指定します。

オーディオ CD を作成する

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

次の表に、サポートされているオーディオ形式を示します。

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

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

▼ オーディオ CD を作成する方法

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

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

   ```
   $ cd /myaudiodir
   ```

3. オーディオファイルを **CD** にコピーします。

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

   -a オプションによってオーディオ CD が作成されます。

例 4–2 オーディオ CD を作成する

次の例は、オーディオ CD を作成する方法を示しています。

```
$ cdrw -a bark.wav chirp.au meow.wav
Initializing device...done.
Writing track 1...done.
done.
```

```
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
```

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

```
$ cdrw -aO groucho.wav chico.au harpo.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
<Re-insert CD>
$ cdrw -a zeppo.au
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

▼ オーディオトラックを CD から抽出する方法

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

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

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

1. オーディオ CD を CD-RW ドライブに挿入します。
2. オーディオトラックを抽出します。

   $ **cdrw -x -T** *audio-type* **1** *audio-file*

   -x — オーディオ CD からオーディオデータを抽出します。

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

   audio-file — 抽出されるオーディオトラックを指定します。
3. 抽出したトラックを新しい CD にコピーします。

   $ **cdrw -a** *audio-file*

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

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

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

次の例は、オーディオ CD にトラックをコピーする方法を示しています。

```
$ cdrw -a song1.wav
Initializing device...done.
Writing track 1...done.
Finalizing (Can take several minutes)...done.
```

▼ CD をコピーする方法

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

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

---

1. オーディオ **CD** を **CD-RW** ドライブに挿入します。
2. オーディオファイル用のディレクトリを作成します。

   ```
   $ mkdir /music_dir
   ```

3. オーディオ **CD** からトラックを抽出します。

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

   トラックごとに Extracting audio ... メッセージが表示されます。

   すべてのトラックが抽出されると、CD が排出されます。
4. 空の **CD** を挿入して、**Return** キーを押します。

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

例 4–4 CD をコピーする

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

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

▼ CD-RW メディアを消去する方法

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

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

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

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

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

◆ ◆ ◆ 第 5 章

デバイスの管理 (概要と手順)

この章では、Solaris リリースのディスク、DVD ドライブ、テープデバイスなどの周辺デバイスを管理する方法について説明します。

この章の内容は以下のとおりです。

- 69 ページの「デバイス管理の新機能」
- 73 ページの「デバイス管理作業についての参照先」
- 75 ページの「デバイスドライバについて」
- 75 ページの「デバイスの自動構成」
- 77 ページの「デバイス構成情報の表示」
- 83 ページの「デバイスへのアクセス」

この章で説明する手順は次のとおりです。

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

Solaris リリース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他社製のデバイスドライバの追加、システム構成情報の表示が含まれます。

デバイス管理の新機能

ここでは、Solaris リリースにおけるデバイス管理の新機能について説明します。

- 70 ページの「新しいホットプラグ機能」
- 70 ページの「x86: デバイス検出ツール」
- 70 ページの「PCI Express (PCIe) のサポート」
- 71 ページの「USB と 1394 (FireWire) サポートの機能拡張」
- 71 ページの「使用中デバイスのエラーチェック機能の向上」

Solaris の新機能の一覧および Solaris リリースについての説明は、『Oracle Solaris 10 9/10 の新機能』を参照してください。

新しいホットプラグ機能

Oracle Solaris 10 9/10: この Solaris リリースでは、`hotplug` コマンドを使って、PCI Express (PCIe) および PCI SHPC (Standard Hot Plug Controller) デバイス上でホットプラグを実行可能な接続を管理できるようになりました。この機能は、USB や SCSI などの、ほかの種類のバスではサポートされません。

ホットプラグを実行可能な USB および SCSI デバイスの管理には、以前の Solaris リリースと同様に、引き続き `cfgadm` を使用します。このリリースでホットプラグ機能を使用する利点は、有効化と無効化の操作に加え、`hotplug` コマンドを使って対応する PCI デバイスをオフラインおよびオンラインにできることです。

詳細については、93 ページの「`hotplug` コマンドによる PCIe ホットプラグ」を参照してください。

x86: デバイス検出ツール

Solaris 10 5/08: デバイス検出ツールを使用すると、使用している x86 ハードウェアがこの Solaris リリースでサポートされているかどうかを確認できます。詳細は、次のサイトを参照してください。

`http://www.sun.com/bigadmin/hcl/hcts/device_detect.jsp`

PCI Express (PCIe) のサポート

Solaris 10 11/06: この Solaris リリースでは、PCI Express (PCIe) インターコネクトをサポートします。PCI Express (PCIe) インターコネクトは、SPARC システムと x86 システムの両方で、周辺デバイスをデスクトップ、エンタープライズ、モバイル、通信、組み込みの各アプリケーションに接続するために設計されています。

以前の Solaris 10 6/06 リリースでは、PCIe デバイスは x86 システムでのみ使用可能でした。

PCIe インターコネクトは、業界標準の高性能シリアル入出力バスです。PCIe 技術の詳細は次のサイトを参照してください。

`http://www.pcisig.com/home`

PCIe ソフトウェアは、この Solaris リリースで次の機能を提供します。

- 拡張された PCIe 構成スペースのサポート

- PCIe ベースラインエラー処理と MSI 割り込みのサポート
- PCIe デバイスに合わせた IEEE-1275 プロパティーの変更
- `cfgadm` コマンドの `cfgadm_pci` コンポーネントの拡張による PCIe ホットプラグ (ネイティブおよび ACPI ベース) のサポート
- Attention ボタンの使用に基づく PCIe 周辺デバイスの自動構成

ホットプラグ対応 PCIe 周辺デバイスは、PCI 周辺デバイスと同じように `cfgadm` コマンドを使用して管理します。

使用しているハードウェアマニュアルを参照して、使用しているシステム上で PCIe および PCIe のホットプラグがサポートされているかどうかを確認します。また、アダプタを物理的にシステムに挿入する手順とシステムから取り外す手順、および該当する場合のデバイスの自動構成のセマンティクスについてよく確認してください。

PCIe 周辺デバイスでの `cfgadm` コマンドの使用方法については、105 ページの「`cfgadm` コマンドによる PCI または PCIe ホットプラグ (作業マップ)」を参照してください。

USB と 1394 (FireWire) サポートの機能拡張

Solaris 10 6/06: この Solaris リリースでは、非リムーバブル USB ストレージデバイスと 1394 大容量デバイスの両方がドライバレベルでホットプラグ対応デバイスとして識別されます。この新しい動作は、システムをリブートせずにこれらのデバイスを接続または取り外しできること、およびユーザーの介入なしで自動的に構成または構成解除できることを意味します。これらの変更は、カーネルレベルで行われているため、デバイスの使用には影響しません。たとえば、これらのデバイスのマウントおよびマウント解除は、リムーバブルメディア管理サービスによって制御されます。

また、 `format` ユーティリティーを使用して、非リムーバブル USB デバイスおよび 1394 大容量ストレージデバイスにアクセスしてラベルを付けることができます。ただし、`/kernel/drv/scsa2usb.conf` ファイル内の `remvalue` を true に設定することで、これらのデバイスの新しいホットプラグ機能を無効にすることができます。このパラメータを true に設定すると、この動作を優先する場合は、デバイスがドライバレベルでリムーバブルメディアとして扱われます。

これらのデバイスの使用方法については、`scsa1394(7D)` のマニュアルページと140 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」を参照してください。

使用中デバイスのエラーチェック機能の向上

Solaris 10 6/06: この機能は、以前はマニュアルに書かれていませんでした。

次のユーティリティーが拡張されて、指定されたデバイスが使用中かどうかを検出するようになりました。

- dumpadm
- format
- mkfs と newfs
- swap

これらの拡張により、上記のユーティリティーが、次のような使用状況のシナリオをいくつか検出する可能性があります。

- デバイスが ZFS ストレージプールに含まれている
- デバイスがダンプデバイスまたはスワップデバイスである
- マウントされたファイルシステムまたはデバイスのエントリが /etc/vfstab ファイルに存在する
- デバイスが Live Upgrade 構成の一部である
- デバイスが Solaris ボリュームマネージャーの構成または Veritas ボリュームマネージャーの構成に含まれている

たとえば、format ユーティリティーを使用してアクティブデバイスにアクセスしようとすると、次のようなメッセージが表示されます。

```
# format
.
.
.
Specify disk (enter its number): 1
selecting c0t1d0
[disk formatted]
Warning: Current Disk has mounted partitions.
/dev/dsk/c0t1d0s0 is currently mounted on /. Please see umount(1M).
/dev/dsk/c0t1d0s1 is currently used by swap. Please see swap(1M).
```

ただし、これらのユーティリティーは、すべてのケースを同様に検出するわけではありません。たとえば、newfs コマンドを使用して、Live Upgrade の構成に含まれているデバイス上に新しいファイルシステムを作成できます。ただし、Live Upgrade の構成に含まれているデバイス上に、マウントされたファイルシステムも存在する場合は、newfs コマンドを使用して新しいファイルシステムを作成することはできません。

デバイス管理作業についての参照先

次の表に、デバイスのホットプラグの実行手順、およびシリアルデバイス (プリンタやモデムなど) や周辺デバイス (ディスク、DVD ドライブ、テープデバイスなど) の追加手順が説明されている参照先を示します。

表 5–1 デバイスを追加する場合の参照先

デバイス管理作業	参照先
ホットプラグ不可ディスクを追加します。	第 12 章「SPARC: ディスクの追加 (手順)」または 第 13 章「x86: ディスクの追加 (手順)」
SCSI または PCI デバイスのホットプラグを実行します。	95 ページの「cfgadm コマンドによる SCSI ホットプラグ」または 105 ページの「cfgadm コマンドによる PCI または PCIe ホットプラグ」
USB デバイスのホットプラグを実行します。	140 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」
DVD ドライブまたはテープドライブを追加します。	81 ページの「周辺デバイスを追加する方法」
モデムを追加します。	『Solaris のシステム管理 (上級編)』の第 1 章「端末とモデムの管理 (概要)」
プリンタを追加します。	『Solaris のシステム管理 (印刷)』
デバイスをセキュリティー保護します。	『Solaris のシステム管理 (セキュリティサービス)』の第 4 章「デバイスアクセスの制御 (作業)」

Solaris OS でのデバイス管理

以降の節では、Solaris OS でデバイスを管理する機能の概要を説明します。デバイスへのアクセスに関する情報については、83 ページの「デバイスへのアクセス」を参照してください。

デバイスの電力管理

米国環境保護局では、エネルギー効率のよいコンピュータシステムの使用を奨励し、エネルギー生成に伴う大気汚染を削減する目的で、コンピュータ製品の省電力 (Energy Star) ガイドラインを作成しました。これらのガイドラインを満たすために、Sun のハードウェアは電力を効率的に使用するよう設計されています。また、電力管理設定を構成するための電力管理ソフトウェアも用意されています。

システムの電力管理方法については、お使いのハードウェアのマニュアルまたは `power.conf(4)` のマニュアルページを参照してください。

ファイバチャネルデバイスの電力管理

Sun システムの電源管理は、これまで多くの Solaris リリースで提供されてきました。たとえば、次のシステム上の内部ドライブは、デフォルトで電力管理対象になります。

- SunBlade 1000 または 2000
- SunBlade 100 または 150
- SunBlade 2500 または 1500

`/etc/power.conf` ファイルのデフォルト設定を使用した場合、Energy Star 準拠が保証され、これらのシステムの電力管理が完全にサポートされます。

次のアダプタは、外部のファイバチャネルストレージデバイスを接続します。

- Sun StorEdge PCI Dual Fibre Channel Host Adapter
- Sun StorEdge PCI Single Fibre Channel Network Adapter

上記アダプタと Sun システムを組み合わせて外部ファイルチャネルストレージデバイスを接続する場合、それらの外部ストレージデバイスもデフォルトで電力管理対象になります。

次の場合は、電力管理を無効にする必要があります。

- SAN (Storage Area Network) に接続されたファイバチャネル接続ディスクがシステム上に存在している場合
- SunCluster ソフトウェアなどを使ってマルチイニシエータ構成で使用されるファイバチャネル接続ディスクがシステム上に存在している場合
- システムがファイバチャネルインタフェース経由で IP を使用している場合 (`fcip(7D)` のマニュアルページを参照)

上記の場合のように同一のデバイスを複数の Solaris システムが共有している可能性がある場合には、電力管理を有効にすることはできません。

システムの電力管理を無効にするには、`/etc/power.conf` ファイル内の `autopm` キーワードを次のように変更します。

```
autopm          disable
```

続いて、電力管理を再構成するために、`pmconfig` コマンドを実行するか、システムをリブートします。

詳細については、`power.conf(4)` と `pmconfig(1M)` のマニュアルページを参照してください。

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容量ストレージデバイスを使用します。たとえば、各システムには、SCSI ディスクドライブ、キーボードとマウス、磁気バックアップメディアなどがあります。これ以外に一般に使用されるデバイスには、次のようなものがあります。

- DVD 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
- デジタイザー
- タブレットとスタイラスのペア

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

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

デバイスの自動構成

カーネルは、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成ります。カーネルは Solaris リリースで自動的に構成されます。

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

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

カーネルモジュールについては、次の表で説明します。

表 5–2　Solaris カーネルモジュール

場所	内容
`/platform/‘uname -m‘/kernel`	プラットフォーム固有のカーネル構成要素
`/kernel`	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
`/usr/kernel`	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はじめてアクセスされると、対応するデバイスドライバがロードされます。このプロセスは、「自動構成」と呼ばれます。これは、すべてのカーネルモジュールが、必要に応じて自動的にロードされるためです。

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

自動構成の機能と利点

自動構成の利点は次のとおりです。

- モジュールが必要に応じてロードされるため、主メモリーをより効率的に使用できます。
- 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ありません。
- 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できます。

自動構成プロセスは、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。デバイスがホットプラグ対応でない場合は、システムの再構成ブートを実行して、システムに新しいデバイスを認識させる必要があります。ホットプラグ対応デバイスについては、第 6 章「デバイスの動的構成 (手順)」を参照してください

標準サポートされていないデバイスを使用する場合

Solaris リリースには、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、その製造元から、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ー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ーが必要になる場合があります。

サポートされていないデバイスに必要な対策については、デバイスのご購入先にお問い合わせください。

デバイス構成情報の表示

システムとデバイスの構成情報を表示するには、次の 3 つのコマンドを使用します。

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

システムのデバイスの識別に使用されるデバイス名については、84 ページの「デバイス名の命名規則」を参照してください。

driver not attached メッセージ

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

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

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

▼ システム構成情報を表示する方法

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

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

- システムおよびデバイスの構成情報を表示します。
  - システムに接続されているすべてのデバイスを表示します。

    たとえば、SunBlade 1000 に対する次の `prtconf -v` の出力は、システムに接続されているディスクデバイスを識別しています。ディスクの詳細情報は、`Device Minor Nodes` セクションの `ssd/fp` ドライバセクションに表示されます。

    ```
    $ /usr/sbin/prtconf -v | more
    .
    .
    .
                Device Minor Nodes:
                   dev=(118,8)
                      dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
    2037bde864,0:a
                          spectype=blk type=minor
                          dev_link=/dev/dsk/c0t1d0s0
                      dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
    2037bde864,0:a,raw
                          spectype=chr type=minor
                          dev_link=/dev/rdsk/c0t1d0s0
                   dev=(118,9)
                      dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
    2037bde864,0:b
                          spectype=blk type=minor
                          dev_link=/dev/dsk/c0t1d0s1
                      dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
    2037bde864,0:b,raw
    .
    .
    .
    ```
  - システムに接続されている特定のデバイスに関する情報を表示します。

    たとえば、SunBlade 1000 に対する次の `prtconf` の出力は、`/dev/dsk/c0t1d0s0` の `ssd` インスタンス番号を示しています。

    ```
    # prtconf -v /dev/dsk/c0t1d0s0
    ssd, instance #1
    ```
  - システムに接続されているデバイスだけを表示します。

    ```
    # prtconf | grep -v not
    ```
  - デバイスの使用状況を表示します。

    たとえば、次の `fuser` コマンドは、どのプロセスが `/dev/console` デバイスにアクセスしているかを示しています。

    ```
    # fuser -d /dev/console
    /dev/console:      346o     323o
    ```

例 5–1 システム構成情報の表示

SPARC システムでは、次の `prtconf` 出力が表示されます。

```
# prtconf
System Configuration:  Sun Microsystems  sun4u
```

```
Memory size: 512 Megabytes
System Peripherals (Software Nodes):

SUNW,Sun-Blade-1000
    scsi_vhci, instance #0
    packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)
        terminal-emulator (driver not attached)
        obp-tftp (driver not attached)
        dropins (driver not attached)
        kbd-translator (driver not attached)
        ufs-file-system (driver not attached)
    chosen (driver not attached)
    openprom (driver not attached)
        client-services (driver not attached)
    options, instance #0
    aliases (driver not attached)
    memory (driver not attached)
    virtual-memory (driver not attached)
    SUNW,UltraSPARC-III, instance #0
    memory-controller, instance #0
    SUNW,UltraSPARC-III, instance #1
    memory-controller, instance #1
    pci, instance #0
        ebus, instance #0
            flashprom (driver not attached)
            bbc (driver not attached)
            ppm, instance #0
            i2c, instance #0
                dimm-fru, instance #0
                dimm-fru, instance #1
                dimm-fru, instance #2
                dimm-fru, instance #3
                nvram, instance #4
                idprom (driver not attached)
            i2c, instance #1
                cpu-fru, instance #5
                temperature, instance #0
                cpu-fru, instance #6
                temperature, instance #1
                fan-control, instance #0
                motherboard-fru, instance #7
                i2c-bridge (driver not attached)
            beep, instance #0
            rtc, instance #0
            gpio (driver not attached)
            pmc (driver not attached)
            floppy (driver not attached)
            parallel (driver not attached)
            serial, instance #0
        network, instance #0
        firewire, instance #0
        usb, instance #0
        scsi (driver not attached)
            disk (driver not attached)
            tape (driver not attached)
        scsi (driver not attached)
```

```
            disk (driver not attached)
            tape (driver not attached)
    pci, instance #1
        SUNW,qlc, instance #0
            fp (driver not attached)
                disk (driver not attached)
            fp, instance #1
                ssd, instance #1
                ssd, instance #0 (driver not attached)
                ssd, instance #2 (driver not attached)
                ssd, instance #3 (driver not attached)
                ssd, instance #4 (driver not attached)
                ssd, instance #5 (driver not attached)
                ssd, instance #6 (driver not attached)
    upa, instance #0
        SUNW,ffb, instance #0 (driver not attached)
    ppm, instance #0
    pseudo, instance #0
```

x86 システムからは、次の `sysdef` 出力が表示されます。

```
# sysdef
* Hostid
*
  29f10b4d
*
* i86pc Configuration
*
*
* Devices
*
+boot (driver not attached)
memory (driver not attached)
aliases (driver not attached)
chosen (driver not attached)
i86pc-memory (driver not attached)
i86pc-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)
.
.
.
```

システムへ周辺デバイスを追加する

プラグイン不可の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

システムにホットプラグインできない次のデバイスを追加する場合は、81 ページの「周辺デバイスを追加する方法」の手順に従ってください。

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

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

ホットプラグ対応デバイスについては、第 6 章「デバイスの動的構成 (手順)」を参照してください。

▼ 周辺デバイスを追加する方法

1. スーパーユーザーになります。
2. (省略可能) 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、**82 ページの「デバイスドライバを追加する方法」**の手順を実行します。
3. `/reconfigure` ファイルを作成します。

   ```
   # touch /reconfigure
   ```

   この `/reconfigure` ファイルがあると、Solaris ソフトウェアは、次にシステムに電源を入れたときまたはブートしたときに、新しくインストールされたデバイスがないかどうかを検査します。
4. システムをシャットダウンします。

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

   `-i0` システムを init 0 状態に戻します。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ります。

   `-g30` システムを 30 秒以内にシャットダウンします。デフォルト値は 60 秒です。

`-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ます。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ます。

5 システムがシャットダウンしたら、次のいずれかを選択して電源を落とします。

- SPARC プラットフォームでは、`ok` プロンプトが表示されたら電源を落としても安全です。
- x86 プラットフォームでは、`type any key to continue` プロンプトが表示されたら電源を落としても安全です。

6 すべての周辺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ほかのデバイスとは異なることを確認します。

通常、ターゲット番号を選択するための小さいスイッチがディスクの裏側に付いています。

デバイスの設置と接続については、周辺デバイスに添付のハードウェアマニュアルを参照してください。

8 システムの電源を入れます。

システムがブートされてマルチユーザーモードになり、ログインプロンプトが表示されます。

9 周辺デバイスにアクセスし、そのデバイスが追加されたことを確認してください。

デバイスにアクセスする方法については、83 ページの「デバイスへのアクセス」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、76 ページの「標準サポートされていないデバイスを使用する場合」を参照してください。

1 スーパーユーザーになります。

2 テープ、フロッピーディスク、または **DVD** をドライブに入れます。

3 ドライバをインストールします。

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

`-d` *device*　パッケージを含むデバイスのパス名を指定します。

package-name　デバイスドライバを含むパッケージ名を指定します。

4 パッケージが正常に追加されたことを確認します。

```
# pkgchk package-name
#
```

パッケージが正しくインストールされている場合は、何も表示されません。

例 5–2 デバイスドライバを追加する

次の例では、`XYZdrv` というパッケージをインストールして確認します。

```
# pkgadd XYZdrv
(licensing messages displayed)
.
.
.
Installing XYZ Company driver as <XYZdrv>
.
.
.
Installation of <XYZdrv> was successful.
# pkgchk XYZdrv
#
```

デバイスへのアクセス

コマンドを使用してディスク、ファイルシステムなどのデバイスを管理する場合、デバイス名を指定する方法を知っている必要があります。通常、論理デバイス名を使用して、システムに接続されたデバイスを表すことができます。論理デバイス名と物理デバイス名は、システム上でそれぞれ論理デバイスファイルと物理デバイスファイルによって表現されます。

デバイス情報が作成される方法

システムがブートされると、デバイス階層が作成されて、システムに接続されたすべてのデバイスが表示されます。カーネルは、このデバイス階層情報を使用して、ドライバを該当するデバイスに対応づけます。また、カーネルは、特定の操作を実行するドライバへの一連のポインタを提供します。

デバイスの管理方法

devfs ファイルシステムは、システム上のすべてのデバイスの名前空間である /devices ディレクトリを管理します。このディレクトリは、実際のバスとデバイスのアドレスから成る「物理」デバイスを表します。

dev ファイルシステムは、「論理」デバイス名の名前空間である /dev ディレクトリを管理します。

デフォルトでは、devfsadm コマンドはすべてのドライバをシステムに読み込もうとし、可能な限りすべてのデバイスインスタンスに接続しようとします。次に、devfsadm コマンドはデバイスファイルを /devices ディレクトリに作成し、論理リンクを /dev ディレクトリに作成します。devfsadm コマンドはまた、path_to_inst インスタンスデータベースの保守も行います。

動的再構成イベントまたはファイルシステムアクセスに対応する /dev および /devices ディレクトリの更新は、devfsadmd (devfsadm コマンドのデーモンバージョン) によって処理されます。このデーモンは、システムのブート時にサービス管理機能によって起動されます。

devfsadmd デーモンは再構成イベントによるデバイス構成の変化を自動的に検出するため、このコマンドを対話的に実行する必要はありません。

詳細は、次のマニュアルページを参照してください。

- devfsadm(1M)
- dev(7FS)
- devfs(7FS)
- path_to_inst(4)

デバイス名の命名規則

Oracle Solaris OS 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、そのデバイスが最初にシステムに追加されるときまでに作成されます。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 ファイル内でマップされます。
- 論理デバイス名 – 論理デバイス名は、そのデバイスが最初にシステムに追加されるときまでに作成されます。論理デバイス名は、デバイスを参照する際に、ほとんどのファイルシステムコマンドで使用されます。論理デバイス名を使用する

ファイルコマンドの一覧は、表 5–3 を参照してください。/dev ディレクトリの論理デバイスファイルは、/devices ディレクトリの物理デバイスファイルにシンボリックリンクされています。

上記のデバイス名の情報は、次のコマンドによって表示できます。

- dmesg
- format
- sysdef
- prtconf

論理ディスクデバイス名

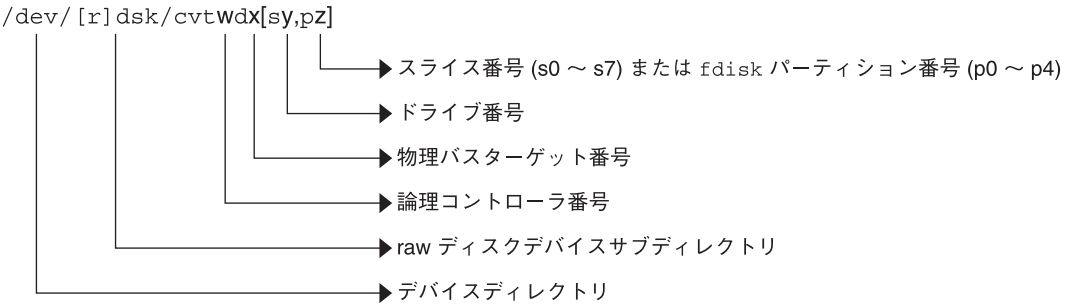
論理デバイス名は、次の作業を行う場合に、ディスクデバイスにアクセスするために使用されます。

- システムに新しいディスクを追加します。
- システム間でディスクを移動します。
- ローカルディスク上にあるファイルシステムにアクセスまたはそれをマウントします。
- ローカルファイルシステムのバックアップをとります。

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/rdsk のどちらか)に続けて、特定のコントローラ、ディスク、およびスライスを識別する文字列を指定することによって、ディスクデバイスを参照してください。

図 5–1 論理デバイス名の説明



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。デバイスからどのような方法でデータを読み取るかによって使い分けます。

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ーが含まれます。

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/rdsk` のどちらを必要とするかがわからない場合は、そのコマンドのマニュアルページの説明を参照してください。

次の表に、一般的に使用されるディスクコマンドとファイルシステムコマンドの一部に必要なインタフェースを示します。

表 5–3 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
df(1M)	ブロック	`df /dev/dsk/c0t3d0s6`
fsck(1m)	Raw	`fsck -p /dev/rdsk/c0t0d0s0`
mount(1M)	ブロック	`mount /dev/dsk/c1t0d0s7 /export/home/ziggy`
newfs(1M)	Raw	`newfs /dev/rdsk/c0t0d1s1`
prtvtoc(1M)	Raw	`prtvtoc /dev/rdsk/c0t0d0s2`

直接コントローラとバス指向コントローラ

ディスクのパーティションやスライスにアクセスする方法は、そのディスクデバイスが直接コントローラとバス指向コントローラのどちらに接続されているかによって異なる場合があります。一般的に、直接コントローラの論理デバイス名には「ターゲット」識別子が含まれません。

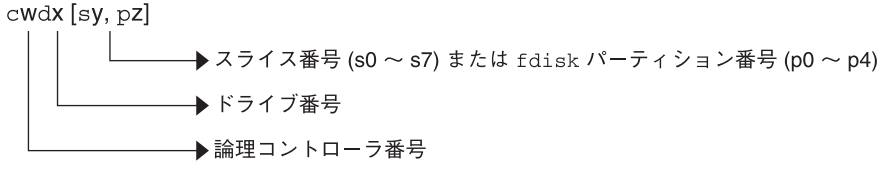
両方のタイプのコントローラについて、次の節で説明します。

注 – コントローラ番号は、システム初期設定時に自動的に割り当てられます。この番号は、厳密に論理的なものであり、物理コントローラに直接対応するものではありません。

x86: 直接コントローラでアクセスされるディスク

x86 システムにおいて IDE コントローラでアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

図 5–2 x86: 直接コントローラでアクセスされるディスク



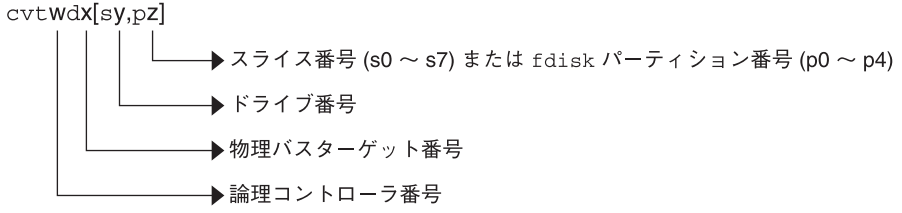
Solaris fdisk パーティション全体を示すには、スライス 2 (s2) を指定してください。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

バス指向コントローラでアクセスされるディスク

バス指向コントローラ (SCSI など) でアクセスされるディスクにスライスを指定するには、下の図に示す命名規則に従ってください。

図 5–3 バス指向コントローラでアクセスされるディスク



直接接続されるディスク (UltraSPARC システムの IDE ディスクなど) を備えている SPARC システムでは、バス指向コントローラを備えているシステムと同じ命名規則になります。

システムにコントローラが 1 つしかない場合、*w* は、通常、0 になります。

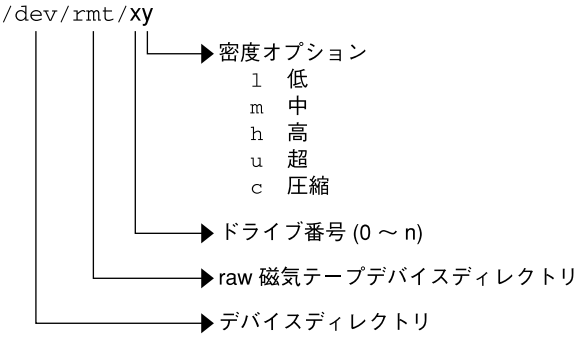
SCSI コントローラの場合、*x* はデバイスの背面にあるスイッチによって設定されたターゲットアドレス、*y* はターゲットに接続されたドライブの論理デバイス番号 (LUN) を示します。ディスクにコントローラが内蔵されている場合、*y* は、通常、0 になります。SPARC システム上の SCSI アドレスの詳細については、SunSolve Info Doc 48041 と、`scsi_address(9S)` のマニュアルページを参照してください。

ディスク全体を示すには、スライス 2 (`s2`) を指定してください。

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/*` ディレクトリ内にあります。

図 5–4 論理テープデバイス名



システムに接続された最初のテープデバイスは 0 (`/dev/rmt/0`) です。テープ密度の値 (`l`、`m`、`h`、`c`、および `u`) の詳細は、第 29 章「テープドライブの管理 (手順)」を参照してください。

論理リムーバブルメディアデバイス名

リムーバブルメディアは、リムーバブルメディア管理サービスによって管理されるため、論理デバイス名は、手動でメディアをマウントしないかぎり、通常使用されません。

システムのリムーバブルメディアデバイスを表す論理デバイス名については、第 3 章「リムーバブルメディアへのアクセス (手順)」を参照してください。

◆ ◆ ◆ 第 6 章

6 デバイスの動的構成 (手順)

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

デバイスを動的に構成する手順については、次の項目を参照してください。

- 94 ページの「cfgadm コマンドによる SCSI ホットプラグ (作業マップ)」
- 105 ページの「cfgadm コマンドによる PCI または PCIe ホットプラグ (作業マップ)」
- 113 ページの「アプリケーション開発者 RCM スクリプト (作業マップ)」
- 114 ページの「システム管理者 RCM スクリプト (作業マップ)」

cfgadm コマンドを使用して USB デバイスのホットプラグを実行する方法については、168 ページの「cfgadm コマンドを使った USB デバイスのホットプラグ」を参照してください。

cfgadm コマンドを使用して InfiniBand デバイスのホットプラグを実行する方法については、第 9 章「InfiniBand デバイスの使用 (概要/手順)」を参照してください。

デバイスへのアクセスに関する情報については、83 ページの「デバイスへのアクセス」を参照してください。

動的再構成とホットプラグ機能

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

一般に、次のバスタイプはホットプラグ対応です。

- USB
- ファイバチャネル
- 1394
- ATA
- SCSI
- Infiniband

また、`cfgadm` コマンドを使用すると、次のデバイスのホットプラグを実行できます。

- SPARC および x86 プラットフォームの USB デバイス
- SPARC および x86 プラットフォームの SCSI デバイス
- SPARC および x86 プラットフォームの PCI デバイス
- SPARC または x86 プラットフォームの PCIe デバイス
- SPARC および x86 デバイス上の InfiniBand デバイス

`cfgadm` コマンドには、次のような機能があります。

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

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

構成要素のホットプラグを実行する手順については、次を参照してください。

- 95 ページの「`cfgadm` コマンドによる SCSI ホットプラグ」
- 105 ページの「`cfgadm` コマンドによる PCI または PCIe ホットプラグ」
- `cfgadm`(1m)

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

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

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

注 – システムによっては、ホットプラグ対応のスロットとホットプラグに対応しないスロットを持つ場合があります。企業レベルのシステムなど、特定のハードウェア構成に対するデバイスのホットプラグについては、お使いのハードウェア構成のマニュアルを参照してください。

接続点

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

接続点は、次の要素から構成されています。

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

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

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

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

```
fas1:scsi
```

通常、SCSI デバイスの論理 `Ap_Id` の形式は次のようになります。

HBA-logical-apid::device-identifier

次の例において、`c0` は SCSI HBA の論理 `Ap_Id` です。

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

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

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

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

```
c0::st4
```

SCSI Ap_Id の詳細は、cfgadm_scsi(1M) のマニュアルページを参照してください。

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

次の表に、SCSI HBA 接続点の受容体と占有装置の状態を示します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI HBA には利用できません	configured	1 つまたは複数のデバイスがバス上で構成されています
disconnected	バスは休止しています	unconfigured	デバイスは構成されていません
connected	バスはアクティブです		

次の表に、SCSI デバイス接続点の受容体と占有装置の状態を示します。

受容体の状態	説明	占有装置の状態	説明
empty	SCSI デバイスには利用できません	configured	デバイスが構成されています
disconnected	バスは休止しています	unconfigured	デバイスは構成されていません
connected	バスはアクティブです		

SCSI 接続点の状態は特別なハードウェアによって示されない限り未知です。SCSI 構成要素の構成情報を表示する手順については、95 ページの「SCSI デバイスに関する情報を表示する方法」を参照してください。

PCI または PCIe アダプタカードの取り外し

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

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

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

PCI または PCIe アダプタカードの取り付け

PCI アダプタカードをシステムに取り付けるには、次の条件が必要です。

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

PCI アダプタカードの取り付けまたは取り外しの手順については、105 ページの「cfgadm コマンドによる PCI または PCIe ホットプラグ」を参照してください。

hotplug コマンドによる PCIe ホットプラグ

hotplug コマンドを使ってホットプラグ処理可能な接続を管理できます。ここで言う接続とは、PCI Express (PCIe) および PCI SHPC デバイス上のコネクタまたはポートのみを指します。ホットプラグコネクタとは、コンポーネントの挿入や取り外しが行われるシステムの物理的機構のことです。ホットプラグポートとは、システムへのデバイスの接続が管理される、システムデバイスツリー内の論理表現です。

hotplug 機能を使用することで、オンボードデバイスを含むデバイスをシステムに物理的に追加したり取り外したりせずに、オンラインやオフラインにできます。

hotplug コマンドを使ってデバイスを管理するには、hotplug サービスを有効にする必要があります。

```
# svcadm enable svc:/system/hotplug:default
```

次の例は、hotplug コマンドの使用方法を示しています。

次のようにして、システム内のホットプラグ処理可能な PCI/PCIe コネクタ/ポート (仮想および物理) をすべて表示します。

```
# hotplug list -lv
```

PCIe スロットの Ethernet カードを構成します。次に例を示します。

```
# hotplug enable /pci0,0 pcie0
```

PCIe スロットの Ethernet カードの構成を解除します。次に例を示します。

```
# hotplug disable /pci0,0 pcie0
```

PCI デバイスノードをオフラインにして、ノードのデバイスドライバを切り離します。次に例を示します。

```
# hotplug offline /pci0,0/pci1 pci.0,2
```

PCI デバイスノードをオンラインにして、ノードのデバイスドライバを接続します。次に例を示します。

```
# hotplug online /pci0,0/pci1 pci.0,2
```

PCI ホットプラグ処理 (hotplug) のトラブルシューティング

hotplug コマンドを使用するには、次のサービスが動作している必要があります。

```
svc:/system/hotplug:default
```

動作していない場合は、次のメッセージが表示されます。

```
ERROR: hotplug service is not available.
```

サポートされる入出力バスがシステムに存在しない場合、次のエラーメッセージが表示されます。

```
ERROR: there are no connections to display.
       (See hotplug(1m) for more information.)
```

上記のエラーメッセージは、ホットプラグ対応のほかの入出力デバイスがシステムに存在するが、それらのデバイスの管理には hotplug コマンドではなく cfgadm コマンドを使用する必要がある場合にも表示されます。

cfgadm コマンドによる SCSI ホットプラグ (作業マップ)

作業	説明	参照先
SCSI デバイスについての情報を表示します。	SCSI コントローラおよびデバイスについての情報を表示します。	95 ページの「SCSI デバイスに関する情報を表示する方法」
SCSI コントローラの構成を解除します。	SCSI コントローラの構成を解除します。	96 ページの「SCSI コントローラの構成を解除する方法」
SCSI コントローラを構成します。	構成を解除された SCSI コントローラを構成します。	97 ページの「SCSI コントローラを構成する方法」
SCSI デバイスを構成します。	特定の SCSI デバイスを構成します。	97 ページの「SCSI デバイスを構成する方法」

作業	説明	参照先
SCSI コントローラを切り離します。	特定の SCSI コントローラを切り離します。	98 ページの「SCSI コントローラを切り離す方法」
SCSI コントローラを接続します。	切り離された特定の SCSI コントローラを接続します。	99 ページの「SPARC: SCSI コントローラを接続する方法」
SCSI デバイスを SCSI バスに取り付けます。	特定の SCSI デバイスを SCSI バスに取り付けます。	100 ページの「SPARC: SCSI デバイスを SCSI バスに取り付ける方法」
SCSI コントローラ上の同一デバイスと交換します。	SCSI バス上のデバイスを、同じタイプの別のデバイスに交換します。	101 ページの「SPARC: SCSI コントローラ上の同一デバイスを交換する方法」
SCSI デバイスを取り外します。	SCSI デバイスをシステムから取り外します。	102 ページの「SPARC: SCSI デバイスを取り外す方法」
SCSI 構成に関する問題の障害追跡を行います。	失敗した SCSI 構成解除操作を解決します。	104 ページの「失敗した SCSI 構成解除操作の解決方法」

cfgadm コマンドによる SCSI ホットプラグ

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

注 – 通常、SCSI フレームワークは SCSI デバイスのホットプラグ機能をサポートしています。ただし、使用している SCSI デバイスでホットプラグ機能がサポートされているかどうか、ハードウェアのマニュアルを参照して確認してください。

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

▼ SCSI デバイスに関する情報を表示する方法

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

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

1 スーパーユーザーになります。

2 システム上の接続点に関する情報を表示します。

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

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

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

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

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

▼ SCSI コントローラの構成を解除する方法

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

1 スーパーユーザーになります。

2 **SCSI** コントローラの構成を解除します。

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

3 **SCSI** コントローラの構成が解除されていることを確認します。

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

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

構成解除処理が失敗した場合は、104 ページの「失敗した SCSI 構成解除操作の解決方法」を参照してください。

▼ SCSI コントローラを構成する方法

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

1 スーパーユーザーになります。

2 SCSI コントローラを構成します。

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

3 SCSI コントローラが構成されていることを確認します。

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

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

▼ SCSI デバイスを構成する方法

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

1 スーパーユーザーになります。

2 構成するデバイスを確認します。

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

3 **SCSI デバイスを構成します。**

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

4 **SCSI デバイスが構成されていることを確認します。**

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

▼ SCSI コントローラを切り離す方法

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

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

1 スーパーユーザーになります。

2 デバイスを切り離す前に、デバイスが接続されていることを確認します。

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

3 **SCSI コントローラを切り離します。**

```
# cfgadm -c disconnect c1
WARNING: Disconnecting critical partitions may cause system hang.
Continue (yes/no)? y
```

注意 – このコマンドは、cfgadm -c connect コマンドを使用するまで、SCSI バス上のすべての入出力動作を中断します。cfgamd コマンドは基本的な検査を行い、重要なパーティションが切り離されるのを防ぎます。しかし、すべての場合を発見できるわけではありません。このコマンドの使い方が不適切な場合、システムがハングし、システムをリブートしなければならなくなる可能性もあります。

4 **SCSI バスが切り離されていることを確認します。**

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

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

▼ SPARC: SCSI コントローラを接続する方法

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

1 スーパーユーザーになります。

2 接続する前に、デバイスが切り離されていることを確認します。

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

3 **SCSI コントローラを接続します。**

```
# cfgadm -c connect c1
```

4 **SCSI コントローラが接続されていることを確認します。**

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

▼ SPARC: SCSI デバイスを SCSI バスに取り付ける方法

SCSI コントローラ c1 を使用して、SCSI デバイスを SCSI バスに取り付ける方法を説明します。

注 – デバイスを取り付けるときは、デバイス自身の Ap_Id ではなく、デバイスを取り付ける SCSI HBA (コントローラ) の Ap_Id を指定します。

---

1 スーパーユーザーになります。

2 現在の **SCSI** 構成を確認します。

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

3 **SCSI** デバイスを **SCSI** バスに取り付けます。

a. 次の cfgadm コマンドを入力します。

次に例を示します。

```
# cfgadm -x insert_device c1
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000
This operation will suspend activity on SCSI bus: c1
```

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

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

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

c. デバイスを接続して、電源を入れます。

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

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

4 デバイスが取り付けられていることを確認します。

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

```
c1::dsk/c1t3d0       disk          connected    configured   unknown
c1::dsk/c1t4d0       disk          connected    configured   unknown
```

コントローラ c1 に新しいディスクが取り付けられました。

▼ SPARC: SCSI コントローラ上の同一デバイスを交換する方法

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

1 スーパーユーザーになります。

2 現在の **SCSI** 構成を確認します。

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

3 **SCSI** バス上のデバイスを、同じタイプの別のデバイスに交換します。

a. 次の cfgadm コマンドを入力します。

次に例を示します。

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

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

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

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

c. デバイスの電源を切ってから、そのデバイスを取り外します。

d. 交換用のデバイスを取り付けます。そして、取り付けたデバイスの電源を入れます。

交換用のデバイスは取り外したデバイスと同じタイプであり、同じアドレス (ターゲットと論理ユニット番号) でなければなりません。

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

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

4 デバイスが交換されていることを確認します。

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

▼ SPARC: SCSI デバイスを取り外す方法

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

1 スーパーユーザーになります。

2 現在の **SCSI** 構成を確認します。

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

3 **SCSI** デバイスをシステムから取り外します。

a. 次の cfgadm コマンドを入力します。

次に例を示します。

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

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

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

ホットプラグ処理の実行中、SCSI バス上の入出力動作は中断されます。

c. デバイスの電源を切ってから、そのデバイスを取り外します。

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

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

注 – この手順は、SCSI RAID デバイスを SCSI RAID アレイから取り外す場合に実行する必要があります。

---

4 デバイスがシステムから取り外されていることを確認します。

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

SCSI 構成に関する問題の障害追跡

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

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to offline:
     device-path
          Resource              Information
     ------------------  -------------------------
     /dev/dsk/c1t0d0s0   mounted filesystem "/file-system"
```

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

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

エラーメッセージ

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

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

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

エラーメッセージ

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

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

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

エラーメッセージ

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

エラーの発生原因
　専用ダンプデバイスを取り外そうとしたか、置き換えようとしました。

解決方法
　専用ダンプデバイスの構成を解除し、cfgadm 処理を再実行します。

▼ 失敗した SCSI 構成解除操作の解決方法

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

1. スーパーユーザーになります。
2. コントローラを再構成します。

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

cfgadm コマンドによる PCI または PCIe ホットプラグ (作業マップ)

次の作業マップでは、システム上の PCI または PCIe デバイスの管理作業について説明します。

作業	説明	参照先
PCI スロット構成情報を表示します。	システムの PCI ホットプラグ対応デバイスとスロットの状態を表示します。	106 ページの「PCI スロット構成情報を表示する方法」
PCI アダプタカードを取り外します。	カードを構成解除してからスロットから電源を外し、システムからカードを取り外します。	107 ページの「PCI アダプタカードを取り外す方法」
PCI アダプタカードを取り付けます。	アダプタカードをホットプラグ対応のスロットに挿入します。スロットを電源に接続し、カードを構成します。	109 ページの「PCI アダプタカードを取り付ける方法」
PCI 構成に関する問題の障害追跡を行います。	PCI 構成障害に対処するために、エラーメッセージと解決策を確認します。	110 ページの「PCI 構成に関する問題の障害追跡」

cfgadm コマンドによる PCI または PCIe ホットプラグ

この節では、SPARC および x86 システム上で PCI または PCIe アダプタカードのホットプラグを実行する手順について説明します。

ホットプラグ操作では、`cfgadm` コマンドに加えて、`prtconf` コマンドが便利です。`prtconf` コマンドはハードウェアに関連する追加の構成情報を表示します。

ハードウェアの追加後に `prtconf` コマンドを使用して、ハードウェアが正しく構成されているかどうかを確認します。たとえば、構成作業後に `prtconf -D` コマンドを使用して、新しくインストールしたハードウェアデバイスにドライバが設定されているかどうかを確認します。ハードウェアの構成の前にデバイスドライバがシステムに追加されていない場合、`add_drv` コマンドを使用して手動で追加された可能性があります。

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

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

PCIe LED インジケータの動作

システムの LED インジケータを監視して、スロットのホットプラグ動作の状態を視覚的に把握できます。PCI Express の場合の LED の動作は、PCI Express の仕様で定義された動作と一致するか、そうでない場合は動作がプラットフォームに依存することがあります。

詳細は、ハードウェアマニュアルを参照してください。PCI Express の場合は、Attention ボタンが押されると電源インジケータが点滅し、状態遷移が始まったことを示します。状態遷移が終了すると、点滅が終了します。

▼ PCI スロット構成情報を表示する方法

この手順は、PCIe 構成情報を含めるように更新されました。

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

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

     次に例を示します。

     ```
     # cfgadm
     Ap_Id                Type         Receptacle   Occupant     Condition
     pci1:hpc0_slot0      unknown      empty        unconfigured unknown
     pci1:hpc0_slot1      unknown      empty        unconfigured unknown
     pci1:hpc0_slot2      unknown      empty        unconfigured unknown
     pci1:hpc0_slot3      ethernet/hp  connected    configured   ok
     pci1:hpc0_slot4      unknown      empty        unconfigured unknown
     ```
   - 具体的な PCI デバイス情報を表示します。

     次に例を示します。

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

     論理 Ap_Id の pci1:hpc0_slot0 は、ホットプラグ対応のスロット Slot 7 の論理 Ap_Id です。構成要素 hpc0 はこのスロットのホットプラグ対応のアダプタカードを示し、pci1 は PCI バスのインスタンスを示します。Type フィールドは、スロット中の PCI アダプタカードのタイプを示します。

- PCIe スロット構成情報を表示します。

  次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          etherne/hp   connected    configured   ok
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

- 具体的な PCIe デバイス情報を表示します。

  次に例を示します。

```
# cfgadm -s "cols=ap_id:busy:o_state" pci
Ap_Id                          Busy     Occupant
pcie1                          n        unconfigured
pcie2                          n        unconfigured
pcie3                          n        unconfigured
pcie4                          n        configured
pcie5                          n        configured
pcie6                          n        configured
```

注 – ほとんどの場合、論理 Ap_Id は、システムのシャーシに貼られているスロットラベルと一致します。ハードウェアマニュアルでホットプラグ対応スロットの cfgadm の出力について参照してください。ホットプラグ操作を試す前に、Busy フィールドを表示して、 Ap_Id が別の状態に遷移していないことを確認できます。

▼ PCI アダプタカードを取り外す方法

PCIe アダプタカードを取り外す場合のために、次の手順が更新されました。ただし、アダプタカードを取り外す手順は、PCI と PCIe のどちらを使用するかに関係なく同じです。

1. スーパーユーザーになります。

2. **PCI アダプタカードが入っているスロットを確認します。**

   次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          etherne/hp   connected    configured   ok
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

3 デバイスを開いているアプリケーションを停止します。
たとえば、デバイスが Ethernet カードの場合、ifconfig コマンドでインタフェースを無効にしてからカードを引き抜きます。

4 下に示すように **cfgadm(1M)** コマンドを使用してデバイスを手動で構成解除します。または、**PCIe** アダプタカードを使用している場合は、自動構成方式を使用します。たとえば、ハードウェアマニュアルに従ってスロットの **Attention** ボタンを押します。

```
# cfgadm -c unconfigure pcie4
```

5 デバイスの構成が解除されていることを確認します。
次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          unknown      connected    unconfigured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

注 – デバイスが構成解除されると、 Type と Condition も unknown になります。

6 スロットへの電源を手動で切り離します。自動構成方式を使用する場合は、この手順は不要です。詳細は、ハードウェアマニュアルを参照してください。

```
# cfgadm -c disconnect pcie4
```

7 デバイスが切り離されていることを確認します。
次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          unknown      disconnected unconfigured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

8 プラットフォームのガイドの適切な手順に従って **PCI** アダプタカードを取り外します。カードを取り外すと、「**Receptable**」の状態が「**empty**」になります。
次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          unknown      empty        unconfigured unknown
```

```
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

注– プラットフォームの実装に応じて、ブート時に自動構成方式を有効または無効にすることができます。環境に適した自動構成方式を設定してください。

▼ PCI アダプタカードを取り付ける方法

PCIe アダプタカードを取り付ける場合のために、次の手順が更新されました。ただし、アダプタカードを追加する手順は、PCI と PCIe のどちらを使用するかに関係なく同じです。

1. スーパーユーザーになります。
2. ホットプラグ対応のスロットを確認して、ラッチを開きます。

   たとえば、pcie3 の場合は次のようになります。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      empty        unconfigured unknown
pcie4                          unknown      empty        unconfigured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

3. ハードウェアマニュアルの適切な手順に従って、**PCI** アダプタカードをスロットに挿入します。
4. **PCI** アダプタカードを挿入した後に、どのスロットに **PCI** アダプタカードが入っているかを確認します。

   次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      disconnected unconfigured unknown
pcie4                          unknown      empty        unconfigured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

5. `cfgadm` コマンドを使用して、手動で電源をスロットに接続します。または、**PCIe** アダプタカードを使用している場合は、自動構成方式を使用します。たとえば、ハードウェアマニュアルに従ってスロットの **Attention** ボタンを押します。

   次に例を示します。

```
# cfgadm -c connect pcie3
```

6 接続点が接続されていることを確認します。

次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          unknown      connected    unconfigured unknown
pcie4                          unknown      empty        unconfigured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

7 下に示すように cfgadm コマンドを使用して **PCI** アダプタカードを手動で構成します。自動構成方式を使用する場合は、この手順は不要です。詳細は、ハードウェアマニュアルを参照してください。

次に例を示します。

```
# cfgadm -c configure pcie3
```

8 スロット中の **PCI** アダプタカードの構成を確認します。

次に例を示します。

```
# cfgadm pci
Ap_Id                          Type         Receptacle   Occupant     Condition
pcie1                          unknown      empty        unconfigured unknown
pcie2                          unknown      empty        unconfigured unknown
pcie3                          etherne/hp   connected    configured unknown
pcie5                          pci-pci/hp   connected    configured   ok
pcie6                          unknown      disconnected unconfigured unknown
```

9 新しいデバイスの場合、サポートソフトウェアを構成します。

たとえば、デバイスが Ethernet カードの場合、ifconfig コマンドでインタフェースを設定します。

注 – プラットフォームの実装に応じて、ブート時に自動構成方式を有効または無効にすることができます。環境に適した自動構成方式を設定してください。

PCI 構成に関する問題の障害追跡

エラーメッセージ

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

エラーの発生原因

無効な移行を行いました。

解決方法

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

エラーメッセージ

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

エラーの発生原因
　指定した接続点は見つかりません。

解決方法
　接続点が正しいかどうかを確認します。cfgadm コマンドを使用して、利用可能な接続点のリストを表示します。物理パスを調べて、まだ接続点があるかどうかも確認してください。

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

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

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

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

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

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

RCM スクリプトについて

- 実行可能シェルスクリプト (Perl、sh、csh、または ksh) または RCM デーモンが実行するバイナリプログラム。推奨言語は、Perl です。

- スクリプトファイル所有者のユーザー ID を使用することにより、自分のアドレス領域で実行されるスクリプト。
- `cfgadm` コマンドを使ってシステムリソースを動的に再構成するときに、RCM デーモンによって実行されるスクリプト。

RCM スクリプトで実行できること

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

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

RCM スクリプト処理の動作方法

次のようにして RCM スクリプトを起動します。

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

RCM スクリプトにより、次の基本的な手順が実行されます。

1. コマンド行引数から RCM コマンドを取得します。
2. コマンドを実行する
3. 結果を名前と値のペアで `stdout` に記述します。
4. 適切な終了ステータスで終了します。

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

RCM スクリプトコマンド

次の RCM コマンドを RCM スクリプトに含める必要があります。

- `scriptinfo` - スクリプト情報を収集します
- `register` - リソースに処理対象を登録します
- `resourceinfo` - リソース情報を収集します

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

- `queryremove` - リソースが解放されたかどうかを問い合わせます
- `preremove` - リソースを解放します

- `postremove` - リソースの削除後に通知します
- `undoremove` - `preremove` で実行された動作を元に戻します

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

RCM スクリプト処理環境

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

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

RCM スクリプトでの作業

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

アプリケーション開発者 RCM スクリプト (作業マップ)

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

作業	説明	参照先
1. アプリケーションが使用するリソースを特定します。	アプリケーションが使用するリソース (デバイス名) を特定します。このデバイスは動的に取り外される可能性があります。	`cfgadm(1m)`
2. リソースを解放するコマンドを特定します。	アプリケーションからリソースを完全に解放するようにアプリケーションに通知するコマンドを特定します。	アプリケーションのマニュアル

作業	説明	参照先
3. リソースを取り外した後に使用するコマンドを特定します。	リソースを取り外したことをアプリケーションに通知するコマンドを含めます。	rcmscript(4)
4. リソースの取り外しに失敗した場合のコマンドを特定します。	使用可能なリソースについてアプリケーションに通知するコマンドを含めます。	rcmscript(4)
5. RCM スクリプトを記述します。	作業 1 - 4 で特定した情報に基づいて RCM スクリプトを記述します。	117 ページの「テープバックアップ用の RCM スクリプトの例」
6. RCM スクリプトをインストールします。	適切なスクリプトディレクトリにスクリプトを追加します。	115 ページの「RCM スクリプトのインストール方法」
7. RCM スクリプトをテストします。	手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストします。	116 ページの「RCM スクリプトのテスト方法」

システム管理者 RCM スクリプト (作業マップ)

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

作業	説明	参照先
1. 動的に削除するリソースを特定します。	`cfgadm -l` コマンドを使って削除する可能性があるリソース (デバイス名) を特定します。	cfgadm(1m)
2. 停止するアプリケーションを特定します。	アプリケーションを完全に停止させるコマンドを特定します。	アプリケーションのマニュアル
3. リソースの取り外し前および取り外し後のコマンドを特定します。	リソースを取り外す前後の動作を特定します。	rcmscript(4)
4. RCM スクリプトを記述します。	作業 1 - 3 で特定した情報に基づいて RCM スクリプトを記述します。	117 ページの「テープバックアップ用の RCM スクリプトの例」
5. RCM スクリプトをインストールします。	適切なスクリプトディレクトリにスクリプトを追加します。	115 ページの「RCM スクリプトのインストール方法」
6. RCM スクリプトをテストします。	手動でスクリプトコマンドを実行し、動的再構成操作を実行してスクリプトをテストします。	116 ページの「RCM スクリプトのテスト方法」

RCM スクリプトに名前を付ける

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

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

service スクリプトが表すサービス名です。

RCM スクリプトのインストールまたは削除

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

表 6–1 RCM スクリプトディレクトリ

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

▼ RCM スクリプトのインストール方法

1 スーパーユーザーになります。

2 適切なディレクトリにスクリプトをコピーします。
表 6–1 を参照してください。

次に例を示します。

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

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

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

4 SIGHUP を RCM デーモンに送信します。

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

▼ RCM スクリプトの削除方法

1 スーパーユーザーになります。

2 RCM スクリプトディレクトリからスクリプトを削除します。
次に例を示します。

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

3 SIGHUP を RCM デーモンに送信します。

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

▼ RCM スクリプトのテスト方法

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

```
$ export RCM_ENV_FORCE=TRUE
```

2 コマンド行から手動でスクリプトコマンドを実行してスクリプトをテストします。
次に例を示します。

```
$ script-name scriptinfo
$ script-name register
$ script-name preremove resource-name
$ script-name postremove resource-name
```

3 スクリプトの各 RCM スクリプトコマンドにより、適切な出力結果が stdout に印刷されるかどうかを確認します。

4 適切なスクリプトディレクトリにスクリプトをインストールします。
詳細は、115 ページの「RCM スクリプトのインストール方法」を参照してください。

5 動的な削除操作を実行してスクリプトをテストします。
たとえば、スクリプトによってデバイス /dev/dsk/c1t0d0s0 が登録されたとします。次のコマンドを実行してください。

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

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

テープバックアップ用の RCM スクリプトの例

ここでは、RCM スクリプトを使用したテープバックアップの例を示します。

テープバックアップ用の RCM スクリプトの役割

テープバックアップ用の RCM スクリプトは、次の手順を実行します。

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

   ```
   rcm_script_func_info=Tape backup appl script for DR
   ```
4. すべてのテープドライブのデバイス名を `stdout` に印刷して、すべてのテープドライバをシステムに登録します。

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

   エラーが発生した場合、スクリプトによりエラー情報が `stdout` に出力されます。

   ```
   rcm_failure_reason=$errmsg
   ```
5. テープデバイスのリソース情報を設定します。

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

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

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

テープバックアップ再構成シナリオの結果

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

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

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

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

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

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

例 — テープバックアップ用の RCM スクリプト

```
#! /usr/bin/perl -w
   #
   # A sample site customization RCM script.
   #
   # When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot
   # release the tape drive when the tape drive is being used for backup.
   #
   # When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive
   # when the tape drive is being used for backup by killing the tape
   # backup application.
   #

    use strict;

    my ($cmd, %dispatch);
    $cmd = shift(@ARGV);
# dispatch table for RCM commands
    %dispatch = (
            "scriptinfo"    =>      \&do_scriptinfo,
            "register"      =>      \&do_register,
            "resourceinfo"  =>      \&do_resourceinfo,
            "queryremove"   =>      \&do_preremove,
            "preremove"     =>      \&do_preremove
    );


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

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

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

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

            while ($f = readdir(RMT)) {
                # ignore hidden files and multiple names for the same device
                if (($f !~ /^\./) && ($f =~ /^[0-9]+$/)) {
                        print "rcm_resource_name=/dev/rmt/$f\n";
                    }

            }

            closedir(RMT);
            exit (0);
    }
sub do_resourceinfo
    {
      my ($rsrc, $unit);

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

    sub do_preremove
    {
            my ($rsrc);

            $rsrc = shift(@ARGV);

            # check if backup application is using this resource
            #if (the backup application is not running on $rsrc) {
                    # allow the DR to continue
            #        exit (0);
            #}
            #
            # If RCM_ENV_FORCE is FALSE deny the operation.
            # If RCM_ENV_FORCE is TRUE kill the backup application in order
            # to allow the DR operation to proceed
```

```
                #
                if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
                     if ($cmd eq 'preremove') {
                             # kill the tape backup application
                     }
                     exit (0);
                } else {
                   #
                   # indicate that the tape drive can not be released
                   # since the device is being used for backup by the
                   # tape backup application
                   #
                   print "rcm_failure_reason=tape backup in progress pid=...\n"
;
                   exit (3);
                }
        }
```

◆ ◆ ◆ 第 7 章

USB デバイスの使用 (概要)

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

この章の内容は以下のとおりです。

- 122 ページの「USB デバイスの新機能」
- 126 ページの「USB デバイスの概要」
- 133 ページの「Oracle Solaris OS における USB について」

USB デバイスに関する最新の情報については、次の Web サイトを参照してください。

`http://www.sun.com/io_technologies/usb/USB-Faq.html`

USB デバイスに関する一般的な情報については、次の Web サイトを参照してください。

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

Oracle Solaris OS における USB デバイスの使用手順については、第 8 章「USB デバイスの使用 (手順)」を参照してください。

動的再構成およびホットプラグについての一般的な情報については、第 6 章「デバイスの動的構成 (手順)」を参照してください。

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

USB デバイスの新機能

次の節では、Solaris リリースにおける USB の新機能について説明します。

- 122 ページの「EHCI アイソクロナス転送のサポート」
- 122 ページの「CDC ACM デバイスのサポート」
- 122 ページの「USB デバイスのホットプラグ対応動作の変更」
- 123 ページの「USB の電源割り当て」
- 123 ページの「USB デバイスでの Oracle Solaris ZFS のサポート」
- 123 ページの「Prolific および Keyspan シリアルアダプタのサポート」
- 123 ページの「x86: GRUB ブートにおける USB CD および DVD のサポート」
- 124 ページの「USB 仮想キーボードとマウスのサポート」
- 124 ページの「`vold` によるホットプラグ対応 USB デバイスの認識」

Solaris の新機能の一覧および Solaris リリースについての説明は、『Oracle Solaris 10 9/10 の新機能』を参照してください。

EHCI アイソクロナス転送のサポート

Solaris 10 8/07: USB EHCI ホストコントローラドライバは、USB 2.0 または高速のアイソクロナスデバイスでアイソクロナス転送を行えるようにします。詳細は、`usb_isoc_request(9S)` のマニュアルページを参照してください。

CDC ACM デバイスのサポート

Solaris 10 8/07: このリリースでは、CDC ACM デバイスがサポートされます。詳細については、129 ページの「USB ドライバの機能拡張」を参照してください。

USB デバイスのホットプラグ対応動作の変更

Solaris 10 6/06: この機能の情報は、Solaris 10 11/06 リリースで改訂されています。

この Solaris リリースでは、新規デバイス属性 *hotpluggable* が導入されました。この属性を使用すると、システムをリブートせずに接続または切り離しできるデバイスを識別したり、ユーザーの介入なしで自動的に設定および設定解除を実行したりできます。すべての USB および 1394 デバイスは、ホットプラグ対応デバイスとして識別され、142 ページの「USB 大容量ストレージデバイスの使用」に記述されている利点が得られます。また、非リムーバブルメディア USB および 1394 デバイスは、リムーバブルメディアデバイスとしては識別されなくなり、*removable-media* 属性を持たなくなりました。

この変更は、非リムーバブルメディア USB および 1394 デバイスのサポートおよびそのパフォーマンスを改善する目的で、主にカーネルレベルで行われました。ただし、これらのデバイスの使用法が変更の影響を受けることはありません。たとえば、これらのデバイスのマウントおよびマウント解除は、`vold` を使用して制御します。ユーザーが識別できるのは、デバイスの hotpluggable および removable-media 属性の変更だけです。

詳細は、71 ページの「USB と 1394 (FireWire) サポートの機能拡張」を参照してください。

USB デバイスでの Oracle Solaris ZFS のサポート

ZFS ファイルシステムは、USB 大容量ストレージデバイス上で作成およびマウントできます。USB 大容量ストレージデバイスの使用方法の詳細は、142 ページの「USB 大容量ストレージデバイスの使用」を参照してください。

ZFS ファイルシステムの作成およびマウントについては、`zfs(1M)` と `zpool(1M)` のマニュアルページを参照してください。

Prolific および Keyspan シリアルアダプタのサポート

Solaris 10 6/06: 以前に、この機能は Solaris 10 1/06 リリースで使用可能とされていましたが、これは正しい情報ではありません。この機能は、Solaris 10 6/06 リリース以降で使用できます。

USB の電源割り当て

Solaris 10 6/06: この Solaris リリースには、USB デバイスに供給される電源の管理を改善するための、USB デバイスの電源割り当て機能が実装されています。電源割り当ての制御は、過電流状態の発生を防ぎ、USB デバイスをより安全に使用するのに役立ちます。Solaris USB 電源割り当て制限の詳細については、135 ページの「バス電源供給方式のデバイス」を参照してください。

x86: GRUB ブートにおける USB CD および DVD のサポート

Solaris 10 1/06: GRUB ベースのブート環境で次の USB 機能を使用できます。

- USB CD または DVD ドライブからのインストール

- USB ストレージデバイスからのブート。Solaris リリースをブートする前に、USB ドライブに Solaris リリースをインストールする必要があります。

GRUB ブートの詳細は、『Solaris のシステム管理 (基本編)』の第 9 章「システムのシャットダウンとブート (概要)」を参照してください。

USB 仮想キーボードとマウスのサポート

Solaris 10 1/06: USB 仮想キーボードおよびマウスがサポートされているため、複数のキーボードおよび複数のマウスを接続して、それらを 1 つの仮想キーボードまたは仮想マウスとして動作させることができます。つまり、各物理デバイスの入力が 1 つの入力ストリームに統合されます。たとえば、一方のキーボードで Shift キーを押してもう一方のキーボードで A キーを押した場合、表示される文字は大文字の A になります。

また、USB キーボードまたはマウスをラップトップに追加し、これらのデバイスをラップトップの PS/2 キーボードおよびパッドと一緒に 1 つのデバイスとして動作させる機能もサポートされています。

さらに、仮想キーボードおよびマウス機能により、バーコードリーダーもサポートされています。

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

vold によるホットプラグ対応 USB デバイスの認識

Solaris 10 1/06: リムーバブルメディアマネージャー (`vold`) はホットプラグ対応になりました。ホットインサートされた USB 大容量ストレージデバイスをマウントするために、このデーモンを再起動する必要はありません。ただし、一部のデバイスでは `vold` が正常に機能しないことがあるため、手動によるデバイスのマウントが必要な場合があります。`vold` による USB デバイスの自動マウントが失敗した場合は、次のコマンドを実行して `vold` を停止します。

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

USB 大容量ストレージデバイスを手動でマウントする方法については、159 ページの「vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」を参照してください。

USB デバイスの Solaris サポート

USB 1.1 および USB 2.0 の各デバイスが Solaris でどのようにサポートされるかを確認するには、次の表を使用してください。

USB デバイス	Solaris 8 HW 5/03 以降のリリース	Solaris 9 リリース	Solaris 10 リリース
一般的な **USB 1.1** デバイスのサポート	SPARC と x86	SPARC と x86	SPARC と x86
一般的な **USB 2.0** デバイスのサポート	SPARC のみ	SPARC と x86 (Solaris 9 4/04)	SPARC と x86
特定の **USB 1.1** および **USB 2.0** デバイスのサポート			
オーディオデバイス (下の注意事項を参照してください。)	**USB 1.1** のみ: USB 2.0 ハブではサポートされません	**USB 1.1** のみ: USB 2.0 ハブではサポートされません	**USB 1.1** のみ: USB 2.0 ハブでサポートされます
汎用 USB ドライバ (ugen(7D))	SPARC のみ	SPARC と x86	SPARC と x86
HID デバイス (キーボードおよびマウスデバイス、hid(7D))	SPARC と x86	SPARC と x86	SPARC と x86
ハブ (hubd(7D))	SPARC と x86	SPARC と x86	SPARC と x86
プリンタ	SPARC と x86	SPARC と x86	SPARC と x86
シリアルデバイス (Edgeport (usbser_edge(7D))、Prolific (usbsprl(7D))、Keyspan (usbsksp(7D))	SPARC と x86	SPARC と x86	SPARC と x86
ストレージデバイス (scsa2usb(7D))	SPARC と x86	SPARC と x86	SPARC と x86
ユーザー領域 USB デバイス管理ライブラリ (libusb(3LIB))	サポートされていません	サポートされていません	SPARC と x86

注:

- USB 1.x オーディオデバイスのみがサポートされます。USB 2.0 オーディオデバイスはサポートされません。
- USB 2.0 ポートに接続されている USB 2.0 ハブに接続された USB 1.x オーディオデバイスは、Solaris 10 リリースでのみ使用できます。詳細は、usb_ac(7D) および usb_as(7D) のマニュアルページを参照してください。

- USB ドライバによってサポートされていないデバイスのために、gphoto2、gtkam、および pilotlink などの libusb アプリケーションが用意されています。詳細は、/usr/sfw/share/doc/libusb/libusb.txt を参照してください。
- **Solaris 8** および **Solaris 9** リリース – USB デュアルフレームワークの問題については、次の Web サイトを参照してください。

  http://www.sun.com/io_technologies/usb/USB-Faq.html

大容量ストレージデバイスに関連する作業については、第 8 章「USB デバイスの使用 (手順)」を参照してください。

ugen の詳細は、129 ページの「USB ドライバの機能拡張」を参照してください。

USB デバイスの概要

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

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

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

詳細は、usba(7D) のマニュアルページまたは次の Web サイトを参照してください。

http://www.usb.org/home

よく使用される USB 関連の略語

次の表に、Oracle Solaris OS で使用される USB の略語を示します。USB の構成要素と略語についての詳細は、次の Web サイトを参照してください。

http://www.usb.org/home

略語	定義	参照先
UGEN	USB 汎用ドライバ	ugen(7D)
USB	Universal Serial Bus	usb(7D)
USBA	USB アーキテクチャー (Solaris)	usba(7D)

略語	定義	参照先
USBAI	USBA クライアントドライバインタフェース (Solaris)	なし
HCD	USB ホストコントロールドライバ	なし
EHCI	拡張ホストコントローラインタフェース	ehci(7D)
OHCI	オープンホストコントローラインタフェース	ohci(7D)
UHCI	ユニバーサルホストコントローラインタフェース	uhci(7D)

USB バスの説明

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

USB が採用するトポロジでは、ハブが USB デバイスに接続点を提供します。ホストコントローラには、システム内のすべての USB ポートの起点となるルートハブが含まれます。ハブの詳細は、136 ページの「USB ホストコントローラとハブ」を参照してください。

図 7–1　USB 物理デバイスの階層

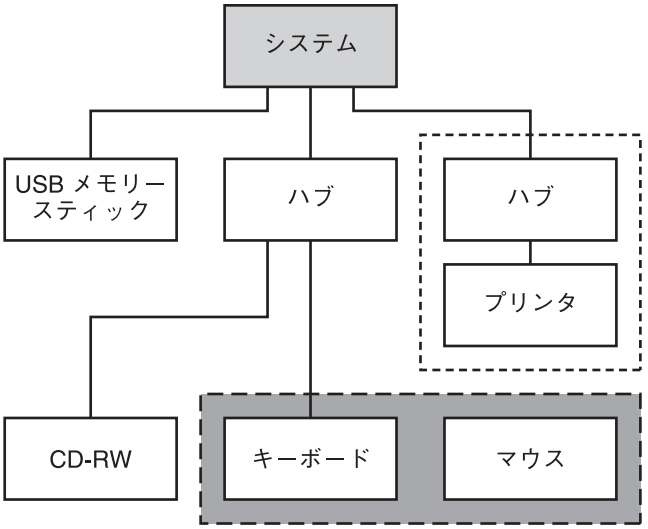


図 7–1 は、有効な USB ポートが 3 つ搭載されたシステムを示しています。1 番目の USB ポートは USB メモリースティックに接続されています。2 番目の USB ポートは外部ハブに接続されており、このハブには CD-RW デバイスと、キーボードとマウスの複合デバイスが接続されています。このキーボードは「複合デバイス」であるため USB コントローラが組み込まれており、このコントローラによって、キーボードとキーボードに接続されたマウスの両方が制御されます。キーボードとマウスは、同じ USB コントローラによって制御されるため、同一の USB バスアドレスを共有します。

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

次の表に、図 7–1 に示したデバイスの一部について、デバイスツリーパス名を一覧表示します。

メモリースティック	`/pci@1f,4000/usb@5/storage@1`
キーボード	`/pci@1f,4000/usb@5/hub@2/device@1/keyboard@0`
マウス	`/pci@1f,4000/usb@5/hub@2/device@1/mouse@1`
CD-RW デバイス	`/pci@1f,4000/usb@5/hub@2/storage@3`

プリンタ　　/pci@1f,4000/usb@5/hub@3/printer@1

USB デバイスとドライバ

属性とサービスが似ている USB デバイスは、いくつかのデバイスクラスに分類されます。各デバイスクラスには対応するドライバが 1 つずつ存在しています。クラス内のデバイスは、同じ組み合わせのデバイスドライバによって管理されます。ただし、USB 仕様では、特定のクラスに属さないベンダー固有のデバイスも許可しています。

Human Interface Device (HID) クラスには、ユーザーが制御する次のようなデバイスが含まれます。

- キーボード
- マウスデバイス
- ジョイスティック

Communication Device クラスには、次のデバイスが含まれます。

- モデム
- Ethernet アダプタ

その他にも、次のようなデバイスクラスがあります。

- Audio Device
- モニター
- プリンタ
- Storage Device

各 USB デバイスはデバイスのクラスを表す記述子を持っています。デバイスクラスは、そのメンバーが構成とデータ転送についてどのように動作するかを指定します。クラス情報の詳細は、次の Web サイトを参照してください。

`http://www.usb.org/home`

Solaris リリースでサポートされる USB デバイスの詳細は、`usb(7D)` のマニュアルページを参照してください。

USB ドライバの機能拡張

次の USB ドライバ拡張が含まれています。

- **USB CDC ACM** デバイスのサポート – `acm` ドライバは、USB Communication Device Class 仕様の Abstract Control Model に準拠するデバイスと、モデム機能を持ついくつかの PCMCIA カードで動作します。

  `pppd` デーモンは、`/dev/term/[0~9]*` エントリを通してこれらのデバイスにアクセスできます。詳細は、`pppd(1M)` のマニュアルページを参照してください。

詳細は、usbsacm(7D) のマニュアルページを参照してください。

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

  Solaris 10 6/06 リリースから、デバイスへの明示的なバインドに ugen ドライバは不要になりました。デフォルトでは、usb_mid により、クラスドライバの存在しないデバイスへのバインド、および libusb で動作する ugen インタフェースのエクスポートが実行されます。たとえば、大容量ストレージデバイスではない USB カメラを接続し、libusb アプリケーションを使ってアクセスできます。また、scsa2usb ドライバと usbprn ドライバの両方が ugen インタフェースをエクスポートするため、これらのデバイスクラス上で libusb アプリケーションを直接使用できます。

  詳細は、ugen(7D) のマニュアルページを参照してください。
- **USB** シリアルドライバのサポート
  - Digi Edgeport USB サポート – Edgeport USB ドライバは、Edgeport デバイスでのみ機能し、ほかの USB シリアルデバイスでは機能しません。
    - 新しいデバイスには、/dev/term/[0-9]* または /dev/cua/[0-9]* としてアクセスできます。
    - USB シリアルポートをその他のシリアルポートと同じように使用できます。ただし、ローカルシリアルコンソールの機能はありません。ユーザーから見て、USB ポート経由でデータが実行されていることはわかりません。

    詳細は、usbser_edge(7D) のマニュアルページを参照するか、次の Web サイトにアクセスしてください。
    - http://www.digi.com
    - http://www.sun.com/io
  - Keyspan – Keyspan USB シリアルドライバは Keyspan デバイスでのみ動作します。現在サポートしているのは USA-19HS モデルと USA-49WLC モデルです。

    詳細は、usbsksp(7D) のマニュアルページを参照してください。
  - Prolific – Prolific USB シリアルドライバは、PL2303 チップセットをベースとするデバイスだけで動作します。

    詳細は、usbsprl(7D) のマニュアルページを参照してください。

USB to serial デバイスのサポートについては、次のサイトを参照してください。

`http://www.sun.com/io_technologies/usb/USB-Faq.html`

- ユーザーが記述したカーネルおよびユーザー独自のドライバの文書サポートおよびバイナリサポート – USB ドライバの開発に関する最新の情報については、次の Web サイトを参照してください。
  - `http://www.sun.com/io_technologies/usb/USB-Faq.html`
  - http://developers.sun.com/solaris/developer/support/driver/usb.html
  - 『Writing Device Drivers』の第 20 章「USB Drivers」
  - 『Writing Device Drivers』の付録 C「Making a Device Driver 64-Bit Ready」
  - 『Device Driver Tutorial 』
  - `Intro(7)`、`Intro(9F)`、および `Intro(9S)`
  - `http://developers.sun.com/solaris/driverdev/reference/codesamples/usb_security/index.html`

EHCI ドライバ、OHCI ドライバ、および UHCI ドライバ

EHCI ドライバには、次の特徴があります。

- USB 2.0 をサポートする拡張ホストコントローラインタフェースに準拠しています。
- 高速の制御転送、バルク転送、割り込み転送、およびアイソクロナス転送をサポートします。
- USB 2.0 チップには、1 つの EHCI コントローラと 1 つ以上の OHCI または UHCI コントローラが組み込まれています。
- USB 1.1 デバイスを接続すると、OHCI または UHCI コントローラに動的に割り当てられます。USB 2.0 デバイスを接続すると、EHCI コントローラに動的に割り当てられます。

`prtconf` コマンドの出力を使用して、システムが USB 1.1 デバイスまたは USB 2.0 デバイスをサポートしているかどうかを確認します。次に例を示します。

```
# prtconf  -D | egrep "ehci|ohci|uhci"
```

`prtconf` の出力で EHCI コントローラが確認される場合、システムは USB 2.0 デバイスをサポートしています。

`prtconf` の出力で OHCI または UHCI コントローラが確認される場合、システムは USB 1.1 デバイスをサポートしています。

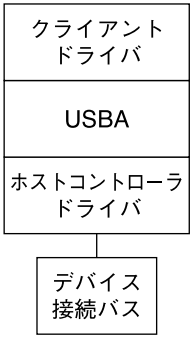
Solaris USB アーキテクチャー (USBA)

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

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

Solaris USB アーキテクチャー (USBA) は、USB 1.1 および USB 2.0 の仕様に準拠しており、Solaris デバイスドライバインタフェース (DDI) の一部です。USBA モデルは Sun Common SCSI Architecture (SCSA) に似ています。次の図が示すように、USBA は、汎用 USB トランスポート層という概念をクライアントドライバに提供する薄い層で、汎用 USB の主要な機能を実装するサービスをクライアントドライバに提供します。

図 7–2　Solaris USB アーキテクチャー (USBA)



Oracle Solaris OS における USB について

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

USB 2.0 機能

次に示す USB 2.0 の機能が含まれます。

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

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

USB デバイスおよび USB 用語については、126 ページの「USB デバイスの概要」を参照してください。

USB 2.0 デバイスの機能および互換性の問題

USB 2.0 デバイスは、USB 2.0 仕様に準拠した高速デバイスです。USB 2.0 仕様は、`http://www.usb.org/home` で確認できます。

Solaris 10 リリースで USB デバイスの速度を識別するには、`/var/adm/messages` ファイルで次のようなメッセージを確認します。

```
Dec 13 17:05:57 mysystem usba: [ID 912658 kern.info] USB 2.0 device
(usb50d,249) operating at hi speed (USB 2.x) on USB 2.0 external hub:
storage@4, scsa2usb0 at bus address 4
```

この Solaris リリースでは、たとえば、次の USB デバイスがサポートされます。

- 大容量ストレージデバイス。たとえば CD-RW、ハードディスク、DVD、デジタルカメラ、フロッピーディスク、テープドライブ、メモリースティック、およびマルチフォーマットのカードリーダー
- キーボードおよびマウス
- オーディオデバイス。たとえばスピーカおよびマイク

今回の Solaris リリースで検証済みの USB デバイスの一覧は、次のサイトで確認できます。

http://www.sun.com/io_technologies/USB.html

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

Solaris USB 2.0 デバイスのサポートとして、次の機能が用意されています。

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

  次のいずれかの USB 2.0 ポートを使用できます。

  - USB 2.0 PCI カード上のポート
  - USB 2.0 ポートに接続された USB 2.0 ハブ上のポート
  - SPARC または x86 コンピュータのマザーボード上のポート

  以前の SPARC プラットフォームでは、USB 2.0 PCI カードが必要な場合があります。
- Solaris リリースで検証済みの USB 2.0 PCI カードについては、次の Web サイトを参照してください。

  http://www.sun.com/io_technologies/USB.html
- 同じシステム上に USB 1.1 デバイスと USB 2.0 デバイスが共存する場合でも、USB 1.1 デバイスは以前と同様に機能します。
- USB 2.0 デバイスは USB 1.x ポートでも動作しますが、USB 2.0 ポートに接続するとパフォーマンスが大幅に向上します。
- USB 2.0 ホストコントローラには、1 つのハイスピード EHCI (Enhanced Host Controller Interface) と、1 つ以上の OHCI (OpenHCI Host Controller Interface) または UHCI (Universal Host Controller Interface) が組み込まれたコントローラが用意されています。USB 2.0 ポートに接続されているデバイスは、USB 2.0 をサポートしているかどうかに応じて、EHCI または OHCI コントローラに動的に割り当てられます。

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

また、USB デバイスの速度は親ポートがサポートできる速度に制限されます。たとえば、USB 1.x ハブに USB 2.0 外部ハブが接続され、その USB 2.0 外部ハブに USB 2.0 デバイスが接続されている場合、そのデバイスは最大限の速度で動作した場合でもそれほど高速では動作しません。

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

バス電源供給方式のデバイス

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

Solaris 10 6/06 リリースから、USB デバイスへの電源割り当てが実装されています。この機能には、次の制限があります。

- 2 台のバス電源供給方式のハブをカスケード接続しないでください。
- バス電源供給方式のハブの各ポートの最大消費電力は100mAです。
- バス電源供給方式のハブに接続できるのは、自己電源のデバイスまたはバス電源供給方式の低電力デバイスだけです。バス電源供給方式の高電力デバイスは接続を拒否されます。接続は予測できない場合があるため、誤った電源をレポートするハブやデバイスもあります。

USB キーボードとマウス

USB キーボードおよびマウスデバイスを使用するときは、次のことに注意してください。

- SPARC システムにおいて、リブート中または ok プロンプトの出ている間は、キーボードおよびマウスを移動しないでください。システムのリブート後であれば、いつでもキーボードおよびマウスを別のハブに移動できます。キーボードおよびマウスは、差し込んだ後は再び完全に機能します。

- Sun 社製以外の USB キーボードでは、キーパッドの左側にある機能は使用できない場合があります。
- **SPARC** – SPARC システムにおいて USB キーボードおよびマウスデバイスを使用するときは、次のことに注意してください。
  - USB キーボードの電源キーと Sun タイプ 5 キーボードの電源キーの動作は異なります。USB キーボードでは、「SUSPEND/SHUTDOWN」キーを使用してシステムを中断またはシャットダウンできます。ただし、そのキーを使用してシステムの電源を入れることはできません。
  - ブートプロセスが完了するまでは、OpenBoot PROM (OBP) の制限により、キーボードおよびマウスデバイスはマザーボードのルートハブのポートにしか接続できません。
  - Ultra 80 などの旧バージョンの SPARC システムでは、USB キーボードおよびマウスデバイスをタイプ 3、タイプ 4 またはタイプ 5 キーボードと同時に使用できません。
- 複数のキーボードおよびマウスデバイスのサポートについては、`virtualkm(7D)` のマニュアルページを参照してください。

USB ホイール付きマウスのサポート

Solaris 9 9/04 リリース以降では、次のホイール付きマウス機能がサポートされています。

- USB マウスまたは PS/2 マウスデバイスでは、3 ボタン以上の使用がサポートされています。
- USB マウスまたは PS/2 マウスデバイスでは、ホイール付きマウスによるスクロールが使用可能です。USB マウスまたは PS/2 マウスでホイールを回転させると、マウスが置かれたアプリケーションまたはウィンドウがスクロールします。StarSuite、Firefox、および GNOME アプリケーションはホイール付きマウスによるスクロール機能をサポートします。その他のアプリケーションの中には、この機能をサポートしないものもあります。

USB ホストコントローラとハブ

USB ハブは次のことを行います。

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

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

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

USB ハブデバイス

- SPARC システムと x86 システムのどちらにおいても、ハブは 4 段を超えて多段接続しないでください。SPARC システムでは、OpenBoot PROM (OBP) は 4 段を超えるデバイスを正確に認識できません。
- バス電源供給方式のハブ同士をカスケード接続しないでください。バス電源供給方式のハブは独自の電源を持っていません。
- 大量の電源を必要とするデバイスをバス電源供給方式のハブに接続しないでください。これらのデバイスがバス電源供給方式のハブへの接続を拒否されたり、ほかのデバイス用の電源がなくなったりする可能性があります。このようなデバイスとして、USB フロッピーディスクデバイスなどがあります。

SPARC: USB 電源管理

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

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

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

電源消費を減らすために電源管理を行なっている場合は、まず USB 末端デバイスの電源が切断されます。また、ハブのポートに接続されているすべてのデバイスの電

源が切断されると、しばらくしてからハブの電源が切断されます。もっとも効率的に電源管理をするためには、あまり多くのハブをカスケード接続しないでください。

SPARC システムの SUSPEND/SHUTDOWN キーの使用についての詳細は、135 ページの「USB キーボードとマウス」を参照してください。

USB ケーブルに関するガイドライン

USB ケーブルを接続する際には、次のガイドラインに従ってください。

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

詳細は、次の Web サイトを参照してください。

`http://www.usb.org/about/faq/ans5`

◆ ◆ ◆ 第 8 章

USB デバイスの使用 (手順)

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

USB デバイスの使用手順については、次の節を参照してください。

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

USB デバイスに関する最新の情報については、次の Web サイトを参照してください。

`http://www.sun.com/io_technologies/usb/USB-Faq.html`

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

Oracle Solaris OS での USB デバイスの管理 (作業マップ)

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

Oracle Solaris OS で USB コンポーネントを使用する方法については、133 ページの「Oracle Solaris OS における USB について」を参照してください。

作業	説明	参照先
USB 大容量ストレージデバイスを使用します。	USB 大容量ストレージデバイスにデータを追加する前に、USB デバイスにファイルシステムが作成されている必要があります。また、ファイルシステムを作成およびマウントする前に、USB フロッピーディスクをフォーマットする必要があります。 この節では、USB デバイスをシステムに物理的に取り付け、システムから物理的に取り外す方法についても説明します。	140 ページの「USB 大容量ストレージデバイスの使用 (作業マップ)」
USB オーディオデバイスを追加します。	この作業マップを使用して、USB オーディオデバイスの追加に関連する作業を特定します。	163 ページの「USB オーディオデバイスの使用 (作業マップ)」
`cfgadm` コマンドを使用して、USB デバイスをシステムに追加したり、システムから取り外したりします。	`cfgadm` コマンドを使用して、USB デバイスをシステムに論理的に追加したり、システムから論理的に取り外したりします。	167 ページの「`cfgadm` コマンドを使用した USB デバイスのホットプラグ (作業マップ)」

USB 大容量ストレージデバイスの使用 (作業マップ)

作業	説明	参照先
USB 大容量ストレージデバイスの取り付けまたは取り外しを行います。	次のいずれかの方法を選択して、USB 大容量ストレージデバイスを取り付けます。	
	USB 大容量ストレージデバイスを追加します。	146 ページの「`vold` を使用しないで USB 大容量ストレージデバイスを追加する方法」
	USB カメラを追加して、デジタル画像にアクセスします。	146 ページの「USB カメラを追加する方法」
	`vold` を使用しないで USB 大容量ストレージデバイスを取り外します。	148 ページの「`vold` を使用しないで USB 大容量ストレージデバイスを取り外す方法」

作業	説明	参照先
非準拠 USB 大容量ストレージデバイスを追加します。	`scsa2usb.conf` にエントリを追加する方法で、非準拠 USB 大容量ストレージデバイスを追加します。	144 ページの「互換性のない USB 大容量ストレージデバイスの使用方法」
USB 大容量ストレージデバイスを使用できるように準備します。 USB デバイス情報を表示します。	`vold` を使用して USB 大容量ストレージデバイスを使用できるように準備します。 USB デバイス情報を表示します。	148 ページの「`vold` を使用した USB 大容量ストレージデバイスの使用の準備」 149 ページの「USB デバイス情報を表示する方法」
USB 大容量ストレージデバイス上にファイルシステムを作成します。	デバイスにデータを格納する前に、デバイス上にファイルシステムを作成する必要があります。	150 ページの「USB 大容量ストレージデバイス上にファイルシステムを作成する方法 」 152 ページの「`vold` を使用しないで USB 大容量ストレージデバイス上にファイルシステムを作成する方法」
USB 大容量ストレージデバイス上で、パーティションを変更してファイルシステムを作成します。	USB 大容量ストレージデバイス上にファイルシステムを作成する前に、既存のパーティションの変更が必要な場合があります。	154 ページの「`vold` を使用しないで USB 大容量ファイルシステム上でパーティションを変更し、PCFS ファイルシステムを作成する方法」
USB 大容量ストレージデバイス上に Solaris パーティションを作成してスライスを変更します。	`rmformat` を使用してスライス情報を変更する前に、スライス情報を持つファイルを作成する必要があります。	157 ページの「USB 大容量ストレージデバイス上で `vold` を使用しないで Solaris パーティションの作成およびスライスの変更を行う方法」
`vold` を使用しないで USB 大容量ストレージデバイスをマウントします。	`vold` を使用しないで USB 大容量ストレージデバイスをマウントします。	159 ページの「`vold` を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」
(省略可能) USB デバイスドライバを無効にします。	システムで USB サポートを必要としない場合は、USB デバイスドライバを無効にします。	162 ページの「特定の USB ドライバを無効にする方法」
(省略可能) 未使用の USB デバイスリンクを削除します。	`devfsadm` コマンドを使って、USB デバイスのリンクを削除します。	162 ページの「使用されていない USB デバイスのリンクを削除する方法」

USB 大容量ストレージデバイスの使用

Solaris 9 リリースから、次の USB リムーバブル大容量ストレージデバイスがサポートされるようになりました。

- CD-RW
- ハードディスク
- DVD
- デジタルカメラ
- フロッピーディスクデバイス
- SmartMedia および CompactFlash デバイス

Oracle Solaris ZFS ファイルシステムで USB 大容量ストレージデバイスを使用する方法については、122 ページの「USB デバイスの新機能」を参照してください。

Oracle Solaris OS でサポートされるすべての USB デバイスを確認するには、次の Web サイトを参照してください。

`http://www.sun.com/io_technologies/USB.html`

以前の Solaris リリースでは、すべての USB ストレージデバイスはリムーバブルメディアとして識別されていました。これにより、自動マウントをはじめ、次に示す利点の大部分が実現されていました。Solaris 10 6/06 リリースでは、USB 大容量ストレージデバイスがホットプラグ対応デバイスとして認識されますが、次に示すような USB リムーバブルメディアの利点も得られます。ホットプラグの動作の詳細は、71 ページの「USB と 1394 (FireWire) サポートの機能拡張」を参照してください。

- Solaris 10 1/06 リリース以降、ホットプラグ対応デバイスは自動的にマウントされます。詳細は、124 ページの「`vold` によるホットプラグ対応 USB デバイスの認識」を参照してください。
- 標準の MS-DOS または Windows (FAT) ファイルシステムを使用する USB ストレージデバイスがサポートされます。
- 使いやすい `rmformat` コマンドを使用して、スライスを作成できます。`fdisk` コマンドを使用して USB デバイスのパーティションを作成できますが、`format` ユーティリティーまたは `rmformat -F` コマンドを使用して USB デバイスを物理的にフォーマットしないでください。
- `rmformat` コマンドを使用して、メディアが挿入されているすべての USB デバイスを表示します。例については、149 ページの「USB デバイス情報を表示する方法」を参照してください。
- `mount` コマンドが必要なくなったため、ルート以外のユーザーでも USB ストレージデバイスにアクセスできます。デバイスは自動的にマウントされ、`/rmdisk` ディレクトリの下で利用できます。システムの停止中に新しいデバイスを接続した場合は、`boot -r` コマンドを使用して再構成ブートを実行し、そのデバイスを認識させます。

- これらのデバイスは、リムーバブルメディアサービスを実行している場合でも実行していない場合でも管理できます。
- FAT ファイルシステムを持つディスクをマウントし、アクセスできるようになりました。次に例を示します。

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

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

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

USB フロッピーディスクデバイスの使用

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

USB フロッピーディスクデバイスは SCSI リムーバブルメディアデバイスとして認識されます。デバイスは /rmdisk ディレクトリの下で利用できます。

USB フロッピーディスクデバイスの使用方法の詳細は、第 1 章「リムーバブルメディアの管理 (概要)」を参照してください。

非準拠 USB 大容量ストレージデバイスの使用

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

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

推奨される USB 大容量ストレージデバイスについては、次の Web サイトを参照してください。

http://www.sun.com/io_technologies/USB.html

詳細は、scsa2usb(7D) のマニュアルページを参照してください。

▼ 互換性のない USB 大容量ストレージデバイスの使用方法

1. スーパーユーザーになります。
2. /kernel/drv/scsa2usb.conf ファイルにエントリを追加します。
   次のエントリは、USB メモリースティック用です。

   ```
   attribute-override-list = "vid=* reduced-cmd-support=true";
   ```

3. システムをリブートするか、次の操作を実行します。
   a. デバイスを取り外します。
   b. ドライバを手動で更新します。

      ```
      # update_drv -f scsa2usb
      ```

   c. デバイスを追加します。

USB 大容量ストレージデバイスのホットプラグ

デバイスのホットプラグとは、オペレーティングシステムをシャットダウンすることなくあるいはシステムの電源を切ることなく、デバイスを追加したり取り外したりすることを指します。USB デバイスはすべてホットプラグ対応です。

リムーバブルメディアマネージャーがホットプラグ対応のデバイスを認識するようになりました。デバイスをプラグインするだけで、数秒でマウントされます。何も起こらない場合はデバイスがマウントされているどうかを確認してください。

リムーバブルメディアサービスが動作していることを確認します。

```
# svcs volfs
STATE          STIME    FMRI
online         10:39:12 svc:/system/filesystem/volfs:default

# svcs hal dbus rmvolmgr
STATE          STIME    FMRI
online         May_03   svc:/system/dbus:default
online         May_03   svc:/system/hal:default
online         May_03   svc:/system/filesystem/rmvolmgr:default
```

デバイスが有効で認識されている場合は、デバイスからファイルシステムをマウントできます。

マウントに失敗した場合は、`vold` を停止します。

```
# svcadm disable volfs
```

デバイス上のファイルシステムが自動的にマウントされない場合は、手動によるマウントを試みます。

デバイスをホットリムーブする前に、`eject -l` コマンドの別名からそのデバイスの名前を探します。次に、デバイスのメディアを取り出します。この処理を行わない場合でも、デバイスが解放されてポートが使用できる状態に戻りますが、デバイス上のファイルシステムが破損する場合があります。

USB デバイスは、ホットプラグを実行するとすぐにシステムのデバイス階層に表示されます (`prtconf` コマンドで確認可能)。USB デバイスを使用していないときにそのデバイスを取り外すと、システムのデバイス階層からそのデバイスが削除されます。

デバイスを取り外したときにそのデバイスを使用していた場合、デバイスノードは残りますが、このデバイスを制御しているドライバはデバイス上のすべての動作を停止します。このデバイスに発行されるすべての新しい入出力動作はエラーを返します。

このような状況で、 システムは元のデバイスを差し込むように要求します。デバイスが使用できない場合は、アプリケーションを停止してください。数秒後に、ポートが再び使用できるようになります。

注 – 動作中の、つまり開いているデバイスを削除すると、データの整合性が損なわれる可能性があります。デバイスを取り外す前には、必ずデバイスを閉じるようにしてください。ただし、接続されているキーボードとマウスは例外で、動作中でも移動することができます。

▼ vold を使用しないで USB 大容量ストレージデバイスを追加する方法

1. リムーバブルメディア管理を無効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。

2. vold が動作していないことを確認します。

```
# svcs volfs
STATE          STIME    FMRI
disabled       10:39:12 svc:/system/filesystem/volfs:default
```

3. USB 大容量ストレージデバイスを接続します。

4. USB デバイスが追加されたことを確認します。

   非 USB ストレージデバイスのデバイスリンクの間にあるかもしれない USB ディスクのデバイスリンクを、次のようにして確認します。

```
$ rmformat
Looking for devices...
1. Logical Node: /dev/rdsk/c3t0d0p0
   Physical Node: /pci@0,0/pci108e,4131@2,1/storage@4/disk@0,0
   Connected Device: USB2.0   Flash Disk       2.00
   Device Type: Removable
```

▼ USB カメラを追加する方法

カメラのメディアが PCFS ファイルシステムを使用している場合は、自動的にマウントされます。デバイスが scsa2usb ドライバにバインドされない場合は、libusb アプリケーションを使用して写真を転送します。詳細は、/usr/sfw/share/doc/libusb/libusb.txt を参照してください。

1. スーパーユーザーになります。

2. USB カメラを差し込んで電源を入れます。

   システムによって、カメラ用の論理デバイスが作成されます。カメラが差し込まれると、/var/adm/messages ファイルにメッセージが出力され、デバイスの接続が確認されます。システムでは、カメラはストレージデバイスとして扱われます。

3. /var/adm/messages ファイルの出力を確認します。

```
# more /var/adm/messages
```

出力表示を確認すると、どの論理デバイスが作成されたかを確認でき、そのデバイスを使用してイメージにアクセスできます。出力表示は次のようになります。

```
Jul 15 09:53:35 buffy usba: [ID 349649 kern.info]    OLYMPUS, C-3040ZOOM,
 000153719068
Jul 15 09:53:35 buffy genunix: [ID 936769 kern.info] scsa2usb1 is
/pci@0,0/pci925,1234@7,2/storage@2
Jul 15 09:53:36 buffy scsi: [ID 193665 kern.info] sd3 at scsa2usb1:
target 0 lun 0
```

次のコマンドを実行して、デバイスをマウント可能な /dev/dsk リンクエントリに関連付けます。

```
# ls -l /dev/dsk/c*0 | grep /pci@0,0/pci925,1234@7,2/storage@2
 lrwxrwxrwx   1 root     root          58 Jun 14  2010 c3t0d0p0 ->
 ../../devices/pci@0,0/pci925,1234@7,2/storage@2/disk@0,0:a
```

4 **USB カメラファイルシステムをマウントします。**

ほとんどの場合、カメラのファイルシステムは PCFS ファイルシステムです。ファイルシステムが PCFS の場合は、自動的にマウントされます。

x86 システム上でファイルシステムを手動でマウントするには、次のような構文を使用します。

```
# mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

SPARC システム上でファイルシステムを手動でマウントするには、次のような構文を使用します。

```
# mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
```

ファイルシステムのマウント方法については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

さまざまな PCFS ファイルシステムのマウント方法については、mount_pcfs(1M) のマニュアルページを参照してください。

5 **イメージファイルが使用可能であることを確認します。**

次に例を示します。

```
# ls /mnt/DCIM/100OLYMP/
P7220001.JPG*  P7220003.JPG*  P7220005.JPG*
P7220002.JPG*  P7220004.JPG*  P7220006.JPG*
```

6 **USB カメラが作成したイメージファイルを表示します。**

次に例を示します。

```
# /usr/dt/bin/sdtimage P7220001.JPG &
```

7 **カメラを切り離す前に、ファイルシステムをマウント解除します。**

次に例を示します。

```
# umount /mnt
```

8 (省略可能) カメラの電源をオフにし、切り離します。

▼ vold を使用しないで USB 大容量ストレージデバイスを取り外す方法

この手順は、リムーバブルメディアサービスが無効になっている状態で実行します。

1 スーパーユーザーになります。

2 リムーバブルメディアサービスを無効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。

3 リムーバブルメディアサービスが動作していないことを確認します。

```
# svcs volfs
STATE          STIME    FMRI
disabled       Sep_29   svc:/system/filesystem/volfs:default
```

4 そのデバイスを使用しているアプリケーションを実行中の場合は、そのアプリケーションを停止します。

5 デバイスをマウント解除します。

6 デバイスを取り外します。

vold を使用した USB 大容量ストレージデバイスの使用の準備

リムーバブルメディア上の情報にアクセスするには、リムーバブルメディアサービスを使用する方法と使用しない方法があります。GNOME のファイルマネージャーを使用してリムーバブルメディア上の情報にアクセスする方法については、GNOME デスクトップのマニュアルを参照してください。

USB デバイスのフォーマットが終了すると、通常は `/rmdisk/`*label* ディレクトリの下にマウントされます。

デバイスノードは `/vol/dev` ディレクトリの下に作成されます。リムーバブルストレージデバイスの構成の詳細は、`rmmount.conf(4)` または `vold.conf(4)` のマニュアルページを参照してください。

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

リムーバブルメディアサービスでデバイスを認識できない場合は、デバイスを手動でマウントしてみます。まず rmformat コマンドを使用してデバイスのパスを特定し、次にスーパーユーザーとして mount コマンドを使用してデバイスを手動でマウントします。

デバイスを手動でマウントした場合は、umount コマンドをスーパーユーザーとして使ってマウントを解除します。

USB デバイスのマウントおよびマウント解除の方法については、159 ページの「vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」を参照してください。

▼ USB デバイス情報を表示する方法

- **USB デバイスに関する情報を表示します。**

  たとえば、prtconf コマンドを使用して USB デバイス情報を表示します。この例の prtconf の出力は、USB デバイス情報のみを表示するように省略されています。

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

  たとえば、rmformat コマンドを使用して USB デバイス情報を表示します。

```
$ rmformat
Looking for devices...
     1. Volmgt Node: /vol/dev/aliases/cdrom0
        Logical Node: /dev/rdsk/c0t2d0s2
        Physical Node: /pci@1f,0/pci@1,1/ide@3/sd@2,0
        Connected Device:    LG    CD-ROM CRD-8322B 1.03
        Device Type: CD Reader
```

▼ USB 大容量ストレージデバイス上にファイルシステムを作成する方法

USB フロッピーディスクは、ファイルシステムを追加する前にフォーマットする必要があります。その他のすべての USB 大容量ストレージデバイスを使用するには、PCFS または UFS ファイルシステムが必要です。USB デバイスをフォーマットするときには、次の点に注意してください。

- USB フロッピーディスク以外で `rmformat -F` を使用しないでください。
- デフォルトのスライスを使用しない場合は、`rmformat -s` コマンドを使用してスライスを作成します。必要に応じて、`fdisk` ユーティリティーを使用して USB デバイスにパーティションを作成します。手順については、次を参照してください。
  - 154 ページの「`vold` を使用しないで USB 大容量ファイルシステム上でパーティションを変更し、PCFS ファイルシステムを作成する方法」
  - 157 ページの「USB 大容量ストレージデバイス上で `vold` を使用しないで Solaris パーティションの作成およびスライスの変更を行う方法」
  - USB デバイスが自動的にマウントされている場合は、USB デバイス上にファイルシステムを作成する前に、マウント解除する必要があります。

注 – 手順 4 と 5 は、USB 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にのみ実行してください。

---

1. スーパーユーザーになります。
2. **USB** デバイスをシステムに追加します。**USB** デバイスのホットプラグを実行する方法については、次の節を参照してください。
   - 145 ページの「USB 大容量ストレージデバイスのホットプラグ」
   - 167 ページの「`cfgadm` コマンドを使用した USB デバイスのホットプラグ (作業マップ)」
3. **(省略可能) USB** デバイスを特定します。

   次に例を示します。

   ```
   # cd /dev/rdsk
   # ls -l c*0 | grep usb
   lrwxrwxrwx   1 root   root   55 Mar  5 10:35 c2t0d0s0 ->
   ../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
   ```

   この例では、フロッピーディスクデバイスは `c2t0d0s0` です。
4. 必要に応じて、フロッピーディスクをフロッピーディスクドライブに挿入します。

5 必要に応じて、フロッピーディスクをフォーマットします。

```
% rmformat -F long raw-device
```

SPARC システムの場合、たとえば次のようになります。

```
% rmformat -F long /dev/rdsk/c2t0d0s2
```

x86 システムの場合、たとえば次のようになります。

```
% rmformat -F long /dev/rdsk/c3t0d0p0
```

6 ファイルシステムのタイプを判別し、デバイスがマウント解除されていることを確認します。次のいずれかの手順に従います。

USB デバイスのマウント解除の方法については、159 ページの「vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法」を参照してください。

- PCFS ファイルシステムを作成します。

  ```
  # mkfs -F pcfs -o nofdisk,size=size raw-device
  ```

  -size オプションを 512 バイトブロック単位で指定します。

  次の例は、SPARC システムにおいて 1.4M バイトのフロッピーディスクに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs /dev/rdsk/c2t0d0p0
  Construct a new FAT file system on /dev/rdsk/c2t0d0p0: (y/n)? y
  ```

  次の例は、x86 システムにおいて 1.4M バイトのフロッピーディスクに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs /dev/rdsk/c2t0d0s2
  Construct a new FAT file system on /dev/rdsk/c2t0d0s2: (y/n)? y
  ```

  次の例は、SPARC システムにおいて 100M バイトの USB メモリースティックに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs /dev/rdsk/c5t0d0s2:c
  ```

  次の例は、x86 システムにおいて 100M バイトの USB メモリースティックに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs /dev/rdsk/c5t0d0p0:c
  ```

  このコマンドの実行には、数分かかることがあります。

- UFS ファイルシステムを作成します。

  ```
  # newfs raw-device
  ```

  次に例を示します。

  ```
  # newfs /dev/rdsk/c4t0d0s7
  ```

大容量の USB ハードディスクの場合は、`newfs -f 4096` オプションまたは `newfs -T` オプションの使用を検討してください。

注 – フロッピーディスクの記憶容量は少ないので、UFS ファイルシステムのためにフロッピーディスクのかなりの容量が消費されます。

USB 大容量ストレージデバイス上での PCFS ファイルシステムの作成とスライスの変更の詳しい例については、次の 2 つの手順を参照してください。

▼ `vold` を使用しないで USB 大容量ストレージデバイス上にファイルシステムを作成する方法

リムーバブルメディアサービスが無効になっている状態で USB 大容量ストレージデバイスにファイルシステムを追加するには、次の手順を実行します。

USB フロッピーディスクは、ファイルシステムを追加する前にフォーマットする必要があります。その他のすべての USB 大容量ストレージデバイスを使用するには、PCFS または UFS ファイルシステムが必要です。USB フロッピーディスクをフォーマットするときには、次の点に注意してください。

- USB フロッピーディスク以外で `rmformat -F` を使用しないでください。
- デフォルトのスライスを使用しない場合は、`rmformat -s` コマンドを使用してスライスを作成します。必要に応じて、`fdisk` ユーティリティーを使用して USB デバイスにパーティションを作成します。手順については、次を参照してください。
  - 154 ページの「`vold` を使用しないで USB 大容量ファイルシステム上でパーティションを変更し、PCFS ファイルシステムを作成する方法」
  - 157 ページの「USB 大容量ストレージデバイス上で `vold` を使用しないで Solaris パーティションの作成およびスライスの変更を行う方法」

注 – 手順 7 と 8 は、USB 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にのみ実行してください。

---

1. スーパーユーザーになります。
2. リムーバブルメディアサービスを無効にする方法については、**48 ページの「リムーバブルメディアサービスを無効または有効にする方法」**を参照してください。
3. リムーバブルメディアサービスが動作していないことを確認します。

```
# svcs volfs
STATE          STIME    FMRI
disable        10:39:12 svc:/system/filesystem/volfs:default
```

4 **USB デバイスをシステムに追加します。**

USB デバイスのホットプラグを実行する方法については、次を参照してください。

- 145 ページの「USB 大容量ストレージデバイスのホットプラグ」
- 167 ページの「cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)」

5 **(省略可能) USB デバイスを特定します。**

次の例では、USB フロッピーディスクデバイスを特定します。

```
# cd /dev/rdsk
# ls -l c*0 | grep usb
lrwxrwxrwx   1 root   root    55 Mar  5 10:35 c2t0d0s0 ->
../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
```

この例では、フロッピーディスクデバイスは `c2t0d0s0` です。

6 **必要に応じて、フロッピーディスクをフロッピーディスクドライブに挿入します。**

7 **必要に応じて、フロッピーディスクをフォーマットします。**

```
% rmformat -F long raw-device
```

SPARC システムの場合、たとえば次のようになります。

```
% rmformat -F long /dev/rdsk/c2t0d0s2
```

x86 システムの場合、たとえば次のようになります。

```
% rmformat -F long /dev/rdsk/c3t0d0p0
```

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

- PCFS ファイルシステムを作成します。

  ```
  # mkfs -F pcfs -o nofdisk,size=size raw-device
  ```

  `-size` オプションを 512 バイトブロック単位で指定します。

  次の例は、SPARC システムにおいて 1.4M バイトのフロッピーディスクに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs -o nofdisk,size=2880 /dev/rdsk/c4t0d0s2
  ```

  次の例は、x86 システムにおいて 1.4M バイトのフロッピーディスクに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs -o nofdisk,size=2880 /dev/rdsk/c4t0d0p0
  ```

  次の例は、SPARC システムにおいて 100M バイトの USB メモリースティックに PCFS ファイルシステムを作成する方法を示しています。

  ```
  # mkfs -F pcfs /dev/rdsk/c5t0d0s2:c
  ```

次の例は、x86 システムにおいて 100M バイトの USB メモリースティックに PCFS ファイルシステムを作成する方法を示しています。

```
# mkfs -F pcfs /dev/rdsk/c5t0d0p0:c
```

このコマンドの実行には、数分かかることがあります。

- UFS ファイルシステムを作成します。

  ```
  # newfs raw-device
  ```

  次に例を示します。

  ```
  # newfs /dev/rdsk/c4t0d0s7
  ```

  大容量の USB ハードディスクの場合は、`newfs -f 4096` オプションまたは `newfs -T` オプションの使用を検討してください。

注 – フロッピーディスクの記憶容量は少ないので、UFS ファイルシステムのためにフロッピーディスクのかなりの容量が消費されます。

USB 大容量ストレージデバイス上での PCFS ファイルシステムの作成とスライスの変更の詳しい例については、次の 2 つの手順を参照してください。

▼ vold を使用しないで USB 大容量ファイルシステム上でパーティションを変更し、PCFS ファイルシステムを作成する方法

次の手順では、リムーバブルメディアサービスが無効になっている状態で、USB デバイス上で既存のパーティションを削除し、新しいパーティションを作成してから PCFS ファイルシステムを作成する方法を説明します。この作業を実行する前にデータをバックアップしてください。

1. スーパーユーザーになります。
2. リムーバブルメディアサービスを無効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。
3. `fdisk` ユーティリティーを起動します。

   次に例を示します。

   ```
   # fdisk /dev/rdsk/c3t0d0p0
   ```

4 オプション**3**を選択してパーティションを削除します。

次に例を示します。

```
      Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks

                                                 Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1       Active    Solaris2          1    28      28     97

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection: 3
```

5 削除するパーティションの番号を選択します。

次に例を示します。

```
      Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks

                                                 Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1       Active    Solaris2          1    28      28     97

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Specify the partition number to delete (or enter 0 to exit): 1

Partition deleted.
```

6 パーティションを作成します。

次に例を示します。

```
      Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks

                                                 Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
```

```
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection: 1
```

7 パーティションタイプとして**FAT32**を選択します。

```
      Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks

                                                Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Select the partition type to create:
   1=SOLARIS2  2=UNIX        3=PCIXOS     4=Other
   5=DOS12     6=DOS16       7=DOSEXT     8=DOSBIG
   9=DOS16LBA  A=x86 Boot    B=Diagnostic C=FAT32
   D=FAT32LBA  E=DOSEXTLBA   F=EFI        0=Exit? c
```

8 このパーティションに使用するディスクの割合を指定します。

```
       Total disk size is 29 cylinders
              Cylinder size is 2048 (512 byte) blocks

                                                Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Select the partition type to create:
Specify the percentage of disk to use for this partition (or type "c" to
specify the size in cylinders). 100
```

9 新しいパーティションをアクティブなパーティションにするか非アクティブなパーティションにするかを選択します。

```
             Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks
```

```
                                                  Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Select the partition type to create:
Should this become the active partition? If yes, it  will be activated
each time the computer is reset or turned on.
Please type "y" or "n". n
```

10 ディスク構成を更新して終了します。

```
      Total disk size is 29 cylinders
             Cylinder size is 2048 (512 byte) blocks

                                                  Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 Win95 FAT32       1    28      28     97

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection: 5
```

11 このパーティション上に **PCFS** ファイルシステムを作成します。

次に例を示します。

```
# mkfs -F pcfs -o fat=32 /dev/rdsk/c3t0d0p0:c
Construct a new FAT file system on /dev/rdsk/c3t0d0p0:c: (y/n)? y
```

▼ USB 大容量ストレージデバイス上で vold を使用しないで Solaris パーティションの作成およびスライスの変更を行う方法

次の手順では、リムーバブルメディアサービスが無効になっている状態で、Solaris パーティションを作成してスライスを変更する方法を説明します。

この作業を実行する前にデータをバックアップしてください。

1 スーパーユーザーになります。

2 リムーバブルメディアサービスを無効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。

3 fdisk ユーティリティーを起動します。

次に例を示します。

```
# fdisk /dev/rdsk/c5t0d0s2
No fdisk table exists. The default partition for the disk is:

  a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type "n" to edit the
 partition table.
y
```

4 現在のスライスを表示します。

次に例を示します。

```
# prtvtoc /dev/rdsk/c5t0d0s2
* /dev/rdsk/c5t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*    5836 cylinders
*    5836 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      0    00          0  93755340  93755339
       2      0    00          0  93755340  93755339
```

5 スライスの情報を含むテキストファイルを作成します。

次に例を示します。

```
slices: 0 = 0, 5GB, "wm", "home" :
        1 = 8225280000, 6GB :
        2 = 0, 44GB, "wm", "backup" :
        6 = 16450560000, 15GB
```

各スライスがシリンダ境界から始まっていることを確認します。たとえば、スライス 1 は 822280000 バイトから始まっています。この数は、シリンダサイズ (バイト数) の 1000 倍になります。

詳細は、rmformat(1) の -s オプションの説明を参照してください。

6 上記の手順で作成したスライスファイルを含むスライスを作成します。

次に例を示します。

```
# rmformat -s slice_file /dev/rdsk/c5t0d0s2
```

7 新しいスライスの情報を表示します。

次に例を示します。

```
# prtvtoc /dev/rdsk/c5t0d0s2
* /dev/rdsk/c5t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*    5836 cylinders
*    5836 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*    10485760   5579240  16064999
*    28647912   3482088  32129999
*    63587280  30168060  93755339
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      8    00          0  10485760  10485759
       1      3    01   16065000  12582912  28647911
       2      5    00          0  92274688  92274687
       6      4    00   32130000  31457280  63587279
```

▼ vold を使用しないで USB 大容量ストレージデバイスをマウントまたはマウント解除する方法

次の手順では、リムーバブルメディアサービスが無効になっている状態で、USB 大容量ストレージデバイスをマウントまたはマウント解除する方法を説明します。

1 スーパーユーザーになります。

2 リムーバブルメディアサービスを無効にする方法については、48 ページの「リムーバブルメディアサービスを無効または有効にする方法」を参照してください。

3 リムーバブルメディアサービスが動作していないことを確認します。

```
# svcs volfs
STATE          STIME    FMRI
disabled       10:39:12 svc:/system/filesystem/volfs:default
```

4 (省略可能) デバイスを特定します。

次に例を示します。

```
# cd /dev/rdsk
# ls -l c*0 | grep usb
lrwxrwxrwx   1 root  root   55 Mar  5 10:35 c2t0d0s0 ->
../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
```

この例では、フロッピーディスクデバイスは `c2t0d0s0` です。

5 次のいずれかの方法を選択して、**USB** 大容量ストレージデバイスをマウントまたはマウント解除します。

- USB 大容量ストレージデバイスをマウントします。

  ```
  # mount [ -F fstype ] block-device mount-point
  ```

  次の例は、UFS ファイルシステムを使用しているデバイスのマウント方法を示しています。

  ```
  # mount /dev/dsk/c1t0d0s2 /mnt
  ```

  次の例は、SPARC システムにおいて、PCFS ファイルシステムを使用しているデバイスをマウントする方法を示しています。

  ```
  # mount -F pcfs /dev/dsk/c3t0d0s2:c /mnt
  ```

  次の例は、x86 システムにおいて、PCFS ファイルシステムを使用しているデバイスをマウントする方法を示しています。

  ```
  # mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
  ```

  次の例は、読み取り専用の HSFS ファイルシステムを使用している CD のマウント方法を示しています。

  ```
  # mount -F hsfs -o ro /dev/dsk/c1t0d0s2 /mnt
  ```

- USB 大容量ストレージデバイスをマウント解除します。

  デバイス上のファイルシステムを使用しているユーザーがいないことを最初に確認してください。

  次に例を示します。

  ```
  # fuser -c -u /mnt
  # umount /mnt
  ```

6 **CD** またはフロッピーディスクデバイスの場合には、デバイスを取り出します。

```
# eject /dev/[r]dsk/cntndnsn
```

次に例を示します。

```
# eject /dev/rdsk/c1t0d0s2
```

USB 大容量ストレージデバイスの問題の障害追跡

USB 大容量ストレージデバイスの追加または取り外しで問題が発生した場合には、次のヒントを参考にしてください。

/var/adm/messages ファイルでデバイスの列挙の失敗を確認します。列挙の失敗の場合は、USB ハブを挿入するか、ハブを取り外してルート USB ハブに直接接続します。

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

  ```
  ok boot -r
  ```

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

  ```
  # devfsadm
  ```
- 保存停止モードでシステムの電力消費を抑えている場合は、デバイスを移動しないでください。詳細は、137 ページの「SPARC: USB 電源管理」を参照してください。
- アプリケーションがデバイスを使用しているときに、そのデバイスが取り外されて使用できなくなっている場合は、アプリケーションを停止してください。そのデバイスノードが削除されているかどうかを確認するには、 prtconf コマンドを使用します。

特定の USB ドライバを無効にする

特定の種類の USB デバイスを無効にするには、対応するクライアントドライバを無効にします。たとえば、USB プリンタを無効にするには、そのプリンタを使用している usbprn ドライバを無効にします。usbprn を無効にしても、USB ストレージデバイスなどのほかのデバイスには影響しません。

次の表に、USB デバイスの種類の一部とそれらに対応するドライバを示します。

デバイスの種類	無効にするドライバ
Audio Device	usb_ac および usb_as
HID (通常はキーボードとマウス)	hid
ストレージ	scsa2usb
プリンタ	usbprn
シリアル	usbser_edge

システムに接続されている USB デバイスのドライバを無効にすると、次のようなコンソールメッセージが表示されます。

```
usba10: WARNING: usba:    no driver found for device name
```

▼ 特定の USB ドライバを無効にする方法

1. スーパーユーザーになります。
2. ドライバの別名エントリを /etc/system ファイルから除外します。
   たとえば、次の exclude 文を追加して usbprn ドライバを除外します。

   ```
   exclude: usbprn
   ```

3. システムを再起動します。

   ```
   # init 6
   ```

▼ 使用されていない USB デバイスのリンクを削除する方法

システムの電源がオフのときに USB デバイスを取り外した場合には、次の手順を実行します。システムの電源が切断されているときに USB デバイスを取り外すと、存在しないデバイスへのリンクが残る場合があります。

1. スーパーユーザーになります。
2. そのデバイスにアクセスする可能性のあるアプリケーションをすべて閉じます。
3. 特定の USB クラスの未使用のリンクを削除します。
   次に例を示します。

   ```
   # devfsadm -C -c audio
   ```

   または、関連するリンクをすべて削除します。

   ```
   # devfsadm -C
   ```

USB オーディオデバイスの使用 (作業マップ)

作業	説明	参照先
USB オーディオデバイスを追加します。	USB マイクおよびスピーカを追加します。	165 ページの「USB オーディオデバイスを追加する方法」
システムの主オーディオデバイスを特定します。	システムの主オーディオデバイスを特定します。	165 ページの「システムの主オーディオデバイスを識別する方法」
主 USB オーディオデバイスを変更します。	USB オーディオデバイスを取り外したり変更したりする場合、1 つのオーディオデバイスを主オーディオデバイスにすることがあります。	166 ページの「主 USB オーディオデバイスを変更する方法」
使用していない USB デバイスリンクを削除します。	システムの電源がオフのときに USB オーディオデバイスを取り外すと、/dev/audio デバイスが、存在しない /dev/sound/* デバイスを指したままになることがあります。	162 ページの「使用されていない USB デバイスのリンクを削除する方法」
USB オーディオに関する問題の障害追跡を行います。	USB スピーカからサウンドが出力されない場合は、この節を参照します。	166 ページの「USB オーディオデバイスに関する問題の障害追跡」

USB オーディオデバイスの使用

特定の Solaris リリースでの USB オーディオサポートについては、125 ページの「USB デバイスの Solaris サポート」を参照してください。

この Solaris リリースでは、2 つの連携するドライバ、usb_ac および usb_as の実装によって、USB オーディオサポートを提供しています。オーディオコントロールドライバである usb_ac は Solaris USB Architecture 準拠のクライアントドライバで、ユーザーアプリケーションのインタフェースを制御します。オーディオストリーミングドライバである usb_as は、再生中および録音中にオーディオデータメッセージを処理します。また、サンプル周波数と精度を設定し、usb_ac ドライバからの要求を符号化します。どちらのドライバも、USB オーディオクラス 1.0 仕様に準拠しています。

一部のオーディオデバイスでは、ソフトウェアが制御している音量を設定できます。この機能を管理するために、STREAMS モジュールの usb_ah が HID ドライバの先頭に置かれます。

Solaris では、再生専用、録音専用、録音および再生用の USB オーディオデバイスをサポートします。USB オーディオデバイスのホットプラグがサポートされます。

- USB オーディオデバイスは、USB コネクタを備えた SPARC Ultra および x86 プラットフォームでサポートされます。
- Solaris 8 10/01、Solaris 8 2/02、Solaris 9 のいずれかのリリース上で再生または録音するには、USB オーディオデバイスが 44100 Hz または 48000 Hz の固定サンプリングレートをサポートしている必要があります。Solaris 10 リリースでは、44100 Hz または 48000 Hz のサンプリングレートは必要ではなくなりました。
- 完全にサポートされているオーディオデータ形式を確認するには、usb_ac(7D) のマニュアルページを参照してください。

主オーディオデバイスは、/dev/audio です。次のコマンドを使用して、/dev/audio が USB オーディオを指しているかを確認できます。

```
% mixerctl
Device /dev/audioctl:
  Name    = USB Audio
  Version = 1.0
  Config  = external

Audio mixer for /dev/audioctl is enabled
```

USB オーディオデバイスを接続した後、audioplay コマンドおよび audiorecord コマンドを使用し、/dev/sound/*N* デバイスリンクを介してデバイスにアクセスします。

/dev/audio および /dev/sound/N デバイスは、スピーカ、マイク、またはコンボデバイスを参照できます。不正なデバイスタイプを参照すると、そのコマンドは失敗します。たとえば、マイクに対して audioplay を使用しようとすると、そのコマンドは失敗します。

ほとんどの Sun オーディオアプリケーションでは、特定のデフォルトオーディオデバイスを選択できます。たとえば、audioplay や audiorecord の場合には、AUDIODEV シェル変数を設定するか、-d オプションを指定します。ただし、/dev/audio をオーディオファイルとしてハードコードしている他社製のアプリケーションでは AUDIODEV は動作しません。

USB オーディオデバイスを差し込むと、/dev/audio が使用中でない限り、自動的にそれが主オーディオデバイス /dev/audio になります。オンボードのオーディオから USB オーディオへ、および USB オーディオからオンボードのオーディオへ /dev/audio を変更する方法については、166 ページの「主 USB オーディオデバイスを変更する方法」および usb_ac(7D) のマニュアルページを参照してください。

複数の USB オーディオデバイスのホットプラグ

USB オーディオデバイスがシステムに差し込まれると、それが主オーディオデバイス /dev/audio になります。そのデバイスは、システムをリブートした後でも、主オーディオデバイスのままです。USB オーディオデバイスが追加で差し込まれた場合、最後に差し込まれたデバイスが主オーディオデバイスになります。

USB オーディオデバイスに関する問題の障害追跡について、さらに詳しい情報を得るには、usb_ac(7D) のマニュアルページを参照してください。

▼ USB オーディオデバイスを追加する方法

1. **USB スピーカを差し込みます。**

   主オーディオデバイス /dev/audio は、USB スピーカを指します。

   ```
   % ls -l /dev/audio
   lrwxrwxrwx   1 root     root    10 Feb 13 08:46 /dev/audio -> usb/audio0
   ```

2. **(省略可能) スピーカを取り外します。その後、再度差し込みます。**

   スピーカを取り外すと、/dev/audio デバイスがオンボードのオーディオに戻ります。

   ```
   % ls -l /dev/audio
   lrwxrwxrwx   1 root     root   7 Feb 13 08:47 /dev/audio -> sound/0
   ```

3. **USB マイクを追加します。**

   ```
   % ls -l /dev/audio
   lrwxrwxrwx   1 root     root    10 Feb 13 08:54 /dev/audio -> usb/audio1
   ```

▼ システムの主オーディオデバイスを識別する方法

この手順は、すでに USB オーディオデバイスが接続されていることを前提としています。

- システムの新しいオーディオリンクを調べます。
  - ls コマンドを使用して、システムの新しいオーディオリンクを表示します。

    次に例を示します。

    ```
    % ls -lt /dev/audio*
    lrwxrwxrwx   1 root  root      7 Jul 23 15:46 /dev/audio -> usb/audio0
    lrwxrwxrwx   1 root  root     10 Jul 23 15:46 /dev/audioctl ->
    usb/audioctl0/
    % ls -lt /dev/sound/*
    lrwxrwxrwx   1 root  root     74 Jul 23 15:46 /dev/sound/1 ->
    ../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...
    ```

```
lrwxrwxrwx   1 root  root     77 Jul 23 15:46 /dev/sound/1ctl ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...
lrwxrwxrwx   1 root  other    66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root  other    69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl
%
```

主オーディオデバイス `/dev/audio` が、新しく差し込まれた USB オーディオデバイスの `/dev/usb/audio0` を指していることがわかります。

- `prtconf` コマンドを使用して USB デバイス情報を参照して、システム上の USB オーディオデバイスを調べることもできます。

```
% prtconf
.
.
.
usb, instance #0
   hub, instance #0
        mouse, instance #0
        keyboard, instance #1
        device, instance #0
            sound-control, instance #0
            sound, instance #0
            input, instance #0
.
.
.
```

▼ 主 USB オーディオデバイスを変更する方法

- 主 USB オーディオデバイスを変更するには、次のいずれかの手順を選択します。
  - オンボードのオーディオデバイスを主オーディオデバイスにするには、USB オーディオデバイスを取り外してください。すると、`/dev/audio` リンクは `/dev/sound/0` エントリを指すはずです。`/dev/sound/0` エントリが主オーディオデバイスでない場合は、システムをシャットダウンして `boot -r` コマンドを実行するか、`devfsadm -i` コマンドをスーパーユーザーとして実行してください。
  - USB オーディオデバイスを主オーディオデバイスにするには、そのデバイスを差し込んでデバイスリンクを確認するだけです。

USB オーディオデバイスに関する問題の障害追跡

ドライバを適用し、音量も上げているのに、USB スピーカから音が出ないことがあります。デバイスのホットプラグを実行してもこの動作が変わらないことがあります。

この問題を解決するには、USB スピーカの電源を再投入します。

オーディオデバイスの所有権に関する注意事項

オーディオデバイスを操作するときは、オーディオデバイスの所有権に関する、次に挙げる点に注意してください。

- USB オーディオデバイスを差し込む時にコンソールからログインしていると、コンソールが /dev/* エントリの所有者になります。つまり、コンソールでログインしている限り、オーディオデバイスを使用できることになります。
- USB オーディオデバイスを差し込むときにコンソールにログインしていない場合、root がそのデバイスの所有者になります。ただし、その後にコンソールにログインして USB オーディオデバイスにアクセスしようとすると、デバイスの所有権はコンソールに変更されます。詳細は、logindevperm(4) のマニュアルページを参照してください。
- リモートから rlogin コマンドでログインして USB オーディオデバイスにアクセスしようとした場合は、所有権は変更されません。たとえば、権限のないユーザーが、ほかの人の所有するマイクを通して行われる会話を聞くことはできません。

cfgadm コマンドを使用した USB デバイスのホットプラグ (作業マップ)

作業	説明	参照先
USB バス情報を表示します。	USB デバイスおよびバスについての情報を表示します。	169 ページの「USB バス情報を表示する方法 (cfgadm)」
USB デバイスの構成を解除します。	システムに物理的に接続されている USB デバイスを論理的に構成解除します。	170 ページの「USB デバイスの構成を解除する方法」
USB デバイスを構成します。	以前に構成を解除した USB デバイスを構成します。	170 ページの「USB デバイスの構成方法」
論理的に USB デバイスの接続を解除します。	物理的にシステムの近くにいないときには、USB デバイスを論理的に切り離すことができます。	171 ページの「論理的に USB デバイスを接続解除する方法」
論理的に USB デバイスに接続します。	以前に論理的に接続解除または構成解除した USB デバイスを論理的に接続します。	171 ページの「論理的に USB デバイスを接続する方法」
USB デバイスのサブツリーの接続を解除します。	ハブの下位階層 (または下位のツリー) である USB デバイスサブツリーの接続を解除します。	172 ページの「論理的に USB デバイスのサブツリーを接続解除する方法」

作業	説明	参照先
USB デバイスをリセットします。	USB デバイスをリセットして、デバイスを論理的に削除し、再作成します。	172 ページの「USB デバイスのリセット方法」
複数の構成を持つ USB デバイスのデフォルト構成を変更します。	複数の構成を持つ USB デバイスのデフォルト構成を変更します。	172 ページの「複数の構成を持つ USB デバイスのデフォルト構成を変更する方法」

cfgadm コマンドを使った USB デバイスのホットプラグ

cfgadm コマンドを使用せずに稼働中のシステムから USB デバイスを追加または削除できます。ただし、USB デバイスは、デバイスを物理的に削除しなくても「論理的に」ホットプラグを実行できます。この方法は、リモートで作業中に機能していない USB デバイスを無効にしたりリセットしたりする必要がある場合に便利です。cfgadm コマンドを使うと、製造元や製品情報を含む USB デバイスツリーを表示することもできます。

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

- 占有装置 (occupant)。USB デバイスなどの、システムに構成可能なハードウェアリソースのことです
- 受容体 (receptacle)。USB ポートなどの、占有装置を受け入れる場所のことです

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

cfgadm コマンドを使用すると、USB デバイスステータス情報を取得できます。

受容体の状態	説明
empty/unconfigured	デバイスが物理的に接続されていません。
disconnected/unconfigured	デバイスは物理的に接続されているかもしれませんが、論理的に接続解除されており利用できません。
connected/unconfigured	デバイスは論理的に接続されていますが利用できません。このデバイスは、prtconf の出力に表示されます。

受容体の状態	説明
connected/configured	デバイスは接続されており利用可能です。

次の節では、ソフトウェアから cfgadm コマンドを使用して USB デバイスのホットプラグを実行する方法について説明します。次のすべてのサンプル USB デバイス情報は、関連した情報に焦点を合わせるために一部省略されています。

▼ USB バス情報を表示する方法 (cfgadm)

prtconf コマンドを使用して USB 構成情報を表示する例については、149 ページの「USB デバイス情報を表示する方法」を参照してください。

1 **USB バス情報を表示します。**

次に例を示します。

```
% cfgadm
Ap_Id                  Type         Receptacle   Occupant     Condition
usb0/4.5               usb-hub      connected    configured   ok
usb0/4.5.1             usb-device   connected    configured   ok
usb0/4.5.2             usb-printer  connected    configured   ok
usb0/4.5.3             usb-mouse    connected    configured   ok
usb0/4.5.4             usb-device   connected    configured   ok
usb0/4.5.5             usb-storage  connected    configured   ok
usb0/4.5.6             usb-communi  connected    configured   ok
usb0/4.5.7             unknown      empty        unconfigured ok
```

前の例で usb0/4.5.1 は、第 2 レベルの外部ハブのポート 1 に接続されているデバイスを識別します。この第 2 レベルハブは第 1 レベルの外部ハブのポート 5 に接続されており、また第 1 レベルのハブは最初の USB コントローラのルートハブであるポート 4 に接続されています。

2 **特定の USB デバイス情報を表示します。**

次に例を示します。

```
% cfgadm -l -s "cols=ap_id:info"
Ap_Id       Information
 usb0/4.5.1  Mfg: Inside Out Networks Product: Edgeport/421 NConfigs: 1
Config: 0  : ...
 usb0/4.5.2  Mfg: <undef> Product: <undef>   NConfigs: 1 Config: 0 ...
 usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1
Config: 0 ...
 usb0/4.5.4  Mfg: NMB  Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0
 usb0/4.5.5  Mfg: Hagiwara Sys-Com  Product: SmartMedia R/W  NConfigs: 1
Config: 0 : ...
 usb0/4.5.6  Mfg: 3Com Inc.  Product: U.S.Robotics 56000 Voice USB Modem
NConfigs: 2 ...
 usb0/4.5.7
```

▼ USB デバイスの構成を解除する方法

システムに物理的に接続されている USB デバイスの構成を解除することはできます。しかし、そのデバイスにドライバを適用することはできません。USB デバイスの構成を解除しても、そのデバイスは prtconf 出力に表示されることに注意してください。

1. スーパーユーザーになります。
2. USB デバイスの構成を解除します。

   次に例を示します。

```
# cfgadm -c unconfigure usb0/4.7
Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7
This operation will suspend activity on the USB device
Continue (yes/no)? y
```

3. デバイスの構成が解除されていることを確認します。

   次に例を示します。

```
# cfgadm
Ap_Id                  Type         Receptacle   Occupant     Condition
usb0/4.5               usb-hub      connected    configured   ok
usb0/4.5.1             usb-device   connected    configured   ok
usb0/4.5.2             usb-printer  connected    configured   ok
usb0/4.5.3             usb-mouse    connected    configured   ok
usb0/4.5.4             usb-device   connected    configured   ok
usb0/4.5.5             usb-storage  connected    configured   ok
usb0/4.5.6             usb-communi  connected    configured   ok
usb0/4.5.7             unknown      empty        unconfigured ok
usb0/4.6               usb-storage  connected    configured   ok
usb0/4.7               usb-storage  connected    unconfigured ok
```

▼ USB デバイスの構成方法

1. スーパーユーザーになります。
2. USB デバイスを構成します。

   次に例を示します。

```
# cfgadm -c configure usb0/4.7
```

3. USB デバイスが構成されていることを確認します。

   次に例を示します。

```
# cfgadm usb0/4.7
Ap_Id                  Type         Receptacle  Occupant    Condition
usb0/4.7               usb-storage  connected   configured  ok
```

▼ 論理的に USB デバイスを接続解除する方法

物理的にシステムの近くにいないときに、システムから USB デバイスを取り外し、prtconf 出力を削除する場合、USB デバイスの接続を論理的に解除できます。デバイスは物理的に接続されたままです。しかし、論理的に接続解除され、使用できなくなり、システムにも表示されません。

1 スーパーユーザーになります。

2 USB デバイスを接続解除します。

次に例を示します。

```
# cfgadm -c disconnect -y usb0/4.7
```

3 デバイスが接続解除されていることを確認します。

次に例を示します。

```
# cfgadm usb0/4.7
Ap_Id                  Type      Receptacle    Occupant      Condition
usb0/4.7               unknown   disconnected  unconfigured  ok
```

▼ 論理的に USB デバイスを接続する方法

次の手順を使用して、以前に論理的に接続解除または構成解除された USB デバイスを論理的に接続します。

1 スーパーユーザーになります。

2 USB デバイスを接続します。

次に例を示します。

```
# cfgadm -c configure usb0/4.7
```

3 デバイスが接続されていることを確認します。

次に例を示します。

```
# cfgadm usb0/4.7
Ap_Id                  Type         Receptacle  Occupant    Condition
usb0/4.7               usb-storage  connected   configured  ok
```

デバイスを利用できるようになり、システムにも表示されるようになります。

▼ 論理的に USB デバイスのサブツリーを接続解除する方法

次の手順を使用して、USB デバイスのサブツリーを接続解除します。サブツリーは、ハブの下位デバイスの階層 (ツリー) です。

1. スーパーユーザーになります。
2. USB デバイスのサブツリーを接続解除します。
   次に例を示します。

```
# cfgadm -c disconnect -y usb0/4
```

3. USB デバイスサブツリーの接続解除を確認します。
   次に例を示します。

```
# cfgadm usb0/4
Ap_Id                  Type       Receptacle   Occupant     Condition
usb0/4                 unknown    disconnected unconfigured ok
```

▼ USB デバイスのリセット方法

USB デバイスでエラーが発生した場合は、cfgadm コマンドを使ってデバイスをリセットします。このコマンドを使うと、デバイスを論理的に削除し、再作成できます。

1. スーパーユーザーになります。
2. デバイスが使用中でないことを確認します。
3. デバイスをリセットします。
   次に例を示します。

```
# cfgadm -x usb_reset -y usb0/4.7
```

4. デバイスが接続されていることを確認します。
   次に例を示します。

```
# cfgadm usb0/4.7
Ap_Id                  Type         Receptacle   Occupant     Condition
usb0/4.7               usb-storage  connected    configured   ok
```

▼ 複数の構成を持つ USB デバイスのデフォルト構成を変更する方法

複数の構成を持つ USB デバイスを操作するときには、次の点を考慮してください。

- USB デバイスの構成には、デバイス自体がどのようにオペレーティングシステムに表示されるかを定義します。この構成方法は、前の節で cfgadm について説明したシステムデバイスの構成方法とは異なります。
- 一部の USB デバイスでは、複数の構成がサポートされますが、一度に有効にできる構成は 1 つだけです。
- 複数の構成を持つデバイスを特定するには、cfgadm -lv の出力を確認します。Nconfigs は、1 より大きい値になります。
- デフォルトの USB 構成は configuration 1 です。現在の構成は、cfgadm -lv の出力に Config として反映されます。
- デフォルトの構成を変更しても、デバイスを同じポートに再接続している間は、デバイスのリブート、ホットリムーブ、および再構成を行なっても、構成の変更は適用されません。

1. デバイスが使用中でないことを確認します。

2. デフォルトの USB 構成を変更します。

   次に例を示します。

```
# cfgadm -x usb_config -o config=2 usb0/4
   Setting the device: /devices/pci@1f,0/usb@c,3:4
   to USB configuration 2
   This operation will suspend activity on the USB device
   Continue (yes/no)? yes
```

3. デバイスの変更を確認します。

   次に例を示します。

```
# cfgadm -lv usb0/4
Ap_Id  Receptacle   Occupant     Condition  Information When  Type
     Busy     Phys_Id
usb0/4 connected    unconfigured ok         Mfg: Sun  2000
Product: USB-B0B0 aka Robotech
With 6 EPPS High Clk Mode   NConfigs: 7  Config: 2  : EVAL Board Setup
unavailable
usb-device   n        /devices/pci@1f,0/usb@c,3:4
```

   Config が「2」になっていることに注目してください。

◆ ◆ ◆ 第 9 章

InfiniBand デバイスの使用 (概要/手順)

InfiniBand (IB) は、スイッチファブリックに基づく入出力テクノロジです。この技術により、入出力デバイスとホストとの接続やホスト間の通信で、帯域幅が広く応答時間の短い相互接続が提供されます。

この章の内容は以下のとおりです。

- 175 ページの「InfiniBand デバイスの概要」
- 178 ページの「IB デバイスの動的再構成 (cfgadm)」

IB デバイスの使用手順については、次の節を参照してください。

- 177 ページの「IB デバイスの動的再構成 (作業マップ)」
- 187 ページの「InfiniBand デバイスでの uDAPL アプリケーションインタフェースの使用」

動的再構成およびホットプラグについての一般的な情報については、第 6 章「デバイスの動的構成 (手順)」を参照してください。

InfiniBand デバイスの概要

IB デバイスは Solaris IB 連結ドライバによって管理されます。このドライバは、次の 5 種類のデバイスをサポートしています。

- IB Port デバイス
- IB 仮想物理接続点 (VPPA) デバイス
- IB HCA サービス (HCA_SVC) デバイス
- 疑似デバイス
- 入出力コントローラ (IOC) デバイス

IB 連結ドライバは、Solaris IB デバイスマネージャー (IBDM) にサービス (このマニュアルでは「通信サービス」と呼ぶ) を照会して、IB Port、HCA_SVC、および IB VPPA デバイスを列挙します。

Port デバイスは、Host Channel Adapter (HCA) の特定のポート番号に通信サービスをバインドします。これに対し、VPPA デバイスは、ポート番号とパーティションキー番号の組み合わせに通信サービスをバインドします。HCA_SVC デバイスは、特定の HCA に通信サービスをバインドします。Port デバイスと HCA_SVC デバイスでは、パーティションキー p_key の値として常に 0 が使用されます。Port、HCA_SVC、および VPPA デバイスは、HCA の子であり、ib.conf ファイルから列挙されます。詳細は、ib(7D) のマニュアルページを参照してください。

IOC デバイスは、IB 連結ドライバの子であり、入出力ユニットの一部です。擬似デバイスも IB 連結ドライバの子です。独自の構成ファイルを持つほかのすべてのデバイスを参照して列挙されます。詳細は、ib(4) のマニュアルページを参照してください。

次の表に、IB デバイスツリーのパス名の形式を示します。

IOC デバイス	/ib/ioc@1730000007F510C,173000007F50
IB 擬似デバイス	/ib/<driver>@<unit-address>
IB VPPA デバイス	/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@<port#>,<p_key>,<service>
IB HCA_SVC デバイス	/pci@1f,2000/pci@1/pci15bc,5a44@0/ibport@0,0,<service>
IB Port デバイス	/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@<port#>,0,<service>
HCA	/pci@1f,2000/pci@1/pci15b3,5a44@0

IB HCA_SVC デバイスの port# および p_key は 0 です。

上記の表で、IB の構成要素はそれぞれ次のものを表します。

<*services*> 通信サービスです。たとえば、ipib は、ibd カーネルクライアントドライバで使用される通信サービスです。

<*p_key*> 使用されるパーティションキーの値です。

<*port*> ポート番号です。

<*unit-address*> IB カーネルクライアントドライバの driver.conf ファイルにこの名前で指定されているプロパティーを参照します。詳細は、driver.conf(4) のマニュアルページを参照してください。

IB デバイスの動的再構成 (作業マップ)

作業	説明	参照先
IB デバイス情報を表示します。	システム上の IB デバイスに関する情報を表示します。	179 ページの「IB デバイス情報を表示する方法」
IOC デバイスを構成または構成解除します。	次のいずれかの手順に従います。 IOC デバイスの構成を解除します。 IOC デバイスを構成します。	 180 ページの「IOC デバイスの構成を解除する方法」 181 ページの「IOC デバイスを構成する方法」
ポートデバイスまたは VPPA デバイスを構成または構成解除します。	次のいずれかの手順に従います。 ポートデバイスまたは VPPA デバイスの構成を解除します。 ポートデバイスまたは VPPA デバイスを構成します。	 181 ページの「IB Port、HCA_SVC、VPPA デバイスの構成を解除する方法」 182 ページの「IB Port、HCA_SVC、VPPA デバイスを構成する方法」
IB 擬似デバイスを構成または構成解除します。	次のいずれかの手順に従います。 IB 擬似デバイスの構成を解除します。 IB 擬似デバイスを構成します。	 183 ページの「IB 擬似デバイスの構成を解除する方法」 183 ページの「IB 擬似デバイスを構成する方法」
HCA のカーネル IB クライアントを表示します。	HCA の構成を解除する場合などに、HCA のカーネル IB クライアントの情報を表示する必要が生じることがあります。	183 ページの「HCA のカーネル IB クライアントを表示する方法」
IB HCA を構成または構成解除します。	次のいずれかの手順に従います。 HCA に接続されている IB デバイスの構成を解除します。 HCA に接続されている IB デバイスを構成します。	 184 ページの「HCA に接続されている IB デバイスの構成を解除する方法」 185 ページの「IB HCA を構成する方法」

作業	説明	参照先
IB p_key テーブルを更新します。	HCA ポートの p_key テーブルの情報が変更された場合は、IBTF と IBDM に通知して内部 p_key データベースを更新する必要があります。	185 ページの「IB p_key テーブルを更新する方法」
IB 通信サービスを表示します。	IBTF で現在使用されている IB 通信サービスを表示します。	185 ページの「IB 通信サービスを表示する方法」
VPPA 通信サービスを追加または削除します。	次のいずれかの手順に従います。 VPPA 通信サービスを追加します。 VPPA 通信サービスを削除します。	 186 ページの「VPPA 通信サービスを追加する方法」 186 ページの「既存の IB Port、HCA_SVC、VPPA 通信サービスを削除する方法」
IOC の構成を更新します。	すべての IOC デバイスノードのプロパティーまたは特定の IOC Ap_Id を更新できます。	187 ページの「IOC の構成を更新する方法」

IB デバイスの動的再構成 (cfgadm)

cfgadm CLI だけを使用して、稼働中のシステムの IB デバイスを構成したり、構成を解除したりできます。このコマンドでは、IB ファブリックの表示、通信サービスの管理、および p_key テーブルデータベースの更新を行うこともできます。詳細は、cfgadm_ib(1M) のマニュアルページを参照してください。

cfgadm CLI は、ホストから見た IB ファブリック全体の動的再構成 (このマニュアルでは「DR」と呼ぶ) を管理します。cfgadm の操作は、Port、VPPA、HCA_SVC、IOC、擬似デバイスなど、すべての IB デバイスでサポートされています。

cfgadm コマンドは接続点 (Ap_Id) についての情報を表示します。「接続点」とは、DR 操作を行うことができるシステム内の特定の場所のことです。cfgadm でサポートされている Ap_Id の詳細は、cfgadm_ib(1M) のマニュアルページを参照してください。IB の Ap_Id は、すべて connected と表示されます。

cfgadm コマンドを使用すると、IB デバイスステータス情報を取得できます。

受容体の状態	説明
connected/configured/ok	デバイスは接続されており利用可能です。devinfo ノードが存在します。

受容体の状態	説明
connected/unconfigured/unknown	デバイスは利用不可で、このデバイスの devinfo ノードやデバイスドライバは存在しません。あるいは、このデバイスは ib 連結ドライバで使用できるように構成されませんでした。IB デバイスマネージャーではこのデバイスが認識されている可能性もあります。

次の節では、cfgadm コマンドを使用して IB デバイスの動的再構成 (DR) を行う方法について説明します。次のすべてのサンプル IB デバイス情報は、関連する情報に焦点を合わせるために一部省略されています。

▼ IB デバイス情報を表示する方法

prtconf コマンドを使用して IB デバイスの一般的な情報を表示できます。たとえば、次のようになります。

```
$ prtconf
    pci, instance #0
        pci15b3,5a44, instance #0
             ibport, instance #253
             ibport, instance #254
             ibport, instance #255
    .
    .
    .
    ib, instance #0
        ioc, instance #243
        ioc, instance #244
        ioc, instance #245
        ioc, instance #246
        ioc, instance #247
        ioc, instance #248
        ibgen, instance #249
```

上記の例で、pci15b3,5a44 は IB HCA を指しています。

特定の IB デバイス情報を表示するには、次の手順を使用します。

1 スーパーユーザーになります。

2 IB ファブリック情報を表示します。
次に例を示します。

```
# cfgadm -a
Ap_Id                       Type       Receptacle Occupant   Condition
ib                          IB-Fabric  connected  configured   ok
hca:1730000008070           IB-HCA     connected  configured   ok
ib::1730000007F5198         IB-IOC     connected  configured   ok
```

```
ib::1730000007F5199          IB-IOC      connected  configured   ok
ib::1730000008070,0,hnfs     IB-HCA_SVC connected  configured   ok
ib::1730000008071,0,sdp      IB-PORT     connected  configured   ok
ib::1730000008072,0,sdp      IB-PORT     connected  configured   ok
ib::1730000008071,8001,ipib IB-VPPA     connected  configured   ok
ib::1730000008072,8001,ipib IB-VPPA     connected  configured   ok
ib::ibgen,0                  IB-PSEUDO   connected  configured   ok
#
```

上記の出力例で、構成要素はそれぞれ次のものを表します。

`Ap_Id ib::1730000008072,0,sdp`	ポート 2 に接続され、sdp サービスにバインドされている IB Port デバイスを識別しています。
`Ap_Id ib::1730000008072,8001,ipib`	ポート 2 に接続され、p_key の値として 0x8001 を使用し、ibd サービスにバインドされている IB VPPA デバイスを識別しています。
`Ap_Id ib:: 1730000008070,0,hnfs`	hnfs サービスにバインドされている IB HCA_SVC デバイスを識別しています。
`Ap_Id ib::1730000007F5198`	IOC デバイスを識別しています。
`Ap_Id ib::ibgen,0`	擬似デバイスを識別しています。

3 特定の **IB** デバイス情報を表示します。

たとえば、IB VPPA デバイスの場合は次のようになります。

```
# cfgadm -al -s "cols=ap_id:info" ib::1730000008072,8001,ipib
Ap_Id                          Information
ib::1730000008072,8001,ipib    ipib
#
```

たとえば、IB HCA デバイスの場合は次のようになります。

```
# cfgadm -al -s "cols=ap_id:info" hca::1730000008070
Ap_Id                          Information
hca::1730000008070             VID: 0x15b3, PID: 0x5a44, #ports: 0x2,
port1 GUID: 0x1730000008071, port2 GUID: 0x1730000008072
#
```

上記の出力には、ポート番号とその GUID がそれぞれ表示されています。

▼ IOC デバイスの構成を解除する方法

システムに物理的に接続されている IB デバイスの構成を解除することはできますが、ドライバを適用することはできません。

1 スーパーユーザーになります。

2 **IB デバイスの構成を解除します。**

次に例を示します。

```
# cfgadm -c unconfigure ib::1730000007F5198
Unconfigure the device: /devices/ib:fabric::1730000007F5198
This operation will suspend activity on the IB device
Continue (yes/no)? y
#
```

3 デバイスの構成が解除されていることを確認します。

次に例を示します。

```
# cfgadm -a ib::1730000007F5198
ib::1730000007F5198         IB-IOC     connected  unconfigured unknown
#
```

▼ IOC デバイスを構成する方法

1 スーパーユーザーになります。

2 **IB デバイスを構成します。**

次に例を示します。

```
# cfgadm -yc configure ib::1730000007F5198
```

3 **IB デバイスが構成されていることを確認します。**

次に例を示します。

```
# cfgadm -al ib::1730000007F5198
Ap_Id                Type    Receptacle  Occupant    Condition
ib::1730000007F5198  IB-IOC  connected   configured  ok
```

▼ IB Port、HCA_SVC、VPPA デバイスの構成を解除する方法

IB Port、HCA_SVC、または VPPA デバイスをシステムから削除するには、次の手順を使用します。

次の例では VPPA デバイスの構成を解除する方法を示しますが、Port デバイスや HCA_SVC デバイスにも同じ手順を適用できます。

1 スーパーユーザーになります。

2 **IB VPPA デバイスの構成を解除します。**

次に例を示します。

```
# cfgadm -c unconfigure ib::1730000007F51,8001,ipib
Unconfigure the device: /devices/ib:fabric::1730000007F51,8001,ipib
This operation will suspend activity on the IB device
Continue (yes/no)? Y
#
```

3 デバイスが接続解除されていることを確認します。

次に例を示します。

```
# cfgadm -a ib::1730000007F51,8001,ipib
Ap_Id                          Type    Receptacle Occupant     Condition
ib::1730000007F51,8001,ipib IB-VPPA connected  unconfigured unknown
#
```

▼ IB Port、HCA_SVC、VPPA デバイスを構成する方法

IB Port、HCA_SVC、または VPPA デバイスをシステム上で構成するには、次の手順を使用します。

次の例では VPPA デバイスを構成する方法を示しますが、Port デバイスや HCA_SVC デバイスにも同様の手順を適用できます。

1 スーパーユーザーになります。

2 **IB VPPA デバイスを構成します。**

次に例を示します。

```
# cfgadm -c configure ib::1730000007F51,8001,ipib
```

3 デバイスが接続されていることを確認します。

次に例を示します。

```
# cfgadm -a ib::1730000007F51,8001,ipib
Ap_Id                          Type     Receptacle Occupant   Condition
ib::1730000007F51,8001,ipib IB-VPPA  connected  configured ok
```

注 – IB Port デバイスや HCA_SVC デバイスの場合も、cfgadm による構成操作および構成解除操作は、前述の IB VPPA デバイスの例と同様です。

▼ IB 擬似デバイスの構成を解除する方法

IB 擬似デバイスをシステムから削除するには、次の手順を使用します。

1 スーパーユーザーになります。

2 IB 擬似デバイスの構成を解除します。

次に例を示します。

```
# cfgadm -c unconfigure ib::ibgen,0
Unconfigure the device: /devices/ib:fabric::ibgen,0
This operation will suspend activity on the IB device
Continue (yes/no)? Y
#
```

3 デバイスが接続解除されていることを確認します。

```
# cfgadm -a ib::ibgen,0
Ap_Id                  Type      Receptacle Occupant     Condition
ib::ibgen,0            IB-PSEUDO connected  unconfigured unknown
```

▼ IB 擬似デバイスを構成する方法

IB 擬似デバイスを構成するには、次の手順を使用します。

1 スーパーユーザーになります。

2 IB 擬似デバイスを構成します。

次に例を示します。

```
# cfgadm -yc configure ib::ibgen,0
```

3 デバイスが接続されていることを確認します。

次に例を示します。

```
# cfgadm -a ib::ibgen,0
Ap_Id                  Type       Receptacle Occupant   Condition
ib::ibgen,0            IB-PSEUDO  connected  configured ok
```

▼ HCA のカーネル IB クライアントを表示する方法

次の IB `cfgadm` プラグインコマンドを呼び出すと、この HCA を使用しているカーネル IB クライアントを一覧表示できます。カーネル IB クライアントが別の HCA を使用する場合は、最後の列に「yes」と表示されます。この HCA を使用しない IB マネージャーとカーネルクライアントは、`Ap_Id` の列に「-」と表示されます。

- **HCA** のカーネル **IB** クライアントを表示します。
  次に例を示します。

```
$ cfgadm -x list_clients hca:173000007F50
Ap_Id                        IB Client         Alternate HCA
ib::1730000007F51D0             ibgen              no
ib::1730000007F51D1             ibgen              no
ib::1730000007F51,8001,ipib     ibd                no
ib::ibgen,0                     ibgen              no
-                               ibdm               no
-                               ibmf               no
-                               nfs/ib             no
$
```

▼ HCA に接続されている IB デバイスの構成を解除する方法

IB cfgadm プラグインには、HCA の実際の DR を行う機能は含まれていません。HCA の DR を実際に行うには、基になるバスのプラグインを使用します。たとえば、PCI ベースの HCA の場合は、cfgadm_pci コマンドを使用できます。詳細は、cfgadm_pci(1M) のマニュアルページを参照してください。

ただし、IB cfgadm プラグインは、HCA のカーネル IB クライアントを一覧表示することで、HCA の DR を補助します。その手順を次に示します。

1. スーパーユーザーになります。
2. **HCA** のカーネル **IB** クライアントを一覧表示します。
   次に例を示します。

```
# cfgadm -x list_clients hca:173000007F50
Ap_Id                        IB Client         Alternate HCA
ib::1730000007F51D0             ibgen              no
ib::1730000007F51D1             ibgen              no
ib::1730000007F51,8001,ipib     ibd                no
ib::ibgen,0                     ibgen              no
-                               ibdm               no
-                               ibmf               no
-                               nfs/ib             no
```

3. 代替 **HCA** を持っていないカーネル **IB** クライアント (**Port**、**VPPA**、**HCA_SVC**、**IOC** デバイスなど) の構成を解除します。
   次に例を示します。

```
# cfgadm -x unconfig_clients hca:1730000008070
Unconfigure Clients of HCA /devices/ib:1730000008070
This operation will unconfigure IB clients of this HCA
Continue (yes/no)? y
```

4 **HCA のカーネル IB クライアントの構成が解除されていることを確認します。**

```
# cfgadm -x list_clients hca:173000007F50
Ap_Id                        IB Client         Alternate HCA
-                               ibdm               no
-                               ibmf               no
-                               nfs/ib             no
#
```

IB HCA を構成する方法

バス固有の cfgadm プラグインを呼び出して HCA を構成します。しかし、その詳細についてはこの章では記載していません。

▼ IB p_key テーブルを更新する方法

追加の p_key が有効にされた場合や無効にされた場合など、HCA ポートの p_key テーブルの情報が変更された場合は、InfiniBand Transport Framework (IBTF) と IBDM に通知して内部 p_key データベースを更新する必要があります。cfgadm コマンドは、IBTF および IBDM の p_key データベースの更新を補助します。詳細は、ibtl(7D) および ibdm(7D) のマニュアルページを参照してください。

1 スーパーユーザーになります。

2 p_key テーブルを更新します。

次に例を示します。

```
# cfgadm -x update_pkey_tbls -y ib
```

▼ IB 通信サービスを表示する方法

IBTF で現在使用されている IB 通信サービスを表示するには、次の手順を使用します。

1 スーパーユーザーになります。

2 IB 通信サービスを表示します。

次に例を示します。

```
# cfgadm -x list_services ib
Port communication services:
                srp
VPPA communication services:
                ibd
```

```
HCA_SVC communication services:
                hnfs
```

▼ VPPA 通信サービスを追加する方法

新しい VPPA 通信サービスを追加するには、次の手順を使用します。

新しい HCA_SVC 通信サービスやポート通信サービスも、同様の手順で追加できます。

1 スーパーユーザーになります。

2 新しい **VPPA** 通信サービスを追加します。
次に例を示します。

```
# cfgadm -o comm=vppa,service=new -x add_service ib
```

3 新しいサービスが追加されていることを確認します。
次に例を示します。

```
# cfgadm -x list_services ib
Port communication services:
                srp
VPPA communication services:
                ibd
                new
HCA_SVC communication services:
                nfs_service
#
```

▼ 既存の IB Port、HCA_SVC、VPPA 通信サービスを削除する方法

既存の IB Port、HCA_SVC、VPPA 通信サービスを削除するには、次の手順を使用します。

1 スーパーユーザーになります。

2 **VPPA** 通信サービスを削除します。
次に例を示します。

```
# cfgadm -o comm=vppa,service=new -x delete_service ib
```

3 通信サービスが削除されていることを確認します。

次に例を示します。

```
# cfgadm -x list_services ib
Port communication services:
                srp
VPPA communication services:
                ibd
HCA_SVC communication services:
                hnfs
#
```

▼ IOC の構成を更新する方法

すべての IOC デバイスノードまたは特定の IOC Ap_Id についてプロパティーを更新するには、次の手順を使用します。更新できるプロパティーは次のとおりです。

- port-list
- port-entries
- service-id
- service-name

これらのプロパティーの詳細は、ib(7D) のマニュアルページを参照してください。

構成がまったく変更されていない場合、これらのプロパティーは更新されません。次の例は、特定の IOC の構成を更新する方法を示しています。すべての IOC の構成を更新するには、特定の IOC Ap_Id の代わりに、静的な ib Ap_Id を指定します。

1 スーパーユーザーになります。

2 IOC の構成を更新します。

次に例を示します。

```
# cfgadm -x update_ioc_conf ib::1730000007F5198
This operation can update properties of IOC devices.
Continue (yes/no)? y
#
```

3 prtconf -v を実行して、プロパティーが更新されていることを確認します。

InfiniBand デバイスでの uDAPL アプリケーションインタフェースの使用

uDAPL (User Direct Access Programming Library) は、RDMA (Remote Direct Memory Access) に対応した InfiniBand などの相互接続を介して行われるデータセンターアプ

リケーションのデータメッセージングのパフォーマンス、スケーラビリティー、および信頼性を向上させる標準 API です。uDAPL インタフェースは DAT Collaborative によって定義されています。DAT Collaborative の詳細は、次の Web サイトを参照してください。

http://www.datcollaborative.org

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

- 標準の DAT レジストリライブラリ `libdat`。詳細は、`libdat(3LIB)` のマニュアルページを参照してください。
- 標準のサービスプロバイダ登録ファイル `dat.conf`。詳細は、`dat.conf(4)` のマニュアルページを参照してください。
- 複数のサービスプロバイダのサポート。これにより、各プロバイダが uDAPL ライブラリパスやバージョン番号などを独自の `service_provider.conf` ファイルで指定できます。詳細は、`service_provider.conf(4)` のマニュアルページを参照してください。
- `datadm` コマンド。これは `dat.conf` を構成するための管理ツールです。詳細は、`datadm(1M)` のマニュアルページを参照してください。
- 新しいリソース制御プロパティー `project.max-device-locked-memory`。ロックされる物理メモリーの量を調節します。
- アドレス解決に IP インフラストラクチャーを活用する IPv4 アドレスまたは IPv6 アドレスを使った命名方式。IPv4 での ARP や IPv6 での近隣探索などが含まれます。Solaris uDAPL インタフェースアダプタは、IPoIB デバイスインスタンスに直接対応づけられます。
- DAT Collaborative コミュニティーによって使用されている標準のアドレス変換方式のサポート。
- Mellanox Tavor ホストチャネルアダプタをサポートするための uDAPL サービスプロバイダライブラリ。`dat.conf` 登録ファイルへの自動登録を提供します。
- SPARC プラットフォームと x86 プラットフォームの両方をサポートします。

▼ uDAPL を使用可能にする方法

1. スーパーユーザーになります。
2. 次のパッケージがインストールされていることを確認します。必要に応じて、これらをインストールします。
   - `SUNWib` - Sun InfiniBand フレームワーク
   - `SUNWtavor` - Sun Tavor HCA ドライバ
   - `SUNWipoib` - Sun IP over InfiniBand

- SUNWudaplr - DAT (Direct Access Transport) レジストリパッケージ (root)
- SUNWudaplu - DAT (Direct Access Transport) レジストリパッケージ (usr)
- SUNWudapltr - Tavor パッケージ用サービスプロバイダ (root)
- SUNWudapltu - Tavor パッケージ用サービスプロバイダ (usr)

3 **IPoIB インタフェースを認識させるには、次のいずれかの手順を選択します。**

- ifconfig コマンドと datadm コマンドを使用して、インタフェースを手動で使えるようにします。

  次に例を示します。

```
# ifconfig ibd1 plumb
# ifconfig ibd1 192.168.0.1/24 up
# datadm -a /usr/share/dat/SUNWudaplt.conf
```

- 次の方法で、インタフェースを自動的に使えるようにします。
  - 次のファイルを作成し、適切な IP アドレスを指定します。

    /etc/hostname.ibd1
  - システムを再起動します。

DAT 静的レジストリの更新

datadm コマンドを使用して、DAT 静的レジストリ dat.conf ファイルを管理できます。このファイルの詳細は、dat.conf(4) のマニュアルページを参照してください。

datadm コマンドを使用して、サービスプロバイダを dat.conf ファイルに登録したり登録を解除したりすることもできます。詳細は、datadm(1M) のマニュアルページを参照してください。

IPoIB インタフェースアダプタの追加や削除を行なったときは、システムの現在の状態を反映させるために、datadm コマンドを実行して dat.conf ファイルを更新してください。現在インストールされているすべてのサービスプロバイダについて、インタフェースアダプタの新しいセットが再生成されます。

▼ DAT 静的レジストリを更新する方法

1 スーパーユーザーになります。

2 システムに **IPoIP** インタフェースアダプタを追加したり、システムから **IPoIP** インタフェースアダプタを削したりしたあとに、**DAT** 静的レジストリを更新します。

```
# datadm -u
```

3 更新された **DAT** 静的レジストリを表示します。

```
# datadm
```

▼ DAT 静的レジストリにサービスプロバイダを登録する方法

1 スーパーユーザーになります。

2 **Mellanox Tavor** ホストチャネルアダプタ用の **Sun** のサービスプロバイダを追加したあとに、**DAT** 静的レジストリを更新します。

```
# datadm -a /usr/share/dat/SUNWudaplt.conf
```

3 更新された **DAT** 静的レジストリを表示します。

```
# datadm -v
```

▼ DAT 静的レジストリからサービスプロバイダの登録を解除する方法

1 スーパーユーザーになります。

2 **Mellanox Tavor** ホストチャネルアダプタ用の **Sun** のサービスプロバイダをシステムから削除したあとに、**DAT** 静的レジストリを更新します。

```
# datadm -r /usr/share/dat/SUNWudaplt.conf
```

3 更新された **DAT** 静的レジストリを表示します。

```
# datadm -v
```

◆ ◆ ◆ 第 10 章

10 ディスクの管理 (概要)

この章では、Oracle Solaris ディスクスライスの概要および format ユーティリティーについて説明します。

この章の内容は次のとおりです。

- 191 ページの「ディスク管理の新機能」
- 196 ページの「ディスク管理作業についての参照先」
- 196 ページの「ディスク管理の概要」
- 197 ページの「ディスク関連の用語」
- 197 ページの「ディスクラベルについて」
- 202 ページの「ディスクスライスについて」
- 207 ページの「format ユーティリティー」
- 210 ページの「ディスクをパーティションに分割する」

システムにディスクを追加する方法については、第 12 章「SPARC: ディスクの追加 (手順)」または第 13 章「x86: ディスクの追加 (手順)」を参照してください。

ディスク管理の新機能

この節では、Solaris リリースの新しいディスク管理機能について説明します。

- 192 ページの「Solaris OS のインストール時およびブート時における 2T バイトディスクのサポート」
- 193 ページの「Solaris iSCSI ターゲットおよびイニシエータでの iSNS のサポート」
- 193 ページの「Solaris iSCSI ターゲットのサポート」
- 194 ページの「Solaris iSCSI イニシエータのサポート」
- 195 ページの「x86: GRUB ブート環境でのディスク管理」
- 196 ページの「2T バイトを超える SCSI ディスクのサポート」

Solaris の新機能の一覧および Solaris リリースについての説明は、『Oracle Solaris 10 9/10 の新機能』を参照してください。

Solaris OS のインストール時およびブート時における 2T バイトディスクのサポート

Solaris 10 10/09: 以前の Solaris リリースでは、1T バイトを超えるディスクに Solaris OS をインストールしてブートすることはできませんでした。この Solaris リリースでは、最大 2T バイトのサイズのディスクに Solaris OS をインストールしてブートできます。以前のリリースでは、1T バイトを超えるディスクには EFI ラベルを使用する必要がありました。このリリースでは、任意のサイズのディスクに VTOC ラベルを使用できますが、VTOC でアドレス指定可能な容量は 2T バイトに制限されています。

Solaris ディスクドライバおよびディスクユーティリティーが更新され、次のサポートが提供されるようになりました。

- 2T バイトのディスクに Solaris OS をインストールおよびブートする場合、そのディスクは 1G バイト以上のメモリーを搭載し、64 ビットカーネルの動作するシステムに接続されている必要があります。
- `format -e` ユーティリティーにより、VTOC ラベルを使って任意のサイズのディスクにラベルを付けることができます。ただし、アドレス指定可能な容量は 2T バイトに制限されています。
- `format` ユーティリティーおよびインストールソフトウェアにより、2T バイト未満のサイズのディスクに使用されるデフォルトのラベルは VTOC ラベルです。
- x86 システムの 1T バイトを超えるディスクに対しては、`fdisk` ユーティリティーを使用できます。非 EFI パーティションタイプの MBR で、2T バイト以内のパーティションへのサポートが追加されました。このサポートにより、2T バイトまでの Solaris パーティションを使用できるようになりました。その他の非 EFI パーティションでは、パーティションタイプによって制限が課される場合があります。

  2T バイトを超えるディスクに対して `fdisk` ユーティリティーを実行すると、2T バイトを超える非 EFI パーティションを作成することはできないことを示す警告メッセージが表示されます。
- Solaris ボリュームマネージャーソフトウェアが変更され、作成するメタデバイスで VTOC ラベルの付いた 2T バイト以下の物理ディスクがサポートされるようになりました。

1T バイトを超えるレガシー MBR またはレガシー VTOC ディスクを、以前の Solaris リリースに移動することはできないことに留意してください。EFI ラベルの付いたディスクは、以前の Solaris リリースの場合と同様に引き続き動作します。

このリリースでの EFI ラベルの変更の詳細は、198 ページの「EFI ディスクラベル」を参照してください。

Solaris iSCSI ターゲットおよびイニシエータでの iSNS のサポート

Solaris 10 8/07: この Solaris リリースでは、Solaris iSCSI ターゲットおよびイニシエータソフトウェアで iSNS (Internet Storage Name Service) プロトコルがサポートされます。iSNS プロトコルによって、TCP/IP ネットワーク上での iSCSI デバイスの発見、管理、および設定を自動化できます。

この Solaris リリースでは、`iscsitadm` コマンドを使って既存の他社製 iSNS サーバーへのアクセスを追加したり、Solaris iSNS サーバーを使ってネットワーク内の iSCSI デバイスを自動的に発見したりすることができます。iSNS サーバーはホスト名または IP アドレスで指定できます。iSNS サーバーの情報を追加したあと、サーバーへのアクセスを有効にする必要があります。

詳しい手順については次の情報を参照してください。

- 他社製 iSNS サーバーを使用するように Solaris iSCSI ターゲットを設定する方法については、第 14 章「Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順)」および `iscsitadm(1M)` のマニュアルページを参照してください。
- Solaris Express リリースの Solaris iSNS サーバーを使って Solaris iSCSI ターゲットを設定する方法については、『System Administration Guide: Devices and File Systems』の第 15 章「Configuring and Managing the Solaris Internet Storage Name Service (iSNS)」を参照してください。

Solaris iSCSI ターゲットのサポート

Solaris 10 8/07: この Solaris リリースでは、iSCSI ターゲットデバイス (ディスクまたはテープデバイス) がサポートされます。以前の Solaris リリースでは、iSCSI イニシエータがサポートされます。Solaris iSCSI ターゲットを設定する利点は、ファイバチャネル HBA の費用をかけずに、クライアントに接続できる既存のファイバチャネルデバイスを持てることです。また、専用のアレイを持つシステムが、複製されたストレージを ZFS または UFS ファイルシステムでエクスポートすることも可能になりました。

`iscsitadm` コマンドを使用して、iSCSI ターゲットデバイスを設定および管理できます。iSCSI ターゲットとして選択したディスクデバイスでは、iSCSI デーモン用のバッキングストアとして、同サイズの ZFS または UFS ファイルシステムを提供する必要があります。

ターゲットデバイスの設定後に、`iscsiadm` コマンドを使って iSCSI ターゲットを識別します。これにより、iSCSI ターゲットデバイスが検出および使用されます。

詳細は、第 14 章「Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順)」、`iscsiadm(1M)` のマニュアルページ、および `iscsitadm(1M)` のマニュアルページを参照してください。

注 – このマニュアルの以前のバージョンでは、Solaris 10 11/06 リリースで Solaris iSCSI ターゲットがサポートされると記述されていましたが、それは誤りです。この機能は、Solaris 10 8/07 リリースから使用できるようになりました。

Solaris iSCSI イニシエータのサポート

Solaris 10 1/06: iSCSI は、データストレージサブシステムを接続するための、インターネットプロトコル (Internet Protocol、IP) ベースのストレージネットワーク標準です。SCSI コマンドを IP ネットワーク経由で伝送するため、iSCSI プロトコルにより、ネットワーク上でローカルシステムにディスクデバイスをマウントできます。ローカルシステム上では、デバイスをブロックデバイスのように使用できます。

詳細は、第 14 章「Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順)」を参照してください。

Solaris iSCSI イニシエータサポートの機能拡張

Solaris 10 6/06: Solaris iSCSI イニシエータサポートに次の拡張機能が追加されました。

- 動的なターゲット削除のサポート – システムをリブートしないで iSCSI ターゲットを削除 (またはログアウト) することができます。発見方式または発見アドレスを削除または無効にしようとしたときにターゲットが使用されていない場合は、ターゲットが削除され、関連するリソースが開放されます。ターゲットが使用されている場合は、発見アドレスまたは発見方式が有効のままになり、使用中を示すメッセージが表示されます。

  詳細は、280 ページの「発見された iSCSI ターゲットを削除する方法」を参照してください。

- Internet Storage Name Service (iSNS) クライアントのサポート – これを使用すると、できる限り少ない構成で、iSCSI イニシエータがアクセス権を持つターゲットを発見できます。また、ストレージノードの動作状態が変更されたときに iSCSI イニシエータに通知する状態変更通知機能もあります。`iscsiadm` コマンドが iSNS の発見をサポートするように機能拡張されました。

  詳細は、279 ページの「iSCSI ターゲット発見を構成する方法」を参照してください。

- マルチセッションターゲット (MS/T) のサポート – 必要に応じてより多くの iSCSI セッションまたはターゲットへのパスを作成できます。追加の iSCSI パスを使用すると、ログインのリダイレクションをサポートする iSCSI アレイなどの特定の構成で帯域幅が集約され可用性が向上します。iSCSI MS/T 機能は、MPxIO またはほかのマルチパス化ソフトウェアと組み合わせて使用する必要があります。`iscsiadm` コマンドが MS/T をサポートするように機能拡張されました。

Solaris iSCSI イニシエータの構成の詳細は、第 14 章「Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順)」および iscsiadm(1M) のマニュアルページを参照してください。

x86: GRUB ブート環境でのディスク管理

Solaris 10 1/06: x86 システムの従来のブート方式が GRUB ブートメニューに置き換わっています。ディスク管理の領域では、システムディスクを置き換えるための代替デバイスからブートするとき、またはブートブロックをインストールするときは、GRUB インタフェースを使用します。

GRUB ブート環境は次の機能を提供します。

- **Solaris** フェイルセーフブート – 代わりのデバイスからブートしなくてもシステムをブートできない問題を解決できるように、ミニルートにブートする Solaris フェイルセーフブートのオプション。矢印キーを使用して GRUB ブートメニューから次のオプションを選択したあと、Return キーを押します。

  ```
  Solaris failsafe
  ```

  Solaris フェイルセーフオプションを使用した場合は、あとでシステムをリブートする必要があります。
- ネットワークブート – BIOS の構成中に F12 キーを押して、ネットワークからブートします。
- シングルユーザーブート – Solaris フェイルセーフブートメニューから次のオプションを選択して、シングルユーザーモードでブートします。

  ```
  kernel /platform/i86pc/multiboot
  ```

  次に、e (編集) オプションを使用して -s シングルユーザーオプションを追加します。次に例を示します。

  ```
  kernel /platform/i86pc/multiboot -s
  ```

  Return キーを押してから b キーを押してシステムをブートします。Control + D キーを押して、システムをマルチユーザーモードでブートしなおします。

GRUB 環境では、x86 システム上で動作しているときに、fmthard コマンドを使用してブートブロックを自動的にインストールすることはできません。ブートブロックは別個にインストールする必要があります。

x86 システム上で新しい GRUB ベースのブートを使用するときの機能説明および手順の詳細については、『Solaris のシステム管理 (基本編)』の「GRUB を使用して x86 システムをブートする (作業マップ)」を参照してください。

GRUB ブート環境でディスクを管理する手順については、次の項目を参照してください。

- 265 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」
- 253 ページの「x86: システムディスクの接続方法」

この機能は、SPARC システムでは使用できません。

2T バイトを超える SCSI ディスクのサポート

Solaris 10 1/06: SCSI ドライバ `ssd` または `sd` は、2T バイト以上をサポートします。以前の Solaris リリースでは、SCSI ドライバ `ssd` または `sd` が 2T バイトに制限されています。

`format` ユーティリティーを使用して、これらの大容量ディスクのラベル作成、構成、およびパーティション作成を行うことができます。大容量ディスクで EFI ディスクラベルを使用する方法および `fdisk` ユーティリティーの制限事項については、 199 ページの「EFI ディスクラベルの制限」を参照してください。

ディスク管理作業についての参照先

ディスク管理の手順については、次を参照してください。

ディスク管理作業	参照先
ディスクをフォーマットしてディスクラベルを確認します。	第 11 章「ディスクの管理 (手順)」
SPARC システムに新しいディスクを追加します。	第 12 章「SPARC: ディスクの追加 (手順)」
x86 システムに新しいディスクを追加します。	第 13 章「x86: ディスクの追加 (手順)」
SCSI または PCI ディスクのホットプラグを実行します。	第 6 章「デバイスの動的構成 (手順)」

ディスク管理の概要

一般に、Oracle Solaris OS におけるディスクの管理とは、システムを設定し、Solaris インストールプログラムを実行し、適切なディスクスライスおよびファイルシステムを作成して Oracle Solaris OS をインストールすることを意味します。また、`format` ユーティリティーを使用して、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ーを理解しておく必要があります。特に、次の用語を理解しておいてください。

用語	説明
トラック	ディスクが回転するときに 1 つの静止したディスクヘッドの下を通過する同心リング。
シリンダ	ディスクが回転する軸から同じ距離にあるトラックの集まり。
セクター	各ディスクプラッタのセクション。1 セクターは 512 バイトです。
ブロック	ディスク上のデータ記憶領域。1 ディスクブロックは 512 バイトです。
ディスクコントローラ	ディスクドライブを制御するチップおよび関連する回路。
ディスクラベル	ディスクのジオメトリおよびパーティション情報を含む、ディスクの第 1 セクター。
デバイスドライバ	ハードウェアまたは仮想デバイスを制御するカーネルモジュール。

詳細は、ディスク製造元の製品情報を参照してください。

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、および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。このような情報をディスクの「ラベル」と呼びます。VTOC ラベル付きのディスク上のディスクラベルを「VTOC (Volume Table of Contents)」と呼びます。「ディスクにラベルを付ける」とは、ディスクにスライス情報を書き込むことを意味します。通常は、ディスクのスライスを変更した後にラベルを付けます。

Solaris リリースでは、次の 2 つのディスクラベルをサポートしています。

- SMI – 従来の VTOC ラベル。サイズが 2T バイトに満たないディスク用です。
- EFI – 64 ビット Solaris カーネルを実行しているシステムで 2T バイトを超えるディスクをサポートします。Extensible Firmware Interface GUID Partition Table (EFI GPT) ディスクラベルは、32 ビット Solaris カーネルを実行しているシステムに接続された 2T バイト未満のディスクにも使用できます。

スライスを作成したあとでディスクにラベルを付けないと、OS 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。

EFI ディスクラベル

EFI ラベルは、サイズが 2T バイトを超える物理ディスクボリュームと仮想ディスクボリュームをサポートします。このリリースには、2T バイトを超えるサイズのディスクを管理するための更新版ディスクユーティリティーが付属しています。UFS ファイルシステムには EFI ディスクラベルとの互換性があり、2T バイトを超えるサイズの UFS ファイルシステムを作成できます。マルチテラバイトの UFS ファイルシステムの作成方法については、341 ページの「64 ビット: マルチテラバイトの UFS ファイルシステムのサポート」を参照してください。

1T バイトを超えるファイルシステムを作成する必要がある場合は、別パッケージの Sun QFS ファイルシステムも使用できます。Sun QFS ファイルシステムについては、『Sun QFS, Sun SAM-FS, and Sun SAM-QFS File System Administrator's Guide』を参照してください。

今回の Solaris リリースでは、Solaris ボリュームマネージャーでも 1T バイトを超えるディスク管理が可能です。Solaris ボリュームマネージャーの使用方法については、『Solaris ボリュームマネージャの管理』を参照してください。

サイズが 2T バイトに満たないディスクでは、引き続き VTOC ラベルを使用できます。現在のシステムでサイズが 2T バイトに満たないディスクしか使用しない場合は、以前の Solaris リリースと同じ方法でディスク管理を行います。サイズが 2T バイト以下のディスクに EFI ラベルでラベル付けを行う場合は、`format -e` コマンドを使用します。詳細は、例 11–6 を参照してください。

システム上で適切な Solaris リリースが実行されている場合、`format -e` コマンドを使用することで EFI ラベルをディスクに適用できます。ただし、EFI ラベルを適用する前に、199 ページの「EFI ディスクラベルの制限」に記載された重要な情報を確認することをお勧めします。

EFI ラベルが不要になった場合に、`format -e` コマンドを使って、VTOC ラベルを再度適用することもできます。次に例を示します。

```
# format
Specify disk (enter its number): 2
selecting c0t5d0
[disk formatted]
.
.
.
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Warning: This disk has an EFI label. Changing to SMI label will erase all
current partitions.
Continue? yes
Auto configuration via format.dat[no]?
Auto configuration via generic SCSI-2[no]?
format> quit
```

注意 – ディスクラベルを変更すると、ディスク上のデータがすべて破棄されることに留意してください。

EFI ラベルと VTOC ラベル

EFI ディスクラベルが VTOC ディスクラベルと異なる点は次のとおりです。

- サイズが 2T バイトを超えるディスクをサポートします。
- スライス 0 - 6 を使用できます (スライス 2 はその他のスライス)。
- 一次 (バックアップ) ラベルまたはその他のパーティションを使ってパーティションやスライスをオーバーラップすることはできません。EFI ラベルのサイズは通常 34 セクターなので、パーティションは通常セクター 34 で始まります。つまり、パーティションはセクターゼロ (0) では始められないことを意味します。
- シリンダ、ヘッド、およびセクターの情報は、EFI ラベルには格納されません。サイズはブロック単位で報告されます。
- 代替シリンダ領域 (ディスクの終わりから 2 つめまでのシリンダ) に格納されていた情報は、スライス 8 に格納されます。
- `format` ユーティリティーを使ってパーティションサイズを変更する場合、サイズ 0 のパーティションには `unassigned` パーティションタグが割り当てられます。`format` ユーティリティーでは、0 より大きいサイズのパーティションには、デフォルトにより `usr` パーティションタグが割り当てられます。パーティションを変更したあと新しいパーティションタグを割り当てたい場合は、パーティション変更メニューを使用します。ただし、サイズ 0 以外のパーティションに `unassigned` パーティションタグを割り当て直すことはできません。

EFI ディスクラベルの制限

サイズが 1T バイトを超えるディスクを使用することが現在の環境にとって適切かどうかを判断するときは、次の点を考慮してください。

- VTOC ラベル付きディスクを使用するシステム向けの階層化されたソフトウェア製品で、EFI ラベル付きディスクにアクセスできないことがあります。
- 以前の Solaris リリースを実行しているシステムは、EFI ラベル付きディスクを認識しません。
- EFI ラベル付きディスクからブートを実行できません。
- x86 ベースのシステムでは、EFI ラベル付きの 2T バイトを超えるディスク上で `fdisk` コマンドを使用できます。

- Solaris 管理コンソールの「ディスク」ツールでは、EFI ラベル付きディスクを管理できません。format ユーティリティーを使って EFI ラベル付きディスクをパーティションに分割します。すると、Solaris 管理コンソールの拡張ストレージツールを使って、EFI ラベル付きディスクを含むボリュームやディスクセットを管理できるようになります。
- EFI 仕様では、スライスのオーバーラップは禁止されています。*cxtydz* でディスク全体を表現します。
- EFI ディスクラベルは、ディスクやパーティションのサイズ情報を提供します。使用可能な単位はセクターまたはブロックです。シリンダおよびヘッドは使用できません。
- 次の format オプションは、EFI ラベル付きディスクではサポートされていないか、不適切です。
  - save オプション。EFI ラベル付きディスクは format.dat ファイルにエントリを必要としません。したがって、このオプションはサポートされません。
  - backup オプション。ディスクドライバは、一次ラベルを検出してディスクに書き込みます。したがって、このオプションは不適切です。

x86 システムにおける EFI ラベル付きディスクのサポート

x86 システム上では、EFI ディスクラベルの Solaris サポートが利用可能です。x86 システムで EFI ラベルを追加するには、次のコマンドを使用します。

```
# format -e
> [0] SMI Label
> [1] EFI Label
> Specify Label type[0]: 1
> WARNING: converting this device to EFI labels will erase all current
> fdisk partition information. Continue? yes
```

以前のラベル情報は EFI ディスクラベルに変換されません。

format コマンドを使ってラベルのパーティション情報を手動で作成し直す必要があります。EFI ラベル付きの 2T バイトのディスク上で fdisk コマンドを使用することはできません。2T バイトを超えるディスクで fdisk コマンドを実行して Solaris パーティションを作成する場合、Solaris パーティションは 2T バイトに制限されます。EFI ディスクラベルの詳細は、前の節を参照してください。

EFI ラベル付きディスクを使用したシステムへのインストール

Solaris インストールユーティリティーは、EFI ラベル付きディスクを自動的に認識します。しかし、Solaris インストールプログラムでは、これらのディスクのパーティション分割をやり直すことはできません。インストール前またはインストール後に EFI ラベル付きディスクのパーティション分割をやり直す場合は、format ユーティリティーを使用してください。Solaris Upgrade ユーティリティーおよび Live Upgrade ユーティリティーも EFI ラベル付きディスクを認識します。ただし、EFI ラベル付きディスクからシステムをブートすることはできません。

EFI ラベル付きディスクを使用するシステム上に Solaris をインストールした場合、次のようなパーティションテーブル情報が得られます。

```
Current partition table (original):
Total disk sectors available: 2576924638 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector          Size          Last Sector
  0       root    wm                34         1.20TB           2576924636
  1 unassigned    wm                 0            0                    0
  2 unassigned    wm                 0            0                    0
  3 unassigned    wm                 0            0                    0
  4 unassigned    wm                 0            0                    0
  5 unassigned    wm                 0            0                    0
  6 unassigned    wm                 0            0                    0
  8   reserved    wm        2576924638         8.00MB           2576941021
```

EFI ラベル付きディスクの管理

EFI ラベル付きディスクの管理方法は、次の表で確認できます。

作業	参照先
インストール済みのシステムにディスクを接続し、再構成 (ブート) を実行します。	239 ページの「SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)」または 251 ページの「x86: システムディスクまたは二次ディスクの追加 (作業マップ)」
`format` ユーティリティーを使ってディスクのパーティション分割を行います (必要に応じて)。	243 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 263 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」
ディスクボリュームを作成します。その後、Solaris ボリュームマネージャーでソフトパーティションを作成します (必要に応じて)。または、ZFS ストレージプールを設定します。	『Solaris ボリュームマネージャの管理』の第 2 章「記憶装置管理の概念」または『Oracle Solaris ZFS 管理ガイド』の「ZFS ストレージプールを作成する」
`newfs` コマンドを使用して新しいディスク用の UFS ファイルシステムを作成します。	248 ページの「SPARC: UFS ファイルシステムを作成する方法」または 264 ページの「x86: ファイルシステムを作成する方法」
または ZFS ファイルシステムを作成します。	『Oracle Solaris ZFS 管理ガイド』の「ZFS ファイルシステムを作成する方法」
EFI ラベル付きディスクを複製します。	例 28–2

EFI ディスクラベルに関する問題の障害追跡

EFI ラベル付きディスクに関する問題のトラブルシューティングを行うときは、次のエラーメッセージと解決方法を使用してください。

エラーメッセージ

```
Dec  3 09:26:48 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):
Dec  3 09:26:48 holoship disk has 2576941056 blocks, which is too large
for a 32-bit kernel
```

エラーの発生原因
32 ビットの SPARC または x86 カーネルを実行しているシステムを、サイズが 1T バイトを超えるディスクでブートしようとしました。

解決方法
サイズが 1T バイト以上のディスクでは、64 ビットの SPARC または x86 カーネルを実行しているシステムをブートしてください。

エラーメッセージ

```
Dec  3 09:12:17 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):
Dec  3 09:12:17 holoship corrupt label - wrong magic number
```

エラーの発生原因
古い Solaris リリースで動作するシステムにディスクを追加しようとしました。

解決方法
このディスクは、EFI ディスクラベルをサポートしている Solaris リリースで動作するシステムに追加してください。

ディスクスライスについて

ディスク上に格納されたファイルは、ファイルシステム中で管理されます。ディスク上の各ファイルシステムは「スライス」、つまりファイルシステム用に確保されたセクターセットのグループに割り当てられます。Oracle Solaris OS (および、システム管理者) からは、各ディスクスライスは別個のディスクドライブであるかのように見えます。

ファイルシステムの詳細は、第 16 章「ファイルシステムの管理 (概要)」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。`format` ユーティリティーなど、特定のインタフェースではスライスを「パーティション」と呼びます。

スライスを設定するときには、次の規則に注意してください。

- 各ディスクスライスは、ファイルシステムを 1 つしか持てない。
- ファイルシステムを複数のスライスにまたがって割り当てることはできない。

SPARC プラットフォームと x86 プラットフォームでは、スライスの設定が少し異なります。次の表に、主な相違点を示します。

表 10–1　SPARC プラットフォームと x86 プラットフォームでのスライスの違い

SPARC プラットフォーム	x86 プラットフォーム
ディスク全体が Oracle Solaris OS 用になります。	ディスクはオペレーティングシステムごとに 1 つの `fdisk` パーティションに分割されます。
VTOC – ディスクは 8 スライス (スライス 0 - 7) に分割されます。	**VTOC** – Solaris の `fdisk` パーティションは 10 スライス (スライス 0 - 9) に分割されます。
EFI – ディスクは 7 スライス (スライス 0 - 6) に分割されます。	**EFI** – ディスクは 7 スライス (スライス 0 - 6) に分割されます。

Solaris ボリュームマネージャー (以前の Solstice DiskSuite) には「ソフトパーティション」と呼ばれるパーティション分割機能が備わっています。ソフトパーティションを使えば、1 つのディスクに 9 個以上のパーティションを作成できます。

Solaris ボリュームマネージャーの一般的な情報については、『Solaris ボリュームマネージャの管理』の第 2 章「記憶装置管理の概念」を参照してください。ソフトパーティションについては、『Solaris ボリュームマネージャの管理』の第 12 章「ソフトパーティション (概要)」を参照してください。

ディスクスライス

Oracle Solaris OS を実行するシステムには、次のようなスライスが存在する可能性があります。

x86 システムの場合

- 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、Oracle Solaris OS など、特定のオペレーティングシステムで使用するように予約されたディスクの一部です。
- Oracle Solaris OS では、Solaris の `fdisk` パーティションは 10 スライス (スライス 0 - 9) に分割されます。

表 10–2　ディスクスライス

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	Comments
0	ルート (/)	両方	OS を構成するファイルとディレクトリを含みます。 **EFI** – EFI ラベル付きディスクからはブートを実行できません。

表 10–2 ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	Comments
1	スワップ領域	両方	仮想メモリー、つまり「スワップ空間」を提供します。
2	—	両方	**VTOC** – 慣例的にディスク全体を表します。このスライスのサイズは変更するべきではありません。 **EFI** – サイトの必要に応じてオプションスライスを定義します。
3	`/export` など	両方	サイトの必要に応じて定義可能なスライス (任意)。 クライアントシステムが必要とする代替オペレーティングシステムを格納するため、サーバー上で使用できます。
4		両方	サイトの必要に応じて定義可能なスライス (任意)。
5	`/opt` など	両方	サイトの必要に応じて定義可能なスライス (任意)。 システムに追加されるアプリケーションソフトウェアを格納します。インストール時に、`/opt` ファイルシステムにスライスが割り当てられていなければ、スライス 0 に `/opt` ディレクトリが作成されます。
6	`/usr`	両方	OS のコマンド (「実行可能ファイル」とも呼ぶ) を含みます。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ます。
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みます。
8	なし	なし	**VTOC** – GRUB ブート情報を含んでいます。 **EFI** – デフォルトで作成された予約済みスライス。VTOC の代替シリンダによく似た領域です。このスライスは変更または削除しないでください。

表 10–2 ディスクスライス (続き)

スライス	ファイルシステム	通常クライアントかサーバーのどちらにあるか	Comments
9 (**x86** のみ)	—	両方	**EFI** – 適用できません。 **VTOC** – 代替ディスクブロック用に予約された領域です。スライス 9 は代替セクタースライスと呼ばれます。

注 – VTOC ラベルの付いたディスクでは、スライスを変更したり、スライス 2 を使ってファイルシステムを格納したりしないでください。Solaris ボリュームマネージャー、Solaris Live Upgrade、installgrub などの製品は、スライス 2 が少しでも変更されると正しく動作しません。

raw データスライスの使用

ディスクラベルは、各ディスクのブロック 0 に格納されます。つまり、他社製データベースアプリケーションを使って raw データスライスを作成するときは、ブロック 0 で開始してはいけません。そのようにすると、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 データスライスは他社製のデータベースアプリケーションによって作成されることがあります。

- ブロック 0 (ディスクラベルが格納される領域)
- スライス 2 (VTOC ラベル付きディスク全体を表す)

複数のディスク上のスライス配置

十分な大きさのディスクであれば、1 台ですべてのスライスとそれに対応するファイルシステムを確保できますが、通常はシステムのスライスとファイルシステムを確保するために複数のディスクが使用されます。

注 – 1 つのスライスを複数のディスクに分割することはできません。ただし、複数のスワップスライスを別々のディスクに配置することはできます。

たとえば、1 台のディスクにルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムを入れ、別のディスクにユーザーデータを含む /export/home ファイルシステムやその他のファイルシステムを入れます。

複数のディスクを使用する場合、OS とスワップ領域が入っているディスク (つまり、ルート (/)、/usr ファイルシステム、およびスワップ領域用のスライスが入って

いるディスク) を、「システムディスク」と呼びます。システムディスク以外のディスクを、「二次ディスク」または「非システムディスク」と呼びます。

システムのファイルシステムを複数のディスクに入れると、システムをシャットダウンしたり OS をロードし直したりしなくても、二次ディスクのファイルシステムとスライスを変更できます。

また、複数のディスクを使用すると、入出力 (I/O) のパフォーマンスが改善されます。ディスク負荷を複数のディスクに分散すると、I/O のボトルネックを回避できます。

使用するスライスの決定

ディスクのファイルシステムを設定するときには、各スライスのサイズだけでなく、どのスライスを使用するかも決定します。どのように決定するかは、ディスクを接続するシステムの構成と、ディスクにインストールするソフトウェアによって異なります。

次のシステム構成があります。

- サーバー
- スタンドアロンシステム

システムの構成が異なれば、スライスの使用方法も異なります。次の表に例を挙げます。

表 10–3 システム構成とスライス

スライス	サーバー	スタンドアロンシステム
0	ルート	ルート
1	スワップ領域	スワップ領域
2	—	—
3	`/export`	—
6	`/usr`	`/usr`
7	`/export/home`	`/home`

システム構成の詳細は、『Solaris のシステム管理 (基本編)』の「システムタイプの概要」を参照してください。

注 – Solaris インストールユーティリティーは、インストール用に選択したソフトウェアに基づいてデフォルトのスライスサイズを表示します。

---

format ユーティリティー

手順や参照情報のセクションに進む前に、次の情報に目を通して format ユーティリティーの概要とその使用法を確認してください。

format ユーティリティーは、Solaris システム用にハードディスクドライブを用意するためのシステム管理ツールです。

次の表に、format ユーティリティーの機能とその利点を示します。

表 10–4　format ユーティリティーの機能と利点

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します	次の状態を報告します ■ 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します	修復処理に使用します
欠陥セクターを修復します	回復可能なエラーが発生したディスクドライブを製造元に返送しなくても、熟練した管理者なら修復できます
ディスクをフォーマットして、分析します	ディスク上でセクターを作成し、検査します
ディスクをパーティションに分割します	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します
ディスクにラベルを付けます	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みます

format ユーティリティーのオプションについては、第 15 章「format ユーティリティー (参照情報)」を参照してください。

format ユーティリティーを使用する場合

Solaris のインストール時に、Solaris インストールユーティリティーによってディスクドライブがパーティションに分割され、ラベルが付けられます。次のような場合に、format ユーティリティーを使用できます。

- スライス情報を表示する
- ディスクをパーティションに分割する
- 既存のシステムにディスクドライブを追加する
- ディスクドライブをフォーマットする
- ディスクにラベルを付ける

- ディスクドライブを修復する
- ディスクのエラーを分析する

システム管理者が format ユーティリティーを使用するのは、主にディスクをパーティションに分割するためです。これらの手順については、第 12 章「SPARC: ディスクの追加 (手順)」と第 13 章「x86: ディスクの追加 (手順)」を参照してください。

format ユーティリティーの使用上のガイドラインについては、次の節を参照してください。

format ユーティリティー使用上のガイドライン

表 10–5 format ユーティリティーのガイドライン

作業	ガイドライン	参照先
ディスクをフォーマットします。	■ 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ます。 ■ 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製造元が増えているので、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ます。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 ユーティリティーを使用しなくてもすむことがあります。 ■ ディスクを配置し直して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ことをお勧めします。不良セクターが自動的にマッピングし直されます。	220 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します。	■ 損傷したシステムディスクのデータは、バックアップメディアから復元しなければなりません。復元しなければ、インストールユーティリティー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りません。	241 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」、253 ページの「x86: システムディスクの接続方法」、また、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『Oracle Solaris 10 9/10 インストールガイド (基本編)』

表 10–5 format ユーティリティーのガイドライン (続き)

作業	ガイドライン	参照先
ディスクをスライスに分割します。	■ すでにスライスに分割されているディスクで、パーティションを再分割してラベルを付け直すと、既存のデータが失われます。 ■ ディスクのパーティションを分割し直して復元する前に、既存のデータをバックアップメディアにコピーする必要があります。	243 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 263 ページの「x86: ディスクスライスとディスクラベルを作成する方法」
既存のシステムに二次ディスクを追加します。	■ 二次ディスクをフォーマットし直すか、パーティションに分割し直す場合は、既存のデータをバックアップメディアから復元しなければなりません。	242 ページの「SPARC: 二次ディスクを接続してブートする方法」または 255 ページの「x86: 二次ディスクを接続してブートする方法」
ディスクドライブを修復します。	■ 顧客のサイトによっては、欠陥ドライブの修復ではなくドライブ自体の交換を希望する場合があります。サイトがディスクドライブの製造元と保守契約を結んでいる場合は、format ユーティリティーを使用してディスクドライブを修復する必要はありません。 ■ 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します。新しいコントローラは不良セクターをマップし直すので、システムを中断する必要はありません。 ■ 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ります。	234 ページの「欠陥セクターの修復」

ディスクのフォーマット

ほとんどの場合、ディスクは製造元または再販業者によってフォーマットされています。このため、ドライブをインストールするときにフォーマットし直す必要はありません。ディスクがフォーマットされているかどうかを判別するには、format ユーティリティーを使用します。詳細は、219 ページの「ディスクがフォーマット済みかを調べる方法」を参照してください。

ディスクがフォーマットされていない場合、format ユーティリティーを使用してフォーマットしてください。

ディスクのフォーマットでは、次の 2 つのステップが行われます。

- ディスクメディアを使用できるようにする。
- 表面解析に基づいてディスクの欠陥リストを作成する。

注意 – フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常は製造元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` ユーティリティーを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。詳細は、220 ページの「ディスクをフォーマットする方法」を参照してください。

データに利用できる合計ディスク容量のうち、ごくわずかな容量が欠陥情報とフォーマット情報の格納に使用されます。この容量はディスクのジオメトリによって異なり、使用年数がたち欠陥箇所が多くなるにつれて、少なくなります。

ディスクの種類とサイズに応じて、フォーマットは数分から数時間かかります。

ディスクをパーティションに分割する

`format` ユーティリティーは、主にシステム管理者がディスクをパーティションに分割する場合に使われます。手順を次に示します。

- どのスライスが必要かを決定します
- 各スライスまたはパーティションのサイズを決定します
- `format` ユーティリティーを使用してディスクをパーティションに分割します
- 新しいパーティション情報を使用してディスクにラベルを付けます
- パーティションごとにファイルシステムを作成します

ディスクをパーティションに分割するには、`format` ユーティリティーの `partition` メニューで modify コマンドを実行するのがもっとも簡単です。`modify` コマンドを使用すると、開始シリンダ境界を追跡しなくても、各パーティションのサイズを指定してパーティションを作成できます。`modify` コマンドを使用すると、「free hog」スライス内の残りのディスク領域も追跡できます。

パーティションテーブル関連の用語

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。パーティションテーブルは、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` ユーティリティーを使用して表示できます。次の表に、パーティションテーブル関連の用語を示します。

表 10–6 パーティションテーブル関連の用語

用語	値	説明
数値	0–7	**VTOC** – 0 - 7 の番号が付いたパーティションまたはスライス。 **EFI** – 0 - 6 の番号が付いたパーティションまたはスライス。
タグ	0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME 11=RESERVED	一般にこのパーティションにマウントされたファイルシステムを記述する数値。
フラグ	wm	パーティションは書き込み可能でマウント可能です。
	wu rm	パーティションは書き込み可能でマウントはできません。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す。ただし、mount コマンドでは「マウント不可」のフラグは検査されません。
	rm	パーティションは読み取り専用でマウント可能です。

パーティションのフラグとタグは必ず割り当てられるので、管理する必要はありません。

パーティションテーブルを表示する手順については、次の項目を参照してください。

- 211 ページの「パーティションテーブル情報の表示」
- 222 ページの「ディスクスライス情報を表示する方法」
- 226 ページの「ディスクラベルを検査する方法」

パーティションテーブル情報の表示

次の例は、format ユーティリティーを使って、74G バイトの VTOC ラベル付きディスクのパーティションテーブルを表示したものです。

```
Total disk cylinders available: 38756 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2083        4.00GB    (2081/0/0)    8390592
  1       swap    wu    2084 -  3124        2.00GB    (1041/0/0)    4197312
  2     backup    wm       0 - 38755       74.51GB    (38756/0/0) 156264192
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
```

```
  6 unassigned    wm       0               0         (0/0/0)             0
  7       home    wm    3125 - 38755       68.50GB    (35631/0/0) 143664192
  8       boot    wu       0 -     0        1.97MB    (1/0/0)          4032
  9 alternates    wu       1 -     2        3.94MB    (2/0/0)          8064

partition>
```

format ユーティリティーを使用して表示されるパーティションテーブルには、次の情報が含まれます。

列名	説明
Part	パーティションまたはスライスの番号。この列についての説明は、表 10–6 を参照してください。
Tag	パーティションのタグ。この列についての説明は、表 10–6 を参照してください。
Flag	パーティションのフラグ。この列についての説明は、表 10–6 を参照してください。
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します。EFI ラベル付きディスクでは表示されません。
Size	スライスのサイズを M バイト単位で示します。
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を示します。EFI ラベル付きディスクでは表示されません。
First Sector	**EFI** – 開始ブロック番号。VTOC ラベル付きディスクでは表示されません。
Last Sector	**EFI** – 終了ブロック番号。VTOC ラベル付きディスクでは表示されません。

次に、prtvtoc コマンドを使用して EFI ディスクラベルを表示した例を示します。

```
# prtvtoc /dev/rdsk/c4t1d0s0
* /dev/rdsk/c4t1d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
* 2576941056 sectors
* 2576940989 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector   Mount Directory
       0      2    00          34   629145600  629145633
       1      4    00   629145634   629145600 1258291233
       6      4    00  1258291234  1318633404 2576924637
```

```
       8     11    00  2576924638       16384 2576941021
```

prtvtoc コマンドの出力では、次の 3 つのセクションに情報が表示されます。

- Dimensions
- フラグ
- Partition テーブル

prtvtoc の列名	説明
Partition	パーティションまたはスライスの番号。この列についての説明は、表 10–6 を参照してください。
Tag	パーティションのタグ。この列についての説明は、表 10–6 を参照してください。
Flags	パーティションのフラグ。この列についての説明は、表 10–6 を参照してください。
First Sector	スライスの最初のセクターを示します。
Sector Count	スライス内の合計セクター数を示します。
Last Sector	スライスの最後のセクターを示します。
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します。

free hog スライスの使用方法

format ユーティリティーを使用して 1 つまたは複数のディスクスライスのサイズを変更するときには、サイズ変更操作に対応して拡大縮小する一時スライスを指定します。

このスライスは、スライスを拡大すると領域を「解放 (free)」し、スライスを圧縮すると放棄された領域を「回収 (hog)」します。このため、提供側のスライスを「free hog」と呼びます。

free hog スライスは、インストール時または format ユーティリティーの実行時にのみ存在します。日常の操作中に free hog スライスが継続して存在することはありません。

free hog スライスの使用方法については、243 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 263 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

◆ ◆ ◆ 第 11 章

11 ディスクの管理 (手順)

この章では、ディスク管理の手順について説明します。Oracle Solaris OS 上でディスクを管理する方法に精通している場合は、この章で説明する多くの内容を読み飛ばすことができます。

ディスク管理に関連した手順の詳細は、215 ページの「ディスクの管理 (作業マップ)」を参照してください。

ディスク管理の概要については、第 10 章「ディスクの管理 (概要)」を参照してください。

ディスクの管理 (作業マップ)

作業	説明	参照先
システム上のディスクを確認します。	システム上のディスクの種類がわからない場合は、`format` ユーティリティーを使用して確認します。	216 ページの「システム上のディスクを確認する方法」
ディスクをフォーマットします。	`format` ユーティリティーを使用して、ディスクがフォーマット済みかどうかを判断します。 ほとんどの場合、ディスクはフォーマット済みです。フォーマットする必要があれば、`format` を使用します。	219 ページの「ディスクがフォーマット済みかを調べる方法」 220 ページの「ディスクをフォーマットする方法」
スライス情報を表示します。	`format` ユーティリティーを使用してスライス情報を表示します。	222 ページの「ディスクスライス情報を表示する方法」

作業	説明	参照先
ディスクラベルの作成	format ユーティリティーを使用してディスクラベルを作成します。	224 ページの「ディスクラベルを作成する方法」
ディスクラベルを検査します。	prtvtoc コマンドを使用してディスクラベルを検査します。	226 ページの「ディスクラベルを検査する方法」
破損したディスクラベルを復元します。	システム障害または電源障害のために破損したディスクラベルの復元を試みます。	228 ページの「破損したディスクラベルを復元する方法」
format.dat のエントリを作成します。	他社製のディスクをサポートするために format.dat のエントリを作成します。	231 ページの「format.dat のエントリを作成する方法」
SCSI ディスクを自動構成します。	特定のドライブタイプが /etc/format.dat ファイルに記載されていない場合でも、ディスクデバイスモードセンスページの SCSI-2 仕様を利用して、SCSI ディスクを自動構成できます。	233 ページの「SCSI ドライブを自動構成する方法」
欠陥ディスクセクターを検出します。	format ユーティリティーを使用して欠陥ディスクセクターを調べます。	235 ページの「表面解析を使用して欠陥セクターを調べる方法」
欠陥ディスクセクターを修復します (必要な場合)。	format ユーティリティーを使用して欠陥ディスクセクターを修復します。	236 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 ユーティリティーを使用して、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 ユーティリティーを使用して、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 ユーティリティーの使用方法の詳細は、第 15 章「format ユーティリティー (参照情報)」を参照してください。

▼ システム上のディスクを確認する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 format ユーティリティーを使用して、システム上で認識されるディスクを確認します。

```
# format
```

format ユーティリティーは、AVAILABLE DISK SELECTIONS という見出しの下に、認識されるディスクのリストを表示します。

例 11–1 システム上のディスクを確認する

次に、単一のディスクが搭載されたシステムで format コマンドを実行したときの出力例を示します。

```
# format
AVAILABLE DISK SELECTIONS:
 0. c0t1d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
    /pci@1f,0/pci@1,1/scsi@2/sd@1,0
```

ディスクの物理デバイス名と論理デバイス名は、括弧 <> 内の商品名に対応しています。次の例を参照してください。この方法では、どの論理デバイス名がシステムに接続されたディスクを表しているかをすぐに識別できます。論理デバイス名と物理デバイス名については、84 ページの「デバイス名の命名規則」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 4 つのディスクを表示します。

```
# format /dev/rdsk/c2*
AVAILABLE DISK SELECTIONS:
       0. /dev/rdsk/c2t10d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
       1. /dev/rdsk/c2t11d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
       2. /dev/rdsk/c2t14d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@3,0/SUNW,fas@3,8800000/sd@e,0
       3. /dev/rdsk/c2t15d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@3,0/SUNW,fas@3,8800000/sd@f,0
Specify disk (enter its number):
```

次の例では、SPARC システム上のディスクを確認する方法を示します。

```
# format
0. c0t1d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
   /pci@1f,0/pci@1,1/scsi@2/sd@1,0
Specify disk (enter its number):
```

出力から、ディスク 0 (ターゲット 1) が 2 番目の SCSI ホストアダプタ (scsi@2) に接続され、この 2 番目の SCSI ホストアダプタが 2 番目の PCI インタフェース (/pci@1f0/pci@1,1...) に接続されていることがわかります。物理デバイス名と論理デバイス名は、ディスクの商品名 SUN36G に対応しています。

次の例では、x86 システム上のディスクを確認する方法を示します。

```
# format
AVAILABLE DISK SELECTIONS:
 0. c0d0 <DEFAULT cyl 615 alt 2 hd 64 sec 63>
     /pci@0,0/pci-ide@7,1/ata@0/cmdk@0,0
 1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
     /pci@0,0/pci-ide@7,1/ata@0/cmdk@1,0
 2. c1d0 <DEFAULT cyl 817 alt 2 hd 256 sec 63>
     /pci@0,0/pci-ide@7,1/ata@1/cmdk@0,0
Specify disk (enter its number):
```

出力から、ディスク 0 が最初の PCI ホストアダプタ (pci-ide@7...) に接続され、このアダプタが ATA インタフェース (ata...) に接続されていることがわかります。x86 ベースのシステムで、format の出力にディスクの商品名が含まれていない場合、次に示す手順に従って format ユーティリティーの type および label 機能を使用して、ディスクの商品名を含めることができます。

次に示す手順は、ディスクまたはシステムがアクティブでない状態で (つまり、インストール DVD またはネットワークからブートして) 実行する必要があります。また、手順の最後でディスクにラベルを付け直すことで、既存パーティションの情報やデータをすべて削除できます。

```
# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <DEFAULT cyl 49779 alt 2 hd 4 sec 720>
          /pci@0,0/pci1022,7450@a/pci17c2,10@4/sd@0,0
       1. c1t1d0 <DEFAULT cyl 8921 alt 2 hd 255 sec 63>
          /pci@0,0/pci1022,7450@a/pci17c2,10@4/sd@1,0
       2. c2t1d0 <SEAGATE-ST318404LSUN18G-4203-16.87GB>
          /pci@0,0/pci1022,7450@b/pci1000,10c0@1/sd@1,0
       3. c2t2d0 <SEAGATE-ST318404LSUN18G-4203-16.87GB>
          /pci@0,0/pci1022,7450@b/pci1000,10c0@1/sd@2,0
.
.
.
Specify disk (enter its number): 7
selecting c2t6d0
[disk formatted]
format> type
AVAILABLE DRIVE TYPES:
        0. Auto configure
        1. other
Specify disk type (enter its number)[1]: 0
c2t6d0: configured with capacity of 16.87GB
<IBM-DDYST1835SUN18G-S94N-16.87GB>
selecting c2t6d0
[disk formatted]
format> label
Ready to label disk, continue? yes
format> disk

AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <DEFAULT cyl 49779 alt 2 hd 4 sec 720>
```

```
          /pci@0,0/pci1022,7450@a/pci17c2,10@4/sd@0,0
       1. c1t1d0 <DEFAULT cyl 8921 alt 2 hd 255 sec 63>
.
.
.
       7. c2t6d0 <IBM-DDYST1835SUN18G-S94N-16.87GB>
          /pci@0,0/pci1022,7450@b/pci1000,10c0@1/sd@6,0
```

参考 format ユーティリティーがディスクを認識しない場合

- 第 12 章「SPARC: ディスクの追加 (手順)」または第 13 章「x86: ディスクの追加 (手順)」へ進みます。
- 231 ページの「format.dat のエントリの作成」へ進みます。
- 224 ページの「ディスクラベルを作成する方法」へ進みます。
- 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します。

ディスクのフォーマット

多くの場合、ディスクは製造元または再販業者によってフォーマットされています。通常は再フォーマットしなくてもドライブを取り付けることができます。

次の作業の前にディスクをフォーマットしておかなければなりません。

- データをディスクへ書き込みます。ただし、ほとんどのディスクはフォーマット済みです。
- Solaris インストールユーティリティーを使ってシステムをインストールします。

注意 – フォーマットは、ディスク上のデータを上書きする、破壊的なプロセスです。このため、通常は製造元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 ユーティリティーを使用して表面解析を実行できますが、データを破壊しないコマンドだけを使用するように注意してください。

▼ ディスクがフォーマット済みかを調べる方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 format ユーティリティーを起動します。

```
# format
```

番号付きのディスクのリストが表示されます。

3 確認するディスクの番号を入力します。

```
Specify disk (enter its number): 0
```

4 ディスクがフォーマット済みかを調べます。選択したディスクがフォーマット済みであれば、次のメッセージが表示されます。

```
[disk formatted]
```

例 11–2 ディスクがフォーマット済みかを調べる

次の例は、ディスク c1t0d0 がフォーマット済みであることを示しています。

```
# format /dev/rdsk/c1*
AVAILABLE DISK SELECTIONS:
       0. /dev/rdsk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@0,0
       1. /dev/rdsk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@1,0
       2. /dev/rdsk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@8,0
       3. /dev/rdsk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting /dev/rdsk/c1t0d0s0
[disk formatted]
```

▼ ディスクをフォーマットする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 format ユーティリティーを起動します。

```
# format
```

番号付きのディスクのリストが表示されます。

3 フォーマットするディスクの番号を入力します。

```
Specify disk (enter its number): 0
```

注意 – システムディスクを選択しないでください。システムディスクをフォーマットすると、OS やシステムディスク上のデータがすべて削除されます。

4 ディスクのフォーマットを開始するには、format> プロンプトで format と入力します。y と入力してコマンドを確認します。

```
format> format
Ready to format.  Formatting cannot be interrupted
and takes 23 minutes (estimated). Continue? yes
```

5 フォーマットが正常に行われたことを、次のメッセージによって確認します。

```
Beginning format. The current time Tue ABC xx xx:xx:xx xxxx

Formatting...
done

Verifying media...
        pass 0 - pattern = 0xc6dec6de
   2035/12/18

        pass 1 - pattern = 0x6db6db6d
   2035/12/18

Total of 0 defective blocks repaired.
```

6 format ユーティリティーを終了します。

```
format> quit
```

例 11–3 ディスクのフォーマット

次の例は、ディスク c0t6d0 のフォーマット方法を示します。

```
# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUNW18G cyl 7506 alt 2 hd 19 sec 248
          /pci@1f,0/pci@1,1/scsi@2/sd@0,0
       1. c0t1d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@1,0
       2. c0t2d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@2,0
       3. c0t3d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@3,0
       4. c0t4d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@4,0
       5. c0t5d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@5,0
       6. c0t6d0 <FUJITSU  MAN3367M SUN36G  1804 43d671f>
          /pci@1f,0/pci@1,1/scsi@2/sd@6,0
Specify disk (enter its number): 6
selecting c0t6d0
[disk formatted]
format> format
Ready to format.  Formatting cannot be interrupted
and takes 332 minutes (estimated). Continue? y
Beginning format. The current time is Mon Jul 12 10:52:10 2010
Formatting...
Verifying media...
Total of 0 defective blocks repaired.
format> quit
```

ディスクスライスの表示

format ユーティリティーを使用すると、ディスクに適切なディスクスライスがあるかどうかを検査できます。使用する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 ユーティリティーを使用してスライスを作成し直し、ディスクにラベルを付けます。ディスクスライスの作成方法については、243 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 263 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 ユーティリティーでは、「スライス」ではなく「パーティション」という用語を使用します。

▼ ディスクスライス情報を表示する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. format ユーティリティーを起動します。

   ```
   # format
   ```

   番号付きのディスクのリストが表示されます。
3. スライス情報を表示するディスクの番号を入力します。

   ```
   Specify disk (enter its number):1
   ```
4. partition メニューを選択します。

   ```
   format> partition
   ```
5. 選択されたディスクのスライス情報が表示されます。

   ```
   partition> print
   ```
6. format ユーティリティーを終了します。

   ```
   partition> q
   format> q
   #
   ```
7. 特定のスライスのタグとサイズについてスライス情報が表示されることを確認します。

   画面の出力に、スライスサイズが割り当てられていないことが示された場合は、ディスクにスライスがないものと思われます。

例 11–4 ディスクスライス情報を表示する

次に示すのは、VTOC ラベル付きディスクのスライス情報を表示する例です。

```
# format
Searching for disks...done
Specify disk (enter its number):1
Selecting c0t0d0
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 8892 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm    1110 - 4687        1.61GB    (0/3578/0) 3381210
  1       swap    wu       0 - 1109      512.00MB    (0/1110/0) 1048950
  2     backup    wm       0 - 8891        4.01GB    (0/8892/0) 8402940
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6 unassigned    wm       0               0         (0/0/0)          0
  7       home    wm    4688 - 8891        1.89GB    (0/4204/0) 3972780
partition> q
format> q
#
```

これらの例のスライス情報の詳細は、第 10 章「ディスクの管理 (概要)」を参照してください。

次に示すのは、EFI ラベル付きディスクのスライス情報を表示する例です。

```
# format
Searching for disks...done
Specify disk (enter its number): 9
selecting c4t1d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
partition> q
format> q
Part      Tag    Flag     First Sector          Size          Last Sector
  0       root    wm                34       300.00GB           629145633
  1        usr    wm         629145634       300.00GB          1258291233
  2 unassigned    wm                 0            0                0
  3 unassigned    wm                 0            0                0
  4 unassigned    wm                 0            0                0
  5 unassigned    wm                 0            0                0
  6        usr    wm        1258291234       628.77GB          2576924637
  8   reserved    wm        2576924638         8.00MB          2576941021
```

ディスクラベルの作成と検査

一般に、ディスクにラベルを付ける操作は、システムのインストール時、または新しいディスクスライスを作成するときに行います。ディスクラベルが破損したときは、新しくラベルを付け直す必要があります。ディスクラベルの破損は、電源障害が発生した場合などに起こります。

format ユーティリティーは、ラベルが付いていない SCSI ディスクを自動構成しようとします。format ユーティリティーがラベルの付いていないディスクを自動構成できる場合、次のようなメッセージが表示されます。

```
c0t0d1: configured with capacity of 4.00GB
```

ヒント – 複数のディスクに同じディスクラベルを付ける方法については、237 ページの「prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける」を参照してください。

▼ ディスクラベルを作成する方法

次の処理を行う手順について説明します。

- VTOC または EFI ラベルを使って、2T バイトのサイズのディスクにラベルを付ける。
- EFI ラベルを使って、2T バイトを超えるサイズのディスクにラベルを付ける。

2T バイト未満のディスクに EFI ラベルを付ける方法については、例 11–6 を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. format ユーティリティーを起動します。

   ```
   # format
   ```

   番号付きのディスクのリストが表示されます。
3. ラベルを付けたいディスクの番号を入力します。

   ```
   Specify disk (enter its number):1
   ```

   format ユーティリティー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。
4. 次のいずれかの方法を選択して、ディスクにラベルを付けます。

- ディスクが正常に構成されていて、ラベルが付いていない場合は、手順 5 に進みます。

  format ユーティリティーにより、ディスクラベルを付けるかを尋ねるプロンプトが表示されます。
- ラベル付きディスクのディスクタイプを変更する場合、またはこのディスクを format ユーティリティーで自動構成できなかった場合は、手順 6 に進んでディスクタイプを設定し、ラベルを付けます。

5 Label it now? プロンプトで y と入力して、ディスクにラベルを付けます。

```
Disk not labeled. Label it now? y
```

これでディスクラベルが作成されました。手順 10 に進んで format ユーティリティーを終了します。

6 format> プロンプトで type と入力します。

```
format> type
```

Available Drive Types メニューが表示されます。

7 ディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number)[12]: 12
```

または、0 を選択して SCSI-2 ディスクを自動構成します。詳細は、233 ページの「SCSI ドライブを自動構成する方法」を参照してください。

8 ディスクラベルの作成ディスクにラベルが付いていない場合は、次のメッセージが表示されます。

```
Disk not labeled. Label it now? y
```

ディスクラベルが付いている場合は、次のメッセージが表示されます。

```
Ready to label disk, continue? y
```

9 ディスクラベルを検査します。

```
format> verify
```

10 format ユーティリティーを終了します。

```
format> q
#
```

例 11–5 ディスクラベルを作成する

次の例では、1.05G バイトのディスクを自動構成してラベルを付ける方法を示します。

```
# format
    c1t0d0: configured with capacity of 1002.09MB

AVAILABLE DISK SELECTIONS:
       0. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
       1. c1t0d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
Specify disk (enter its number): 1
Disk not labeled.  Label it now?  yes
format> verify
format> q
#
```

例 11–6 2T バイト以下のサイズのディスクに EFI ラベルを付ける

次に、format -e コマンドを使って、2T バイト以下のディスクに EFI ラベルを付ける例を示します。階層化されたソフトウェア製品が EFI ラベル付きディスクのシステムでも動作することを確認しておいてください。EFI ラベルの一般的な制限事項については、199 ページの「EFI ディスクラベルの制限」を参照してください。

```
# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       1. c1t0d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@0,0
       2. c1t1d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@1,0
       3. c1t8d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@8,0
       4. c1t9d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 4
selecting c1t9d0
[disk formatted]
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 1
Ready to label disk, continue? yes
format> quit
```

▼ ディスクラベルを検査する方法

ディスクラベル情報の検査には、prtvtoc コマンドを使用します。ディスクラベルと prtvtoc コマンドの出力結果の詳しい説明については、第 10 章「ディスクの管理 (概要)」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. ディスクラベル情報を表示します。

```
# prtvtoc /dev/rdsk/device-name
```

device-name には、検査する raw ディスクデバイスを指定してください。

例 11–7 ディスクラベルを検査する

次に示すのは、VTOC ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdsk/c0t0d0s0
* /dev/rdsk/c0t0d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*      15 tracks/cylinder
*     945 sectors/cylinder
*    8894 cylinders
*    8892 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00    1048950   3381210   4430159   /
       1      3    01          0   1048950   1048949
       2      5    00          0   8402940   8402939
       7      8    00    4430160   3972780   8402939   /export/home
```

次に示すのは、EFI ラベル付きディスクのディスクラベル情報を表示する例です。

```
# prtvtoc /dev/rdsk/c3t1d0s0
* /dev/rdsk/c3t1d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
* 2479267840 sectors
* 2479267773 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First      Sector       Last
* Partition  Tag  Flags    Sector     Count        Sector  Mount Directory
       0      2    00          34      262144      262177
       1      3    01      262178      262144      524321
       6      4    00      524322  2478727100  2479251421
       8     11    00  2479251422       16384  2479267805
```

破損したディスクラベルの復元

電源障害やシステム障害が原因で、ディスクが認識されなくなることがあります。ただし、ディスクラベルが破損しても、スライス情報やディスクのデータを作り直したり、復元したりする必要がない場合もあります。

破損したディスクラベルを復元する作業の最初の手順は、正しいジオメトリとディスクタイプ情報を使用してディスクにラベルを付けることです。この作業は、通常のディスクラベル作成方法 (自動構成またはディスクタイプの手動指定) で実行できます。

`format` ユーティリティー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の Solaris DVD 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。

   システムのブートについては、『Solaris のシステム管理 (基本編)』の第 12 章「Oracle Solaris システムのブート (手順)」または『Solaris のシステム管理 (基本編)』の「GRUB を使用して x86 システムをブートする (作業マップ)」を参照してください。

2. ディスクにラベルを付け直します。

   ```
   # format
   ```

   `format` ユーティリティーは、ラベルが付いていない SCSI ディスクを自動構成しようとします。破損し、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、次のメッセージが表示されます。

   c`w`*t*`x`*d*`y`: `configured with capacity of` *abc*`MB`

   次に、システム上のディスクの番号付きリストが表示されます。

3. 復元するディスクの番号を入力します。

   ```
   Specify disk (enter its number): 1
   ```

4. 次のいずれかを選択して、ディスクラベルの作成方法を決定します。

   - ディスクが正常に構成された場合は、手順 5 と 6 を実行します。次に、手順 12 に進みます。

- ディスクが正常に構成されなかった場合は、手順 7 - 11 を実行します。次に、手順 12 に進みます。

5 バックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk or
use the 'backup' command.
Backup label contents:
Volume name = <        >
ascii name  = <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
pcyl        = 2038
ncyl        = 2036
acyl        =    2
nhead       =   14
nsect       =   72
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -  300      148.15MB    (301/0/0)   303408
  1       swap    wu     301 -  524      110.25MB    (224/0/0)   225792
  2     backup    wm       0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6        usr    wm     525 - 2035      743.70MB    (1511/0/0) 1523088
  7 unassigned    wm       0               0         (0/0/0)          0
```

6 format ユーティリティーがバックアップラベルを検出することができ、その内容が適切な場合は、backup コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

これで、ディスクラベルが復元されました。手順 12 へ進みます。

7 format ユーティリティーでディスクを自動構成できなかった場合は、type コマンドを使用してディスクタイプを指定します。

```
format> type
```

Available Drives Type メニューが表示されます。

8 ディスクを自動構成するには、0 を選択します。または、ディスクタイプの候補のリストからディスクタイプを選択します。

```
Specify disk type (enter its number)[12]: 12
```

9 ディスクが正常に自動構成された場合は、ディスクラベルを作成するかどうか尋ねるプロンプトが表示されたときに no と応答します。

```
Disk not labeled.  Label it now?  no
```

10 verify コマンドを使用してバックアップラベルを検索します。

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
.
.
.
```

11 format ユーティリティーがバックアップラベルを検出することができ、その内容が適切な場合は、backup コマンドを実行して、ディスクにバックアップラベルを付けます。

```
format> backup
Disk has a primary label, still continue? y
Searching for backup labels...found.
Restoring primary label
```

これで、ディスクラベルが復元されました。

12 format ユーティリティーを終了します。

```
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 21 章「UFS ファイルシステムの整合性検査 (手順)」を参照してください。

他社製のディスクの追加

Solaris OS では、他社製の多数のディスクがサポートされます。ただし、これらのディスクを認識させるためには、デバイスドライバか format.dat のエントリのいずれか 1 つ、またはその両方を追加しなければならない場合があります。ディスク追加時のその他のオプションを次に示します。

- SCSI ディスクを追加する場合、format ユーティリティーの自動構成機能の使用を試みることができます。詳細は、232 ページの「SCSI ディスクドライブの自動構成」を参照してください。
- PCI、SCSI、または USB ディスクのホットプラグを試みることもできます。詳細は、第 5 章「デバイスの管理 (概要と手順)」を参照してください。

他社製のディスクが標準の SunOS 互換デバイスドライバで機能するように設計されている場合は、適切な format.dat エントリを作成するだけで、ディスクは format ユーティリティーに認識されるはずです。それ以外の場合は、そのディスクをサポートするために他社製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 ユーティリティーが他社製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 ユーティリティー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ディスクフォーマットプログラムを提供しているはずです。

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 ユーティリティーを起動し、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。

この節の説明に従って、不足しているソフトウェアを追加します。その後、第 12 章「SPARC: ディスクの追加 (手順)」または第 13 章「x86: ディスクの追加 (手順)」を参照して、システムディスクまたは二次ディスクを構成します。

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 ディスクには format.dat のエントリは不要です。再構成ブート時にディスクに電源が投入されていれば、format ユーティリティーは SCSI-2 ドライバを自動的に構成します。SCSI ディスクドライブを自動構成する手順については、233 ページの「SCSI ドライブを自動構成する方法」を参照してください。

ディスクが認識されない場合は、テキストエディタを使用して format.dat にディスクのエントリを作成します。作業を始める前に、ディスクとそのコントローラに関連するすべての技術仕様を収集する必要があります。この情報はディスクと一緒に提供されているはずです。提供されない場合は、ディスク製造元または購入先に問い合わせてください。

▼ format.dat のエントリを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. /etc/format.dat ファイルのコピーを作成します。

```
# cp /etc/format.dat /etc/format.dat.gen
```

3 /etc/format.datファイルに他社製のディスクのエントリを追加します。

第15章「formatユーティリティー(参照情報)」で説明しているformat.dat情報を参照してください。

また、ディスクのハードウェア製品マニュアルを参照して、必要な情報を収集してください。

SCSIディスクドライブの自動構成

/etc/format.datファイルに特定のドライブタイプが含まれていない場合でも、formatユーティリティーはSCSIディスクドライブを自動的に構成します。この機能を使って、ディスクデバイスモードセンスページのために、SCSI-2仕様に準拠するディスクドライブをフォーマットしたり、スライスを作成したり、ラベルを付けたりできます。

次のような方法でも、ディスクを追加できます。

- SCSIディスクを追加する場合、formatユーティリティーの自動構成機能の使用を試みることができます。
- PCI、SCSI、またはUSBディスクのホットプラグを試みることもできます。詳細は、第5章「デバイスの管理(概要と手順)」を参照してください。

自動構成を使用してSCSIドライブを構成する場合、次の手順を実行します。

- システムのシャットダウン
- 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ユーティリティーを使用して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ユーティリティーを呼び出すと、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、233ページの「SCSIドライブを自動構成する方法」を参照してください。

1.3GバイトのSCSIディスクドライブに関して、formatユーティリティーが表示するパーティションテーブルの例を、次に示します。

```
Part      Tag    Flag     Cylinders      Size            Blocks
  0       root    wm       0 -   96    64.41MB       (97/0/0)
  1       swap    wu      97 -  289   128.16MB      (193/0/0)
  2     backup    wu       0 - 1964     1.27GB    (1965/0/0)
  6        usr    wm     290 - 1964     1.09GB    (1675/0/0)
```

▼ SCSIドライブを自動構成する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 システムのブート時に読み込まれる /reconfigure ファイルを作成します。

```
# touch /reconfigure
```

3 システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

-i0 システムを init レベル 0 (電源切断) にします。

-g*n* ログインしているユーザーに、*n* 秒後にシステムのシャットダウンを開始することを通知します。

-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します。

システムのシャットダウン後に、ok プロンプトが表示されます。

4 システムとすべての外部周辺デバイスの電源を切ります。

5 追加しようとするディスクに、システム上のほか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、ディスクの裏側には、このための小さいスイッチがあります。

6 ディスクをシステムに接続して、その物理的接続を確認します。

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。

システムがブートし、ログインプロンプトが表示されます。

9 スーパーユーザーになるか、同等の役割を引き受けて、再度ログインします。

10 format ユーティリティーを起動して、自動構成するディスクを選択します。

```
# format
Searching for disks...done
c1t0d0: configured with capacity of 1002.09MB
AVAILABLE DISK SELECTIONS:
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 1
```

11 ディスクにラベルを付けるかどうかを確認するプロンプトが表示されたら、yes と入力します。

y と入力すると、SCSI 自動構成機能により、ディスクラベルの生成およびディスクへの書き込みが実行されます。

```
Disk not labeled. Label it now? y
```

12 ディスクラベルを検査します。

```
format> verify
```

13 format ユーティリティーを終了します。

```
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、次の手順に従って修復できます。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 ユーティリティーの解析機能については、308 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。セクターを正確に検出する方法については、235 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

  次のようなディスクエラー関連のコンソールメッセージが表示される場合があります。

  ```
  WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):
      Error for command 'read' Error Level: Retryable
      Requested Block 126, Error Block: 179
      Sense Key: Media Error
      Vendor 'name':
      ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
  ```

  このメッセージから、ブロック 179 に欠陥があることがわかります。欠陥ブロックを再配置するには、format ユーティリティーの repair コマンドを使用します。または、repair オプションを有効にして analyze コマンドを実行します。

▼ 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るか、同等の役割を引き受けます。

2 欠陥セクターの存在するスライス内のファイルシステムをマウント解除します。

```
# umount /dev/dsk/device-name
```

詳細は、mount(1M) のマニュアルページを参照してください。

3 format ユーティリティーを起動します。

```
# format
```

4 調べるディスクを選択します。

```
Specify disk (enter its number):1
selecting c0t2d0:
[disk formatted]
Warning: Current Disk has mounted partitions.
```

5 analyze メニューを選択します。

```
format> analyze
```

6 analyze> プロンプトで setup と入力して、解析パラメータを設定します。

次のパラメータを使用してください。

```
analyze> setup
Analyze entire disk [yes]? n
Enter starting block number [0, 0/0/0]: 12330
Enter ending block number [2052287, 2035/13/71]: 12360
Loop continuously [no]? y
Repair defective blocks [yes]? n
Stop after first error [no]? n
Use random bit patterns [no]? n
Enter number of blocks per transfer [126, 0/1/54]: 1
Verify media after formatting [yes]? y
Enable extended messages [no]? n
Restore defect list [yes]? y
Create defect label [yes]? y
```

7 read コマンドを使って欠陥を検出します。

```
analyze> read
Ready to analyze (won't harm SunOS). This takes a long time,
but is interruptible with Control-C. Continue? y
        pass 0
   2035/12/1825/7/24
        pass 1
Block 12354  (18/4/18), Corrected media error (hard data ecc)
   25/7/24
^C
Total of 1 defective blocks repaired.
```

▼ 欠陥セクターを修復する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. format ユーティリティーを起動します。

```
# format
```

3. 欠陥セクターの存在するディスクを選択します。

```
Specify disk (enter its number): 1
selecting c0t3d0
[disk formatted]
format>
```

4. repair コマンドを選択します。

```
format> repair
```

5. 欠陥ブロック番号を入力します。

```
Enter absolute block number of defect: 12354
   Ready to repair defect, continue? y
   Repairing block 12354 (18/4/18)...ok.
format>
```

欠陥セクターの特定に使う形式がわからない場合は、235 ページの「表面解析を使用して欠陥セクターを調べる方法」を参照してください。

ディスク管理のヒント

次のヒントに従って、ディスクの管理効率を高めることができます。

format セッションのデバッグ

format -M コマンドを実行して、ATA および SCSI デバイス用の拡張および診断メッセージを有効にします。

例 11–8 format セッションのデバッグ

この例の Inquiry: の下の数値の列は、数値の右側に表示される inquiry データの 16 進数値です。

```
# format -M
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
```

例 11–8 format セッションのデバッグ (続き)

```
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 0
selecting c0t3d0
[disk formatted]
format> inquiry
Inquiry:
00 00 02 02 8f 00 00 12 53 45 41 47 41 54 45 20     ........NAME....
53 54 31 31 32 30 30 4e 20 53 55 4e 31 2e 30 35     ST11200N SUN1.05
38 33 35 38 30 30 30 33 30 32 30 39 00 00 00 00     835800030209....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 43 6f 70 79 72 69 67 68 74 20 28 63 29 20 31     .Copyright (c) 1
39 39 32 20 53 65 61 67 61 74 65 20 41 6c 6c 20     992 NAME    All
72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 20     rights reserved
30 30 30                                            000
Vendor:   name
Product:  ST11200N SUN1.05
Revision: 8358
format>
```

prtvtoc と fmthard コマンドを使用して複数のディスクにラベルを付ける

prtvtoc コマンドと fmthard コマンドを使用して、同じディスクジオメトリを持つ複数のディスクにラベルを付けます。

この for ループをスクリプト内で使用して、1 台のディスクからディスクラベルをコピーし、複数のディスク上で複製します。

```
# for i in x y z
> do
> prtvtoc /dev/rdsk/cwtxdysz | fmthard -s - /dev/rdsk/cwt${i}d0s2
> done
```

例 11–9 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 からほかの 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/rdsk/c2t0d0s0 | fmthard -s - /dev/rdsk/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.
```

例 11–9 複数のディスクにラベルを付ける (続き)

```
fmthard:  New volume table of contents now in place.
#
```

◆ ◆ ◆ 第 12 章

SPARC: ディスクの追加 (手順)

この章では、SPARC システムにディスクを追加する手順について説明します。

SPARC システムへのディスクの追加に関連した手順については、239 ページの「SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

ディスク管理の概要については、第 10 章「ディスクの管理 (概要)」を参照してください。x86 システムにディスクを追加する手順については、第 13 章「x86: ディスクの追加 (手順)」を参照してください。

ZFS ファイルシステムでも、記述された手順に従ってディスクを追加できますが、ディスクやディスクスライスへの直接のマッピングは行われません。ZFS ファイルシステムを作成する前に、ZFS ストレージプールを作成する必要があります。詳細については、『Oracle Solaris ZFS 管理ガイド』を参照してください。

SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)

次の作業マップは、SPARC システムにディスクを追加する手順を示します。

作業	説明	参照先
1. ディスクを接続してブートします。	システムディスク 新しいディスクを接続して、ローカルまたはリモートの Solaris DVD からブートします。	241 ページの「SPARC: システムディスクを接続してブートする方法」

作業	説明	参照先
	二次ディスク 新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行します。	242 ページの「SPARC: 二次ディスクを接続してブートする方法」
2. スライスとディスクラベルを作成します。	ディスクスライスを作成してディスクにラベルを付けます (ディスク製造元により実行されていない場合)。	243 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」
3. ファイルシステムを作成します。	`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します。システムディスクの場合はルート (/) または `/usr` ファイルシステム、あるいはその両方を作成する必要があります。	248 ページの「SPARC: UFS ファイルシステムを作成する方法」
4. UFS ファイルシステムを復元します。	システムディスク上にルート (/) または `/usr` ファイルシステム、あるいはその両方を復元します。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します。	第 26 章「UFS ファイルとファイルシステムの復元 (手順)」
5. ブートブロックをインストールします。	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します。	249 ページの「SPARC: システムディスクにブートブロックをインストールする方法」

SPARC: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと `/usr` ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Oracle Solaris OS 全体を再インストールします。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元します。

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。ディスク容量を増や

すために、二次ディスクを追加できます。または、損傷を受けた二次ディスクを交換できます。システム上の二次ディスクを交換すると、古いディスクのデータを新しいディスク上に復元できます。

▼ SPARC: システムディスクを接続してブートする方法

この手順は、システムがシャットダウンされていることを前提としています。

1. 損傷したシステムディスクをシステムから外します。
2. 追加しようとするディスクに、システム上のほか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、ディスクの裏側には、このための小さいスイッチがあります。
3. 交換用のディスクをシステムに正しく接続します。
   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. ローカルの **Solaris DVD** またはリモートの **Solaris DVD** (ネットワーク経由) のどちらでブートするかに応じて、次の表の手順で操作します。

ブート方法	動作
ローカルドライブの Solaris DVD	1. ドライブに Solaris DVD が入っていることを確認します。 2. メディアからシングルユーザーモードでブートします。 ok **boot cdrom -s**
ネットワーク経由	ネットワークからシングルユーザーモードでブートします。 ok **boot net -s**

数分後に、スーパーユーザーのプロンプト (#) が表示されます。

参考 システムディスクを接続し、ブートしたあとの作業

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。243 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

EFI ディスクラベル付きのディスクを追加する場合、その詳細は、198 ページの「EFI ディスクラベル」を参照してください。

一般に、最近のバスタイプのほとんどで、ホットプラグ機能がサポートされています。システムのバスタイプでホットプラグ機能がサポートされている場合は、次の手順 2 または手順 3 を行う必要がない場合があります。

ホットプラグ対応デバイスの詳細は、第 6 章「デバイスの動的構成 (手順)」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. (省略可能) ディスクタイプが **Solaris** ソフトウェアでサポートされていない場合は、ハードウェアに添付された説明書に従って、そのディスクのデバイスドライバを追加します。

   ディスク用の `format.dat` エントリを作成する方法については、231 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. (省略可能) 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、またはあと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検査します。

4. システムをシャットダウンします。

   ```
   # shutdown -i0 -gn -y
   ```

   `-i0` システムを実行レベル 0 (電源切断状態) にします。

   `-gn` ログインしているユーザーに、*n* 秒後にシステムのシャットダウンを開始することを通知します。

   `-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します。

   Oracle Solaris OS のシャットダウン後に、`ok` プロンプトが表示されます。

5. システムとすべての外部周辺デバイスの電源を切ります。

6. 追加しようとするディスクに、システム上のほか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、ディスクの裏側には、このための小さいスイッチがあります。

7 ディスクをシステムに接続して、その物理的接続を確認します。
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が表示されます。

参考 二次ディスクを接続し、ブートしたあとの作業

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。243 ページの「SPARC: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ SPARC: ディスクスライスとディスクラベルを作成する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 format ユーティリティーを起動します。

```
# format
```

利用可能なディスクの番号付きリストが表示されます。詳細は、format(1M) のマニュアルページを参照してください。

3 パーティションの再分割を実行するディスクの番号を入力します。

```
Specify disk (enter its number): disk-number
```

disk-number は、パーティションの再分割を実行するディスクの番号です。

4 partition メニューを選択します。

```
format> partition
```

5 現在のパーティション (スライス) テーブルを表示します。

```
partition> print
```

6 変更作業を開始します。

```
partition> modify
```

7 ディスクをすべて **free hog** に設定します。

```
Choose base (enter number) [0]?1
```

free hog スライスの詳細は、213 ページの「free hog スライスの使用方法」を参照してください。

8 続行するかどうかを尋ねるプロンプトが表示されたら、y と応答して新しいパーティションテーブルを作成します。

```
Do you wish to continue creating a new partition table based on
above table[yes]? y
```

9 プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0) およびスワップ (スライス 1) (必須)
- `/usr` (スライス 6)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。

ディスクスライスの作成例については、例 12–1 を参照してください。

10 プロンプトが表示されたら y と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? y
```

現在のパーティションテーブルが希望どおりでないために変更する場合は、no と応答して手順 6 に戻ります。

11 パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

partition-name には、新しいパーティションテーブルの名前を指定します。

12 新しいディスク上でスライスの割り当てが終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13 `partition` メニューを終了します。

```
partition> q
```

14 ディスクラベルを検査します。

```
format> verify
```

15 `format` ユーティリティーを終了します。

```
format> q
```

例 12–1 SPARC: システムディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティーを使用して 18G バイトのディスクを 3 つのスライスに分割します。 各スライスをルート (/) ファイルシステム、スワップ領域、/usr ファイルシステムに割り当てます。

```
# format
AVAILABLE DISK SELECTIONS:
       0. /dev/rdsk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@0,0
       1. /dev/rdsk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@1,0
       2. /dev/rdsk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@8,0
       3. /dev/rdsk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting c1t0d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)           0
  1       swap    wu       0               0         (0/0/0)           0
  2     backup    wu       0 - 7505       16.86GB    (7506/0/0) 35368272
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0

Choose base (enter number) [0]? 1
table based on above table[yes]? yes
Free Hog partition[6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 - 1780        4.00GB    (1781/0/0)  8392072
  1       swap    wu    1781 - 3561        4.00GB    (1781/0/0)  8392072
  2     backup    wu       0 - 7505       16.86GB    (7506/0/0) 35368272
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm    3562 - 7505        8.86GB    (3944/0/0) 18584128
  7 unassigned    wm       0               0         (0/0/0)           0

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "disk0"
```

```
Ready to label disk, continue? yes
partition> quit
format> verify
format> quit
```

例 12–2 SPARC: 二次ディスクのディスクスライスとラベルを作成する

次の例では、format ユーティリティーを使用して 18G バイトのディスクを 1 つのスライスに分割し、それを /export/home ファイルシステムに割り当てます。

```
# format /dev/rdsk/c1*
AVAILABLE DISK SELECTIONS:
       0. /dev/rdsk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@0,0
       1. /dev/rdsk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@1,0
       2. /dev/rdsk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@8,0
       3. /dev/rdsk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 1
selecting c1t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)           0
  1       swap    wu       0               0         (0/0/0)           0
  2     backup    wu       0 - 7505       16.86GB    (7506/0/0) 35368272
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0

Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 7
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)           0
  1       swap    wu       0               0         (0/0/0)           0
  2     backup    wu       0 - 7505       16.86GB    (7506/0/0) 35368272
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
```

```
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0 - 7505       16.86GB    (7506/0/0) 35368272
Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "home"
Ready to label disk, continue? y
partition> q
format> verify
format> q
#
```

次の例では、format ユーティリティーを使って、1.15T バイトの EFI ラベル付きディスクを 3 つのスライスに分割します。

```
# format
.
.
.
partition> modify
Select partitioning base:
        0. Current partition table (original)
        1. All Free Hog
Choose base (enter number) [0]? 1
Part      Tag    Flag     First Sector          Size          Last Sector
  0       root    wm                 0            0               0
  1        usr    wm                 0            0               0
  2 unassigned    wm                 0            0               0
  3 unassigned    wm                 0            0               0
  4 unassigned    wm                 0            0               0
  5 unassigned    wm                 0            0               0
  6        usr    wm                 0            0               0
  8   reserved    wm        2576924638         8.00MB          2576941021
Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition 0 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 1 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 2 [0b, 34e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 3 [0b, 838860834e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 5 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Enter size of partition 6 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Part      Tag    Flag     First Sector          Size          Last Sector
  0 unassigned    wm                 0            0               0
  1 unassigned    wm                 0            0               0
  2        usr    wm                34       400.00GB          838860833
  3        usr    wm         838860834       400.00GB          1677721633
  4        usr    wm        1677721634       428.77GB          2576924637
  5 unassigned    wm                 0            0               0
  6 unassigned    wm                 0            0               0
  8   reserved    wm        2576924638         8.00MB          2576941021
Ready to label disk, continue? yes

partition> q
```

参考 ディスクスライスとディスクラベルを作成したあとの作業

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。248 ページの「SPARC: UFS ファイルシステムを作成する方法」を参照してください。

▼ SPARC: UFS ファイルシステムを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. スライスごとにファイルシステムを作成します。

   ```
   # newfs /dev/rdsk/cwtxdysz
   ```

   /dev/rdsk/c*wtxdysx* は、作成するファイルシステムの raw デバイスです。

   newfs コマンドの詳細は、第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」または newfs(1M) のマニュアルページを参照してください。
3. 新しいファイルシステムをマウントして、確認します。

   ```
   # mount /dev/dsk/cwtxdysz /mnt
   # ls
   lost+found
   ```

参考 UFS ファイルシステムを作成したあとの作業

- システムディスク – ディスク上にルート (/) と /usr のファイルシステムを復元する必要があります。
  - 第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。
  - ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールします。249 ページの「SPARC: システムディスクにブートブロックをインストールする方法」を参照してください。
- 二次ディスク – 新しいディスク上にファイルシステムを復元しなければならないことがあります。第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が終わったことになります。
- ユーザーがファイルシステムを利用できるようにする方法については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

▼ SPARC: システムディスクにブートブロックをインストールする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. システムディスクにブートブロックをインストールします。

   UFS ルートファイルシステムの場合:

   ```
   # installboot /usr/platform/‘uname -i‘/lib/fs/ufs/bootblk
   /dev/rdsk/cwtxdys0
   ```

   `/usr/platform/‘uname -i‘/lib/fs /ufs/bootblk`
   これは、ブートブロックコードです。

   `/dev/rdsk/`*cwtxdy* `s0`
   これは、ルート (/) ファイルシステムの raw デバイスです。

   ZFS ルートファイルシステムの場合:

   ```
   # installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk
   /dev/rdsk/cwtxdys0
   ```

   `/usr/platform/‘uname -i‘/lib/fs/zfs/bootblk`
   これは、ブートブロックコードです。

   `/dev/rdsk/`*cwtxdy* `s0`
   これは、ルート (/) ファイルシステムの raw デバイスです。

   詳細は、`installboot(1M)` のマニュアルページを参照してください。

3. システムをリブートし、レベル **3** で実行することによって、ブートブロックがインストールされていることを確認します。

   ```
   # init 6
   ```

例 12–3 SPARC: システムディスクにブートブロックをインストールする

次の例は、UFS 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方法を示しています。

```
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk
/dev/rdsk/c0t0d0s0
```

次の例は、ZFS 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方法を示しています。

```
# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk /dev/rdsk/c0t1d0s0
```

◆ ◆ ◆ 第 13 章

13 x86: ディスクの追加 (手順)

この章では、x86 システムにディスクを追加する手順について説明します。

x86 システムへのディスク追加に関連した手順については、251 ページの「x86: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

ディスク管理の概要については、第 10 章「ディスクの管理 (概要)」を参照してください。SPARC システムへディスクを追加する手順については、第 12 章「SPARC: ディスクの追加 (手順)」を参照してください。

ZFS ファイルシステムでも、記述された手順に従ってディスクを追加したり、`fdisk` パーティションを作成したりできますが、ディスクやディスクスライスへの直接のマッピングは行われません。ZFS ファイルシステムを作成する前に、ZFS ストレージプールを作成する必要があります。詳細については、『Oracle Solaris ZFS 管理ガイド』を参照してください。

x86: システムディスクまたは二次ディスクの追加 (作業マップ)

次の作業マップは、x86 システムにディスクを追加する手順を示します。

作業	説明	参照先
1. ディスクを接続してブートします。	システムディスク 新しいディスクを接続して、ローカルまたはリモートの Solaris DVD からブートします。	253 ページの「x86: システムディスクの接続方法」

作業	説明	参照先
	二次ディスク 新しいディスクを接続し、システムでディスクが認識されるように再構成ブートを実行します。	255 ページの「x86: 二次ディスクを接続してブートする方法」
2. (省略可能) `fdisk` パーティション識別子を変更します。	x86 システムにおける Solaris 10 `fdisk` パーティション識別子が、130 (`0x82`) から 191 (`0xbf`) に変更されました。 `fdisk` の新しいメニューオプションを使用すると、新しい識別子と古い識別子を切り替えられます。	254 ページの「Solaris `fdisk` 識別子を変更する方法」
3. スライスとディスクラベルを作成します。	ディスクスライスを作成してディスクにラベルを付けます (ディスク製造元により実行されていない場合)。	258 ページの「x86: Solaris `fdisk` パーティションを作成する方法」と 263 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」
4. ファイルシステムを作成します。	`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します。システムディスクの場合はルート (`/`) または `/usr` ファイルシステム、あるいはその両方を作成する必要があります。	264 ページの「x86: ファイルシステムを作成する方法」
5. UFS ファイルシステムを復元します。	システムディスク上にルート (`/`) または `/usr` ファイルシステム (あるいは両方) を復元します。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します。	第 26 章「UFS ファイルとファイルシステムの復元 (手順)」
6. ブートブロックをインストールします。	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します。	265 ページの「x86: システムディスクにブートブロックをインストールする方法」

x86: システムディスクまたは二次ディスクの追加

システムディスクには、ルート (/) ファイルシステムと /usr ファイルシステムのうちのいずれか、またはその両方が入っています。この 2 つのファイルシステムのどちらかが入っているディスクが損傷した場合、復元方法は次の 2 つがあります。

- Oracle Solaris OS 全体を再インストールします。
- システムディスクを交換し、ファイルシステムをバックアップメディアから復元します。

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。ディスク容量を増やすために、二次ディスクを追加できます。または、損傷を受け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: システムディスクの接続方法

この手順は、オペレーティング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、システム上のほか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、ディスクの裏側には、このための小さいスイッチがあります。
3. 交換用のシステムディスクをシステムに正しく接続します。

   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. システムをブートします。

   この手順では、GRUB の Solaris フェイルセーフブートオプションからブートしていることを前提としています。

   a. `Press any key to reboot` プロンプトが表示されたら、任意のキーを押してシステムをリブートします。システムがシャットダウンされている場合は、リセットボタンを押してシステムを再起動します。

      数分経過すると、GRUB メニューが表示されます。

   b. 矢印キーを使って **Solaris** フェイルセーフブートオプションを選択します。

   c. **Return** キーを押します。

d. 画面に `Do you wish to automatically update boot archives?` と表示されたら、**no** を入力します。

ルートプロンプト (#) が表示されます。

注 – Solaris フェイルセーフブートモードを終了するときは、システムをリブートする必要があります。システムをリブートしてマルチユーザーモードに戻すには、システムディスクが正常に追加され、データが復元されて、ブートブロックのインストールが完了している必要があります。

参考 システムディスクを接続したあとの作業

ディスクのサイズが、1T バイトより少ない場合は、`fdisk` パーティションを作成できます。258 ページの「x86: Solaris `fdisk` パーティションを作成する方法」を参照してください。

`fdisk` パーティション識別子の変更

x86 システム版 Solaris の `fdisk` パーティション識別子が、130 (`0x82`) から 191 (`0xbf`) に変更されました。Solaris のすべてのコマンド、ユーティリティー、およびドライバは、どちらの `fdisk` 識別子を使用しても機能するように更新されました。`fdisk` の機能に変更はありません。

▼ Solaris `fdisk` 識別子を変更する方法

`fdisk` の新しいメニュー項目を使用すると、新しい識別子と古い識別子を切り替えられます。`fdisk` の識別子は、パーティションに含まれるファイルシステムがマウントされている場合でも変更できます。

`fdisk` メニューでは、次の 2 つの `type` の値によって、古い識別子と新しい識別子が示されます。

- `Solaris` は `0x82` を示す
- `Solaris2` は `0xbf` を示す

1 スーパーユーザーになります。

2 現在の `fdisk` 識別子を表示します。

次に例を示します。

```
            Total disk size is 39890 cylinders
            Cylinder size is 4032 (512 byte) blocks

                                                 Cylinders
```

```
     Partition   Status    Type          Start   End   Length    %
     =========   ======    ============  =====   ===   ======   ===
         1       Active    x86 Boot          1     6       6      0
         2                 Solaris2          7  39889    39883    100
```

3 fdisk メニューのオプション **4** を選択し、fdisk パーティション識別子を 0x82 に変更します。

```
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 4
```

4 オプション **5** を選択し、ディスク構成を更新して終了します。

5 必要であれば、fdisk メニューのオプション **4** を選択し、fdisk パーティション識別子を 0xbf に変更します。

次に例を示します。

```
Total disk size is 39890 cylinders
            Cylinder size is 4032 (512 byte) blocks

                                             Cylinders
     Partition   Status    Type          Start   End   Length    %
     =========   ======    ============  =====   ===   ======   ===
         1       Active    x86 Boot          1     6       6      0
         2                 Solaris           7  39889    39883    100

SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 4
```

6 オプション **5** を選択し、ディスク構成を更新して終了します。

▼ x86: 二次ディスクを接続してブートする方法

x64 システム上で EFI ディスクラベル付きのディスクを追加する場合、その詳細は、198 ページの「EFI ディスクラベル」を参照してください。

一般に、最近のバスタイプのほとんどで、ホットプラグ機能がサポートされています。システムのバスタイプでホットプラグ機能がサポートされている場合は、次の手順 2 または手順 3 を行う必要がない場合があります。

ホットプラグ対応デバイスの詳細は、第 6 章「デバイスの動的構成 (手順)」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 (省略可能) 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 (省略可能) 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、またはあと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検査します。

4 システムをシャットダウンします。

```
# shutdown -i0 -gn -y
```

`-i0` システムを実行レベル 0 (電源切断) にします。

`-g`*n* ログインしているユーザーに、*n* 秒後にシステムのシャットダウンを開始することを通知します。

`-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します。

`Press any key to reboot` 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、システム上のほか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、ディスクの裏側には、このための小さいスイッチがあります。

7 ディスクをシステムに接続して、その物理的接続を確認します。
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が表示されます。

参考 二次ディスクを接続し、ブートしたあとの作業

ディスクのサイズが 1T バイトより小さい場合は、システムのブート後に fdisk パーティションを作成できます。258 ページの「x86: Solaris fdisk パーティションを作成する方法」を参照してください。

x86: fdisk パーティションの作成上のガイドライン

次のガイドラインに従って 1 つまたは複数の fdisk パーティションを設定してください。

- fdisk コマンドは、1T バイトを超えるサイズの EFI ラベル付きディスクで使用することはできません。
- 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 パーティションにする必要があります。
- 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する必要があります。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のことです。
- Solaris の fdisk パーティションは、シリンダ境界から開始する必要があります。
- 最初のディスクの先頭のセクターには、ブート情報 (マスターブートレコードを含む) が書き込まれるので、最初のディスクの 1 番目の fdisk パーティションとして、Solaris の fdisk パーティションを作成する場合は、ディスクのシリンダ 0 ではなくシリンダ 1 から開始しなければなりません。
- ディスク全体を Solaris fdisk パーティションにすることもできます。または、もっと小さいサイズにして、DOS パーティションに多くの容量を割り当てることもできます。十分な空き容量がある場合は、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することもできます。

---

x86 のみ – Solaris スライスはパーティションと呼ばれることがあります。インタフェースによっては、「パーティション」の代わりに「スライス」という用語を使用します。

fdisk パーティションは、x86 システムでのみサポートされます。混乱を避けるため、Solaris のマニュアルでは、fdisk パーティションと Solaris fdisk パーティション内のエンティティーを区別しています。「スライス」または「パーティション」と呼ばれるのは、後者です。

▼ x86: Solaris fdisk パーティションを作成する方法

始める前に　fdisk パーティションについては、257 ページの「x86: fdisk パーティションの作成上のガイドライン」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. format ユーティリティーを起動します。

   ```
   # format
   ```

   番号付きのディスクのリストが表示されます。

   詳細は、format(1M) のマニュアルページを参照してください。

3. Solaris fdisk パーティションを作成するディスクの番号を入力します。

   ```
   Specify disk (enter its number): disk-number
   ```

   disk-number は、Solaris fdisk パーティションを作成するディスクの番号です。

4. fdisk メニューを選択します。

   ```
   format> fdisk
   ```

   表示される fdisk メニューは、fdisk パーティションがすでにディスク上に存在しているかどうかによって異なります。次の表を使用して、次に行う手順を決定してください。

作業	次の手順	参照先
ディスク全体を占有する Solaris fdisk パーティションを作成します。	手順 5	例 13–1
Solaris fdisk パーティションを作成します。既存の Solaris 以外の fdisk パーティションは変更しません。	手順 6	例 13–2
Solaris fdisk パーティションと、Solaris 以外の fdisk パーティションを追加作成します。	手順 6	例 13–3

5. ディスク全体にまたがる Solaris fdisk パーティションを作成してそれをアクティブにするには、プロンプトで y を入力します。次に、手順 13 に進みます。

   ```
   No fdisk table exists. The default partition for the disk is:

   a 100% "SOLARIS System" partition

   Type "y" to accept the default partition, otherwise type "n" to edit the
   partition table.
   y
   ```

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ない場合は、プロンプトで n を入力します。

```
Type "y" to accept the default partition, otherwise type "n" to edit the
 partition table.
n
             Total disk size is 3498 cylinders
             Cylinder size is 1199 (512 byte) blocks
                                          Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
SELECT ONE OF THE FOLLOWING:

  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection:
```

7 「**1.** Create a partition」を選択し、fdisk パーティションを作成します。

```
Enter Selection: 1
```

8 「1(=Solaris2)」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=SOLARIS2   2=UNIX        3=PCIXOS     4=Other
  5=DOS12      6=DOS16       7=DOSEXT     8=DOSBIG
  9=DOS16LBA   A=x86 Boot    B=Diagnostic C=FAT32
  D=FAT32LBA   E=DOSEXTLBA   F=EFI        0=Exit? 1
```

9 **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。

```
Specify the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders). nn
```

10 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。

```
Should this to become the active partition? If yes, it will be
activated each time the computer is reset or turned on.
Please type "y" or "n". y
```

fdisk パーティションがアクティブになったあと、Enter Selection: プロンプトが表示されます。

11 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 パーティションの作成方法については、手順 8 - 10 を参照してください。

12 ディスク構成を更新し、Selection メニューから fdisk メニューを終了します。

```
Selection: 5
```

13 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。

```
format> label
Ready to label disk, continue? yes
format>
```

14 format ユーティリティーを終了します。

```
format> quit
```

例 13–1 x86: ディスク全体を占有する Solaris fdisk パーティションを作成する

次の例は、format ユーティリティーの fdisk オプションを使用して、ディスク全体にまたがる Solaris fdisk パーティションを作成する方法を説明しています。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c0d0 <DEFAULT cyl 2466 alt 2 hd 16 sec 63>
          /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0
       1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
          /pci@0,0/pci-ide@7,1/ide@0/cmdk@1,0
       2. c1d0 <DEFAULT cyl 13102 alt 2 hd 16 sec 63>
          /pci@0,0/pci-ide@7,1/ide@1/cmdk@0,0
Specify disk (enter its number): 0
selecting c0d0
Controller working list found
[disk formatted]
format> fdisk
No fdisk table exists. The default partitioning for your disk is:

  a 100% "SOLARIS System" partition.

Type "y" to accept the default partition, otherwise type "n" to edit the
partition table. y

format> label
Ready to label disk, continue? yes
format> quit
```

例 13–2 x86: 既存の fdisk パーティションを変更せずに、Solaris fdisk パーティションを作成する

次の例は、DOS-BIG fdisk パーティションがすでに存在しているディスクに、Solaris fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
            Total disk size is 3498 cylinders
            Cylinder size is 1199 (512 byte) blocks

                                             Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1       Active    DOS-BIG           1   699     699     20
```

```
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
  1=SOLARIS2  2=UNIX        3=PCIXOS     4=Other
  5=DOS12     6=DOS16       7=DOSEXT     8=DOSBIG
  9=DOS16LBA  A=x86 Boot    B=Diagnostic C=FAT32
  D=FAT32LBA  E=DOSEXTLBA   F=EFI        0=Exit?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Should this become the active partition? If yes, it will be
activated each time the computer is or turned on.
Please type "y" or "n". y
             Total disk size is 3498 cylinders
             Cylinder size is 1199 (512 byte) blocks
                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 DOS-BIG           1   699     699     20
          2       Active    Solaris2        700  3497    2798     80

SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection:5
Partition 2 is now the active partition
format> label
Ready to label disk, continue? yes
format> q
```

例 13–3 x86: Solaris fdisk パーティションと、DOSBIG fdisk パーティションを作成する

次の例は、Solaris fdisk パーティションと DOSBIG fdisk パーティションを作成する方法を説明しています。

```
format> fdisk
No fdisk table exists. The default partitioning for your disk is:
   a 100% "SOLARIS System" partition.
Type "y" to accept the default partition, otherwise type "n" to edit the
partition table.
n
             Total disk size is 3498 cylinders
             Cylinder size is 1199 (512 byte) blocks
                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
```

```
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
1=SOLARIS2   2=UNIX        3=PCIXOS     4=Other
5=DOS12      6=DOS16       7=DOSEXT     8=DOSBIG
9=DOS16LBA   A=x86 Boot    B=Diagnostic C=FAT32
D=FAT32LBA   E=DOSEXTLBA   F=EFI        0=Exit?  8
Specify the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders)20
Should this to become the Active partition? If yes, it will be
activated each time the computer is reset or turned on.
again. Please type "y" or "n". n
             Total disk size is 3498 cylinders
             Cylinder size is 1199 (512 byte) blocks
                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 DOS-BIG           1   699     699     20
  SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 1
Indicate the type of partition you want to create
1=SOLARIS2   2=UNIX        3=PCIXOS     4=Other
5=DOS12      6=DOS16       7=DOSEXT     8=DOSBIG
9=DOS16LBA   A=x86 Boot    B=Diagnostic C=FAT32
D=FAT32LBA   E=DOSEXTLBA   F=EFI        0=Exit? 1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Should this become the active partition? If yes, it will be
activated each time the computer is reset or turned on.
Please type "y" or "n". y
             Total disk size is 3498 cylinders
             Cylinder size is 1199 (512 byte) blocks
                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 DOS-BIG           1   699     699     20
          2       Active    Solaris2        700  3497    2798     80
SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 5
Partition 2 is now the Active partition
format> q
```

参考 Solaris fdisk パーティションを作成したあとの作業

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。263 ページの「x86: ディスクスライスとディスクラベルを作成する方法」を参照してください。

▼ x86: ディスクスライスとディスクラベルを作成する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. format ユーティリティーを起動します。

   ```
   # format
   ```

   番号付きのディスクのリストが表示されます。
3. パーティションの再分割を実行するディスクの番号を入力します。

   ```
   Specify disk (enter its number): disk-number
   ```

   disk-number は、パーティションの再分割を実行するディスクの番号です。
4. partition メニューを選択します。

   ```
   format> partition
   ```
5. 現在のパーティション (スライス) テーブルを表示します。

   ```
   partition> print
   ```
6. 変更作業を開始します。

   ```
   partition> modify
   ```
7. ディスクをすべて **free hog** に設定します。

   ```
   Choose base (enter number) [0]? 1
   ```

   free hog スライスの詳細は、213 ページの「free hog スライスの使用方法」を参照してください。
8. 続行するかどうかを尋ねるプロンプトが表示されたら、yes と応答して新しいパーティションテーブルを作成します。

   ```
   Do you wish to continue creating a new partition
   table based on above table[yes]? yes
   ```
9. プロンプトが表示されたら、**free hog** パーティション (スライス) と各スライスのサイズを指定します。

   システムディスクを追加するときは、次のスライスを設定しなければなりません。

- ルート (スライス 0)、スワップ (スライス 1) (必須) および
- `/usr` (スライス 6)

スライスの情報を設定すると、新しいパーティションテーブルが表示されます。

10 プロンプトが表示されたら yes と応答して、表示されたパーティションテーブルを現在のパーティションテーブルにします。

```
Okay to make this the current partition table[yes]? yes
```

表示されたパーティションテーブルが希望どおりでないために変更する場合は、no と応答して手順 6 に戻ります。

11 パーティションテーブルに名前を付けます。

```
Enter table name (remember quotes): "partition-name"
```

partition-name には、新しいパーティションテーブルの名前を指定します。

12 新しいディスク上でスライスの割り当てが終わったら、新しいパーティションテーブルを使用してディスクにラベルを付けます。

```
Ready to label disk, continue? yes
```

13 `partition` メニューを終了します。

```
partition> quit
```

14 新しいディスクラベルを検査します。

```
format> verify
```

15 `format` ユーティリティーを終了します。

```
format> quit
```

参考 ディスクスライスとディスクラベルを作成したあとの作業

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。264 ページの「x86: ファイルシステムを作成する方法」を参照してください。

▼ x86: ファイルシステムを作成する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 スライスごとにファイルシステムを作成します。

```
# newfs /dev/rdsk/cwtxdysz
```

`/dev/rdsk/c`*w*`t`*x*`d`*y*`s`*z* は、作成するファイルシステム用の raw デバイスです。

newfs コマンドの詳細は、第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」または newfs(1M) のマニュアルページを参照してください。

3 新しいファイルシステムをマウントして、確認します。

```
# mount /dev/dsk/cwtxdysz /mnt
# ls /mnt
lost+found
```

参考 ファイルシステムを作成したあとの作業

- システムディスク – ディスク上にルート (/) と /usr のファイルシステムを復元する必要があります。
  - 第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。
  - ルート (/) と /usr ファイルシステムの復元後、ブートブロックをインストールします。265 ページの「x86: システムディスクにブートブロックをインストールする方法」を参照してください。
- 二次ディスク – 新しいディスク上にファイルシステムを復元しなければならないことがあります。第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。新しいディスク上にファイルシステムを復元しない場合は、二次ディスクを追加する作業が終わったことになります。
- ユーザーがファイルシステムを利用できるようにする方法については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

▼ x86: システムディスクにブートブロックをインストールする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 システムディスクにブートブロックをインストールします。

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/cwtxdysz
```

/boot/grub/stage1 これは、パーティションのブートファイルです。

/boot/grub/stage2 これは、ブートブロックコードです。

/dev/rdsk/cwtxdysz これは、Solaris ルートスライス上の GRUB メニューの場所 /boot/grub/menu.lst を表す raw デバイス名です。

詳細は、installgrub(1M) のマニュアルページを参照してください。

3 システムをリブートし、レベル 3 で実行することによって、ブートブロックがインストールされていることを確認します。

```
# init 6
```

例 13–4 x86: システムディスクにブートブロックをインストールする

次の例では、ブートブロックを x86 システムにインストールする方法を示します。

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0
stage1 written to partition 0 sector 0 (abs 2016)
stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)
```

◆ ◆ ◆ 第 14 章

Oracle Solaris iSCSI ターゲットおよびイニシエータの構成 (手順)

この章では、Oracle Solaris リリースで使用可能な Solaris iSCSI ターゲットおよびイニシエータの構成方法について説明します。iSCSI ターゲットおよびイニシエータの構成に関連する手順については、269 ページの「Solaris iSCSI ターゲットおよびイニシエータの設定 (作業マップ)」を参照してください。

最新の Solaris 10 リリースにおける Solaris iSCSI イニシエータの新機能については、191 ページの「ディスク管理の新機能」を参照してください。

OpenSolaris COMSTAR (Common Multiprotocol SCSI TARget) は、任意の OpenSolaris ホストを SCSI ターゲットデバイスに変換可能なソフトウェアフレームワークです。変換した SCSI ターゲットデバイスには、イニシエータホストからストレージネットワーク経由でアクセス可能になります。OpenSolaris COMSTAR の使用方法については、次のサイトを参照してください。

```
http://wikis.sun.com/display/OpenSolarisInfo/COMSTAR+Administration
```

このリリースで使用可能な iSCSI ブート機能については、次のサイトを参照してください。

```
http://wikis.sun.com/
display/OpenSolarisInfo/iSCSI+Boot+for+OpenSolaris+User%27s+Guide
```

Solaris iSCSI の構成の問題のトラブルシューティングについては、294 ページの「iSCSI 構成に関する問題の障害追跡」を参照してください。

iSCSI 技術 (概要)

iSCSI は Internet SCSI (Small Computer System Interface) の略語であり、データストレージサブシステムを結合するための、インターネットプロトコル (IP) ベースのストレージネットワーキング標準です。このネットワーキング標準は、IETF (Internet Engineering Task Force) によって開発されました。iSCSI 技術の詳細は、次の RFC 3720 を参照してください。

http://www.ietf.org/rfc/rfc3720.txt

iSCSI プロトコルを使用すると、SCSI コマンドが IP ネットワーク経由で転送されるため、ユーザーはあたかもブロックデバイスがローカルシステムに接続されているかのようにネットワーク経由でブロックデバイスにアクセスできます。

既存の TCP/IP ネットワーク内のストレージデバイスを使用する場合、次の解決法が利用できます。

- iSCSI ブロックデバイスまたはテープ – SCSI コマンドとデータをブロックレベルから IP パケットへと変換します。あるシステムと、テープデバイスやデータベースなどのターゲットデバイスとの間で、ブロックレベルのアクセスが必要になる場合には、ネットワーク内で iSCSI を使用することをお勧めします。ブロックレベルデバイスへのアクセスにはロックがかからないため、iSCSI ターゲットデバイスなどのブロックレベルデバイスに複数のユーザーやシステムがアクセスすることができます。
- NFS – ファイルデータを IP 経由で転送します。ネットワーク内で NFS を使用する利点は、ファイルデータを複数のシステム間で共有できることにあります。NFS 環境で利用可能なデータに多数のユーザーがアクセスする場合、必要に応じてファイルデータへのアクセスにロックがかかります。

Solaris iSCSI ターゲットおよびイニシエータを使用する利点を次に示します。

- iSCSI プロトコルは、既存の Ethernet ネットワーク上で動作します。
  - サポートされている任意のネットワークインタフェースカード (NIC)、Ethernet ハブ、または Ethernet スイッチを使用できます。
  - 1 つの IP ポートから複数の iSCSI ターゲットデバイスを処理できます。
  - IP ネットワークの既存のインフラストラクチャーや管理ツールを使用できます。
- クライアントに接続可能な既存のファイバチャネルデバイスを利用でき、ファイバチャネル HBA の費用がかかりません。また、専用のアレイを持つシステムが、複製されたストレージを ZFS または UFS ファイルシステムでエクスポートすることも可能になりました。
- 構成可能な iSCSI ターゲットデバイスの最大数に制限はありません。
- このプロトコルは、適切なハードウェアを備えたファイバチャネル SAN (Storage Area Network) 環境または iSCSI SAN 環境への接続に使用できます。

Solaris iSCSI イニシエータソフトウェア使用時の現時点における制限や制約を、次に示します。

- SLP を使用する iSCSI デバイスは、現時点ではサポートされていません。
- iSCSI デバイスのブートは、現時点ではサポートされていません。
- iSCSI ターゲットをダンプデバイスとして構成することはできません。
- iSCSI は 1 つのセッションで複数の接続をサポートしますが、現在の Solaris 実装は 1 つのセッションで 1 つの接続しかサポートしません。

  詳細は、RFC 3720 を参照してください。
- 既存のネットワーク経由で大量のデータを転送すると、パフォーマンスに影響する可能性があります。

Solaris iSCSI のソフトウェア要件およびハードウェア要件

- Solaris iSCSI ソフトウェアおよびデバイス
- Solaris iSCSI イニシエータソフトウェアの場合は Solaris 10 リリース (1/06 リリース以降)
- Solaris iSCSI ターゲットソフトウェアの場合は Solaris 10 リリース (8/07 リリース以降)
- 次の Solaris 10 ソフトウェアパッケージ:
  - `SUNWiscsir` – Sun iSCSI デバイスドライバ (`root`)
  - `SUNWiscsiu` – Sun iSCSI 管理ユーティリティー (`usr`)
  - `SUNWiscsitgtr` – Sun iSCSI ターゲットデバイスドライバ (`root`)
  - `SUNWiscsitgtu` – Sun iSCSI ターゲット管理ユーティリティー (`usr`)
- サポートされている任意の NIC

Solaris iSCSI ターゲットおよびイニシエータの設定 (作業マップ)

作業	説明	参照先
1. iSCSI のソフトウェア要件およびハードウェア要件を確認します。	iSCSI ベースのストレージネットワークを設定するためのソフトウェア要件およびハードウェア要件を確認します。	269 ページの「Solaris iSCSI のソフトウェア要件およびハードウェア要件」

作業	説明	参照先
2. 使用する iSCSI ターゲットデバイスを設定します。	Solaris iSCSI ターゲットデバイスを接続および設定します。 サードパーティーのターゲットデバイスを設定することもできます。設定手順については、ベンダーのマニュアルを参照してください。 iSNS サーバーが使用可能な場合は、Solaris iSCSI ターゲットの iSNS 発見を設定します。	273 ページの「Solaris iSCSI ターゲットデバイスの設定」 275 ページの「Solaris iSCSI ターゲットの iSNS 発見を設定する方法」
3. Solaris iSCSI の構成を準備します。	正しいバージョンのソフトウェアとハードウェアがインストールされていることを確認します。	273 ページの「Solaris iSCSI 構成の準備を整える方法」
4. (省略可能) Solaris iSCSI 構成の認証を設定します。	使用する Solaris iSCSI 構成内で認証を使用するかどうかを判断します。 単方向 CHAP または双方向 CHAP の使用を検討します。 他社製の RADIUS サーバーを使用した CHAP 管理の単純化を検討します。	276 ページの「iSCSI イニシエータの CHAP 認証を構成する方法」 277 ページの「iSCSI ターゲットの CHAP 認証を構成する方法」 278 ページの「iSCSI 構成の RADIUS を構成する方法」
5. iSCSI ターゲット発見を構成します。	環境に最適な iSCSI ターゲット発見方式を選択します。	279 ページの「iSCSI ターゲット発見を構成する方法」
6. (省略可能) 発見された iSCSI ターゲットを削除します。	発見された iSCSI ターゲットを削除しなければならない場合があります。	280 ページの「発見された iSCSI ターゲットを削除する方法」
7. iSCSI ディスクへのアクセス	iSCSI ディスクへのアクセスには `format` ユーティリティーを使用できます。システムのリブート時に iSCSI ディスクを自動的に使用可能にすることもできます。	282 ページの「iSCSI ディスクへのアクセス」
8. iSCSI 構成を監視します。	`iscsiadm` コマンドを使って iSCSI 構成を監視します。	284 ページの「iSCSI 構成の監視」

作業	説明	参照先
9. (省略可能) iSCSI 構成を変更します。	ヘッダーダイジェストパラメータやデータダイジェストパラメータなど、iSCSI ターゲットの設定を変更する場合があります。	287 ページの「iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法」
10. (省略可能) Solaris iSCSI マルチパスデバイスを設定します。	Solaris iSCSI マルチパスデバイスを設定するかどうかを決定します。 この手順を使用して、単一のターゲットに接続する複数の iSCSI セッションを作成します。	290 ページの「Solaris iSCSI マルチパスデバイスの設定」 291 ページの「ターゲットの複数の iSCSI セッションを有効にする方法」

Solaris iSCSI ターゲットおよびイニシエータの構成

Solaris iSCSI ターゲットおよびイニシエータの構成手順は、次のとおりです。

- ハードウェア要件およびソフトウェア要件の確認
- IP ネットワークの構成
- iSCSI ターゲットデバイスの接続および設定
- (省略可能) 必要に応じて、iSCSI イニシエータと iSCSI ターゲット間における iSCSI 認証の構成
- iSCSI ターゲット発見方式の構成
- iSCSI ディスク上でのファイルシステムの作成
- iSCSI 構成の監視

iSCSI 構成の情報は、`/etc/iscsi` ディレクトリ内に保存されます。この情報を管理する必要はありません。

iSCSI 関連の用語

iSCSI ターゲットおよびイニシエータを構成する前に、次の用語を確認してください。

用語	説明
イニシエータ	iSCSI ターゲットに対する SCSI 要求を発行するドライバ。
ターゲットデバイス	iSCSI ストレージコンポーネント。
発見	利用可能なターゲットの一覧をイニシエータに提供するプロセス。

用語	説明
発見方式	iSCSI ターゲットを発見するための方法。現時点では次の 3 つの方法を使用できます。 ■ Internet Storage Name Service (iSNS) – 1 つ以上の iSNS サーバーと対話することで可能性のあるターゲットを発見します。 ■ SendTargets – *discovery-address* を使用することによって、可能性のあるターゲットが発見されます。 ■ 静的 – 静的なターゲットアドレスが構成されます。

動的または静的ターゲット発見の構成

デバイス発見を実行するために、動的デバイス発見方式のいずれかを構成するか、または静的 iSCSI イニシエータターゲットを使用するかを決定します。

- 動的デバイス発見 – ファイバチャネルブリッジへの iSCSI などのように、iSCSI ノードが多数のターゲットを公開している場合、その iSCSI ノードに IP アドレスとポートの組み合わせを提供すると、その iSCSI イニシエータは SendTargets 機能を使ってデバイス発見を実行できるようになります。

  次の 2 つの動的デバイス発見方式を使用できます。

  - SendTargets - ファイバチャネルブリッジへの iSCSI などのように、iSCSI ノードが多数のターゲットを公開している場合、その iSCSI ノードに IP アドレスとポートの組み合わせを提供すると、その iSCSI イニシエータは SendTargets 機能を使ってデバイス発見を実行できるようになります。
  - iSNS - iSNS (Internet Storage Name Service) を使用すると、できる限り少ない構成情報で、iSCSI イニシエータがアクセス権を持つターゲットを発見できます。また、ストレージノードの動作状態が変更されたときに iSCSI イニシエータに通知する状態変更通知機能もあります。iSNS 発見方式を使用するために、iSNS サーバーのアドレスとポートの組み合わせを指定して、デバイス発見を実行するために指定した iSNS サーバーを iSCSI イニシエータで照会できるようにすることができます。iSNS サーバーのデフォルトポートは 3205 です。 iSNS の詳細については、RFC 4171 を参照してください。

    `http://www.ietf.org/rfc/rfc4171.txt`

    iSNS 発見サービスは、ネットワーク内のすべてのターゲットを発見するための管理モデルを提供します。

- 静的デバイス発見 – iSCSI ノードのターゲットが少数である場合や、イニシエータのアクセス対象となるターゲットを制限する場合には、次の静的ターゲットアドレス命名規則を使用して、*target-name* を静的に構成できます。

  target,*target-address*[:*port-number*]

  また、アレイの管理ツールから静的ターゲットアドレスを決定できます。

注 – 単一の iSCSI ターゲットが静的デバイス発見方式と動的デバイス発見方式の両方によって発見されるように構成しないでください。発見方式を重複して使用すると、イニシエータが iSCSI ターゲットデバイスと通信するときのパフォーマンスが低下する可能性があります。

▼ Solaris iSCSI 構成の準備を整える方法

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスにアクセスするものとします。

1 スーパーユーザーになります。

2 **iSCSI** ソフトウェアパッケージがインストールされていることを確認します。

```
initiator# pkginfo SUNWiscsiu SUNWiscsir
system      SUNWiscsiu Sun iSCSI Device Driver (root)
system      SUNWiscsir Sun iSCSI Management Utilities (usr)
```

3 **Solaris 10 1/06** 以降のリリースが動作していることを確認します。

4 **TCP/IP** ネットワークが設定済みであることを確認します。

5 **iSCSI** ターゲットデバイスを接続し、それらが構成済みであることを確認します。

たとえば、iSCSI ターゲットデバイスが到達可能かどうかを確認するには、`telnet` コマンドを使ってポート 3260 経由で iSCSI ターゲットデバイスに接続します。接続が拒否された場合は、294 ページの「iSCSI 構成に関する問題の障害追跡」を参照してください。

他社製の iSCSI ターゲットデバイスの接続方法については、そのベンダーのドキュメントを参照してください。

Solaris iSCSI ターゲットデバイスの設定

`iscsitadm` コマンドを使って、Solaris iSCSI ターゲットデバイスを設定および管理できます。Solaris iSCSI ターゲットデバイスは、ディスクまたはテープデバイスです。iSCSI ターゲットとして選択したデバイスでは、iSCSI デーモン用のバッキングストアとして、同サイズの ZFS または UFS ファイルシステムを提供する必要があります。

ZFS を使用した Solaris iSCSI ターゲットデバイスの設定方法については、『Oracle Solaris ZFS 管理ガイド』の「ZFS および Solaris iSCSI の向上」を参照してください。

ターゲットデバイスの設定後に、`iscsiadm` コマンドを使って iSCSI ターゲットを識別します。これにより、iSCSI ターゲットデバイスが検出および使用されます。

詳細は、iscsitadm(1M) および iscsiadm(1M) のマニュアルページを参照してください。

基本的な操作を次に示します。

- バッキングストアディレクトリの識別 – iSCSI デーモンは、作成されたターゲットおよび論理ユニットごとに情報を格納する必要があります。デフォルトでは、このデバイス用のバッキングストアはベースディレクトリ内にも配置されます。このため、ホストシステムが使用する ZFS プールが大規模なものである場合、もっとも簡単な方法は、その場所にあるすべてをデーモンで格納できるように許可することです。バッキングストアを分散する必要がある場合は、各論理ユニットを作成するときにバッキングストアの場所を指定できます。
- iSCSI ターゲットの作成 – CLI はデフォルトで、要求されたデバイスの種類が論理ユニット 0 の LBA であるとみなします。文字型デバイスのためにパススルーモードが必要な場合には、-raw オプションを使用する必要があります。最初の LUN を作成したあとで、-lun number を指定して、同じ iSCSI ターゲットのほかの LUN を作成できます。

  デーモンによりバックグラウンドタスクが開始され、LUN がゼロに初期化されます。初期化中に基盤となるファイルシステムの使用率が 100 パーセントになった場合、デーモンによりターゲットが削除されます。この初期化中、LUN はオフラインとしてマーク付けされるため、イニシエータから使用することはできません。ただしこのとき、イニシエータからこの LUN を発見することはできます。待機中の Solaris イニシエータは、Inventory Change 通知を受信するとデバイスを自動的にオンラインにします。

▼ iSCSI ターゲットの作成方法

この手順では、iSCSI ターゲットのあるローカルシステムにユーザーがログインしているものとします。

1. スーパーユーザーになります。
2. バッキングストアディレクトリを識別します。

   次に例を示します。

   ```
   target# iscsitadm modify admin -d /export/sandbox
   ```
3. iSCSI ターゲットを作成します。

   次に例を示します。

   ```
   target# iscsitadm create target --size 2g sandbox
   ```
4. iSCSI ターゲットに関する情報を表示します。

   次に例を示します。

   ```
   target# iscsitadm list target -v sandbox
   ```

5 このターゲットを検出および使用するように、**iSCSI** イニシエータを設定します。
詳細は、279 ページの「iSCSI ターゲット発見を構成する方法」を参照してください。

▼ Solaris iSCSI ターゲットの iSNS 発見を設定する方法

ネットワークに他社製の iSNS サーバーまたは Sun iSNS サーバーが含まれている場合は、Solaris iSCSI ターゲットの iSNS 発見を設定できます。

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスにアクセスするものとします。

1 スーパーユーザーになります。

2 **iSNS** サーバーの情報を追加します。
次に例を示します。

```
initiator# iscsitadm modify admin --isns-server ip-address or hostname[:port]
```

ネットワーク内の iSNS サーバーの *ip-address* を特定します。

この手順により、iSNS サーバーの情報がすべての Solaris iSCSI ターゲットに追加されます。

3 **iSNS** サーバー発見を有効にします。
次に例を示します。

```
initiator# iscsitadm modify admin --isns-access enable
```

この手順により、すべての Solaris iSCSI ターゲットで iSNS 発見が有効になります。

iSCSI ベースのストレージネットワークにおける認証の構成

iSCSI デバイスの認証設定は省略可能です。

セキュリティー保護された環境では、信頼できるイニシエータだけがターゲットにアクセスできるため、認証は必要ありません。

セキュリティー保護の不十分な環境では、ターゲットは、接続要求が本当に指定されたホストからのものなのかを判断できません。そのような場合、ターゲットは、チャレンジハンドシェーク認証プロトコル (CHAP) を使ってイニシエータを認証できます。

CHAP 認証では「チャレンジ」と「応答」の概念が使用され、つまり、ターゲットがイニシエータに対して身元の証明を要求します。このチャレンジ / 応答方式が機能するには、ターゲットがイニシエータの秘密鍵を知っており、かつイニシエータが

チャレンジに応答するように設定されている必要があります。秘密鍵をアレイ上に設定する手順については、アレイのベンダーのマニュアルを参照してください。

iSCSI は単方向認証と双方向認証をサポートします。

- 「単方向認証」では、ターゲットがイニシエータの身元を認証できます。
- 「双方向認証」では、二次レベルのセキュリティーを追加する目的で、イニシエータがターゲットの身元を認証することができます。

▼ iSCSI イニシエータの CHAP 認証を構成する方法

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスに安全にアクセスするものとします。

1 スーパーユーザーになります。

2 単方向 CHAP または双方向 CHAP のどちらを構成するかを決定します。

- 単方向認証 (デフォルトの方式) では、ターゲットがイニシエータを検証できます。手順 3 から 5 のみを実行してください。
- 双方向認証では、二次レベルのセキュリティーを追加する目的で、イニシエータがターゲットを認証することできます。手順 3 から 9 を実行してください。

3 単方向 CHAP – イニシエータ上で秘密鍵を設定します。

たとえば、次のコマンドを実行すると、CHAP の秘密鍵を定義するためのダイアログが起動されます。

```
initiator# iscsiadm modify initiator-node --CHAP-secret
```

注 – CHAP シークレットの長さは 12 文字 - 16 文字である必要があります。

---

4 (省略可能) 単方向 CHAP – イニシエータ上で CHAP 名を設定します。

デフォルトではイニシエータの CHAP 名は、イニシエータのノード名に設定されます。

次のコマンドを使用して、イニシエータの CHAP 名を変更できます。

```
initiator# iscsiadm modify initiator-node --CHAP-name new-CHAP-name
```

Solaris 環境では、CHAP 名はデフォルトで常にイニシエータノード名に設定されます。CHAP 名は、512 バイト未満の任意の長さのテキストに設定できます。512 バイトの長さの制限は Solaris の制限です。ただし、CHAP 名を設定しない場合は、初期化のときにイニシエータノード名に設定されます。

5 単方向 CHAP – シークレットの設定完了後にイニシエータ上で CHAP 認証を有効にします。

```
initiator# iscsiadm modify initiator-node --authentication CHAP
```

CHAP 認証では、イニシエータノードにユーザー名とパスワードが必要です。ユーザー名は通常、渡されたユーザー名のシークレットをターゲットが検索するために使用されます。

6 次のいずれかを選択して双方向 **CHAP** を有効または無効にします。

- 双方向 CHAP – ターゲットの双方向認証パラメータを有効にします。

  次に例を示します。

  ```
  initiator# iscsiadm modify target-param -B enable eui.5000ABCD78945E2B
  ```
- 双方向 CHAP を無効にします。次に例を示します。

  ```
  initiator# iscsiadm modify target-param -B disable eui.5000ABCD78945E2B
  ```

7 双方向 **CHAP** – ターゲット上で認証方法を **CHAP** に設定します。

次に例を示します。

```
initiator# iscsiadm modify target-param --authentication CHAP eui.5000ABCD78945E2B
```

8 双方向 **CHAP** – ターゲット上でターゲットデバイスの秘密鍵を設定します。

たとえば、次のコマンドを実行すると、CHAP の秘密鍵を定義するためのダイアログが起動されます。

```
initiator# iscsiadm modify target-param --CHAP-secret eui.5000ABCD78945E2B
```

9 双方向 **CHAP** - ターゲット上で **CHAP** 名を設定します。

デフォルトでは、ターゲットの CHAP 名はターゲット名に設定されます。

次のコマンドを使用して、ターゲットの CHAP 名を変更できます。

```
initiator# iscsiadm modify target-param --CHAP-name target-CHAP-name
```

▼ iSCSI ターゲットの CHAP 認証を構成する方法

この手順では、iSCSI ターゲットのあるローカルシステムにユーザーがログインしているものとします。

1 スーパーユーザーになります。

2 ターゲットの **CHAP** シークレット名を設定します。

規則では、ホスト名をシークレット名として使用します。次に例を示します。

```
target# iscsitadm modify admin -H stormpike
```

3 **CHAP** シークレットを指定します。

CHAP シークレットは、12 - 16 文字にする必要があります。次に例を示します。

```
target# iscsitadm modify admin -C
Enter secret: xxxxxx
Re-enter secret: xxxxxx
```

4 1 つ以上のターゲットに関連付けられるイニシエータオブジェクトを作成します。

この手順は、毎回 IQN 値を入力しなくても済むように、わかりやすい名前 (通常はホスト名、この場合は monster620) を IQN 値に関連付けるために行います。次に例を示します。

```
# iscsitadm create initiator -n iqn.1986-03.com.sun: 01:00e081553307.4399f40e monster620
```

5 イニシエータで使用したのと同じ CHAP 名を指定します。

この名前は、イニシエータオブジェクトに使用したわかりやすい名前でなくてもかまいません。次に例を示します。

```
target# iscsitadm modify initiator -H monster620 monster620
```

6 イニシエータで使用したのと同じ CHAP シークレットを指定します。

次に例を示します。

```
target# iscsitadm modify initiator -C monster620
Enter secret: xxxxxx
Re-enter secret: xxxxxx
```

7 イニシエータオブジェクトを 1 つ以上のターゲットに関連付けます。

次に例を示します。

```
target# iscsitadm modify target -l monster620 sandbox
```

他社製の RADIUS サーバーを使用して iSCSI 構成内の CHAP 管理を単純化する

他社製の RADIUS サーバーを使用すると、CHAP シークレット管理を単純化できます。RADIUS サーバーは集中管理認証サービスです。RADIUS サーバーを使って双方向認証を行う場合、イニシエータの CHAP シークレットは依然として指定する必要がありますが、各イニシエータ上で各ターゲットの CHAP シークレットを指定する必要はなくなります。

詳細は、次の項目を参照してください。

- CHAP – http://www.ietf.org/rfc/rfc1994.txt
- RADIUS – http://www.ietf.org/rfc/rfc2865.txt

▼ iSCSI 構成の RADIUS を構成する方法

この手順では、ユーザーが現在ログインしているローカルシステムから構成済みの iSCSI ターゲットデバイスに安全にアクセスするものとします。

1 スーパーユーザーになります。

2 **RADIUS サーバーの IP アドレスとポート (デフォルトのポートは 1812) を、イニシエータノードに設定します。**

次に例を示します。

```
initiator# iscsiadm modify initiator-node --radius-server 10.0.0.72:1812
```

3 **RADIUS サーバーの共有鍵をイニシエータノードに設定します。**

```
initiator# iscsiadm modify initiator-node --radius-shared-secret
```

注 – Solaris iSCSI 実装の場合、RADIUS サーバーに共有シークレットが構成されていないと、Solaris iSCSI ソフトウェアは RADIUS サーバーと通信できません。

---

4 **RADIUS サーバーを有効にします。**

```
initiator# iscsiadm modify initiator-node --radius-access enable
```

Solaris iSCSI と RADIUS サーバーに関するエラーメッセージ

この節では、Solaris iSCSI と RADIUS サーバーの構成に関係するエラーメッセージとその考えられる解決法について説明します。

`empty RADIUS shared secret`
: 原因: イニシエータ上で RADIUS サーバーが有効になっているにもかかわらず、RADIUS の共有シークレットが設定されていません。

  対処方法: RADIUS の共有シークレットをイニシエータに設定します。詳細は、278 ページの「iSCSI 構成の RADIUS を構成する方法」を参照してください。

`WARNING: RADIUS packet authentication failed`
: 原因: イニシエータによる RADIUS データパケットの認証が失敗しました。このエラーが発生する可能性があるのは、イニシエータノード上に設定された共有シークレットが RADIUS サーバー上の共有シークレットと異なっている場合です。

  正しい RADIUS 共有シークレットをイニシエータに設定し直します。詳細は、278 ページの「iSCSI 構成の RADIUS を構成する方法」を参照してください。

▼ iSCSI ターゲット発見を構成する方法

この手順では、ユーザーが現在ログインしているローカルシステムから iSCSI ターゲットデバイスへのアクセスを構成するものとします。

1 スーパーユーザーになります。

2 次のいずれかの方式を使って、ターゲットデバイスが動的または静的に発見されるように構成します。

- デバイスが動的に発見されるように構成します (SendTargets)。

  次に例を示します。

  ```
  initiator# iscsiadm add discovery-address 10.0.0.1:3260
  ```
- デバイスが動的に発見されるように構成します (iSNS)。

  次に例を示します。

  ```
  initiator# iscsiadm add iSNS-server 10.0.0.1:3205
  ```
- デバイスが静的に発見されるように構成します。

  次に例を示します。

  ```
  initiator# iscsiadm add static-config eui.5000ABCD78945E2B,10.0.0.1
  ```

発見方式が有効化されるまで iSCSI 接続は起動されません。次の手順を参照してください。

3 次のいずれかを使って **iSCSI** ターゲット発見方式を有効にします。

- 動的に発見される (SendTargets) デバイスを構成した場合は、 SendTargets 発見方式を有効にします。

  ```
  initiator# iscsiadm modify discovery --sendtargets enable
  ```
- 動的に発見される (iSNS) デバイスを構成した場合は、 iSNS 発見方式を有効にします。

  ```
  initiator# iscsiadm modify discovery --iSNS enable
  ```
- 静的ターゲットを構成した場合は、静的ターゲット発見方式を有効にします。

  ```
  initiator# iscsiadm modify discovery --static enable
  ```

4 ローカルシステム用の **iSCSI** デバイスリンクを作成します。

```
initiator# devfsadm -i iscsi
```

▼ 発見された **iSCSI** ターゲットを削除する方法

発見アドレス、iSNS サーバー、または静的構成を削除するか、あるいは発見方式を無効にしたあとで、関連付けられたターゲットがログアウトします。たとえば、これらの関連付けられたターゲットが引き続き使用され、ファイルシステムがマウントされている場合は、これらのデバイスのログアウトが失敗し、アクティブターゲットリスト上に残ります。

この省略可能な手順では、ユーザーが現在ログインしているローカルシステム上で、iSCSI ターゲットデバイスへのアクセスがすでに構成されているものとします。

1 スーパーユーザーになります。

2 **(省略可能)** 次のいずれかを使って **iSCSI** ターゲット発見方式を無効にします。

- SendTargets 発見方式を無効にする必要がある場合は、次のコマンドを使用します。

```
initiator# iscsiadm modify discovery --sendtargets disable
```

- iSNS 発見方式を無効にする必要がある場合は、次のコマンドを使用します。

```
initiator# iscsiadm modify discovery --iSNS disable
```

- 静的ターゲット発見方式を無効にする必要がある場合は、次のコマンドを使用します。

```
initiator# iscsiadm modify discovery --static disable
```

3 次のいずれかを使って **iSCSI** デバイス発見エントリを削除します。

- iSCSI SendTargets 発見エントリを削除します。

  次に例を示します。

```
initiator# iscsiadm remove discovery-address 10.0.0.1:3260
```

- iSCSI iSNS 発見エントリを削除します。

  次に例を示します。

```
# iscsiadm remove isns-server 10.0.0.1:3205
```

- 静的 iSCSI 発見エントリを削除します。

  次に例を示します。

```
initiator# iscsiadm remove static-config eui.5000ABCD78945E2B,10.0.0.1
```

注 – 使用中の論理ユニットが関連付けられた発見エントリを無効化または削除しようとすると、次のメッセージが表示され、無効化または削除が失敗します。

```
logical unit in use
```

このエラーが発生した場合は、論理ユニット上の関連付けられたすべての入出力を停止し、ファイルシステムのマウント解除などを行います。そのあとで無効化または削除の操作を再び実行します。

---

4 **iSCSI** ターゲットデバイスを削除します。

論理ユニット番号 (LUN) を指定して、ターゲットを削除します。ターゲットの作成時に LUN を指定しなかった場合、値 0 が使用されます。ターゲットに複数の LUN が関連付けられている場合は、LUN 0 を最後に削除する必要があります。

次に例を示します。

```
initiator# iscsitadm delete target --lun 0 sandbox
```

iSCSI ディスクへのアクセス

リブート時に iSCSI ディスクにアクセスするには、ディスク上にファイルシステムを作成し、SCSI デバイス上の UFS ファイルシステムの場合と同じ方法で、/etc/vfstab エントリを追加します。次に、iSCSI イニシエータサービスに合わせて、iSCSI ディスクをマウントする SMF サービスを新規作成します。詳細は、282 ページの「リブート時に iSCSI ディスクにアクセスする方法」を参照してください。

Solaris iSCSI イニシエータによってデバイスが発見されると、ログインネゴシエーションが自動的に行われます。Solaris iSCSI ドライバは、利用可能な LUN の個数を判断し、デバイスノードを作成します。この時点で、iSCSI デバイスをほかのすべての SCSI デバイスと同様に扱えます。

ローカルシステム上で iSCSI ディスクを表示するには、format ユーティリティーを使用します。

次の format 出力では、ディスク 2 および 3 が MPxIO の制御下にない iSCSI LUN です。ディスク 21 と 22 が MPxIO の制御下にある iSCSI LUN です。

```
initiator# format
AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e010685cf1,0
       1. c0t2d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e0106e3ba1,0
       2. c3t0d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>
          /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-477d70401-
           b0fff044352423a2-hostname-020000,0
       3. c3t1d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>
           /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-3fcd70401
           -085ff04434f423a2-hostname-010000,0
.
.
.
      21. c4t60A98000686F694B2F59775733426B77d0 <ABCSTORAGE-LUN-0.2 cyl
          4606 alt 2 hd 16 sec 256>
          /scsi_vhci/ssd@g60a98000686f694b2f59775733426b77
      22. c4t60A98000686F694B2F59775733434C41d0 <ABCSTORAGE-LUN-0.2 cyl
          4606 alt 2 hd 16 sec 256>
          /scsi_vhci/ssd@g60a98000686f694b2f59775733434c41
```

▼ リブート時に iSCSI ディスクにアクセスする方法

次に示す手順に従って、システムのリブート後に iSCSI ディスクにアクセスします。

1. **iSCSI LUN** のエントリを /etc/vfstab ファイルに追加します。mount at boot オプションを no に設定します。

```
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
/dev/dsk/c3t600144F04B555F370000093D00495B00d0s0        -       /mnt    ufs     -       no      -
```

2 システムのリブート後に **iSCSI LUN** を自動的にマウントする **SMF** サービスマニフェストを作成します。

たとえば、次の内容を含む XML マニフェストファイル `/var/svc/manifest/network/iscsi/iscsimount.xml` を作成します。

```
<?xml version="1.0"?>
 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

 <service_bundle type="manifest" name="iscsimount">
     <service
         name="iscsimount"
         type="service"
         version="1" >

         <single_instance/>

         <!--
         We depend on the iscsi initiator service
         -->
         <dependency
             name="client"
             type="service"
             grouping="require_all"
             restart_on="none">
             <service_fmri value="svc:/network/iscsi/initiator"/>
         </dependency>

         <instance name="default" enabled="false">

         <!-- mount iSCSI volumes -->
             <exec_method
                 type="method"
                 name="start"
                 exec="/var/tmp/iscsi_mount"
                 timeout_seconds="60"/>

         <!-- umount iSCSI volumes -->
             <exec_method
                 type="method"
                 name="stop"
                 exec="/var/tmp/iscsi_umount"
                 timeout_seconds="60" />

             <property_group name="startd" type="framework">
                 <propval name="duration" type="astring" value="transient"/>
             </property_group>
         </instance>

         <template>
             <common_name>
                 <loctext xml:lang="C">
                     iSCSI Mount Service
                 </loctext>
             </common_name>
         </template>
     </service>
 </service_bundle>
```

3 /var/tmp ディレクトリ内に次のファイルを作成します。

次の bash シェルスクリプトは、/mnt ディレクトリで iSCSI ディスクの mount および umount コマンドを実行します。

```
bash-4.0# cat /var/tmp/iscsi_mount
#!/usr/bin/bash
mount /mnt           #Add multiple iscsi lun(s) to mount
bash-4.0# cat /var/tmp/iscsi_umount
#!/usr/bin/bash
umount /mnt          #Add multiple iscsi lun(s) to umount
```

これらのスクリプトに実行権を設定します。次に例を示します。

```
bash-4.0# chmod u+x /var/tmp/iscsi_mount
bash-4.0# chmod u+x /var/tmp/iscsi_umount
```

4 マニフェストをインポートします。

```
# svccfg import /var/svc/manifest/network/iscsi/iscsimount.xml
# svccfg enable iscsimount
```

注 – このサービスを有効にしたあとで、システムのシャットダウン時またはリブート時に次のメッセージが表示されることがあります。

```
iscsi: NOTICE: iscsi session(4) - session logout failed (1)
iscsi: NOTICE: iscsi discovery failure - SendTargets method is not enabled
iscsi: NOTICE: iscsi discovery failure - iSNS method is not enabled
```

これらのメッセージは無視しても問題ありません。または、リブートの直前に次のコマンドを実行することで、メッセージの表示を回避できます。

```
# svcadm disable -t iscsimount
```

Solaris iSCSI イニシエータによってデバイスが発見されると、ログインネゴシエーションが自動的に行われます。Solaris iSCSI ドライバは、利用可能な LUN の個数を判断し、デバイスノードを作成します。この時点で、iSCSI デバイスをほかのすべての SCSI デバイスと同様に扱えます。

▼ iSCSI 構成の監視

iscsiadm list コマンドを使用して、iSCSI イニシエータとターゲットデバイスに関する情報を表示できます。

1 スーパーユーザーになります。

2 **iSCSI イニシエータに関する情報を表示します。**

次に例を示します。

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
        Login Parameters (Default/Configured):
                Header Digest: NONE/-
                Data Digest: NONE/-
        Authentication Type: NONE
        RADIUS Server: NONE
        RADIUS access: unknown
        Configured Sessions: 1
```

3 使用中の発見方式に関する情報を表示します。

次に例を示します。

```
# iscsiadm list discovery
Discovery:
    Static: enabled
    Send Targets: enabled
    iSNS: enabled
```

例 14–1 iSCSI ターゲットの情報の表示

次の例では、特定の iSCSI ターゲットのパラメータの設定を表示する方法を示します。

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
        Target: iqn.1992-08.com.abcstorage:sn.33592219
```

`iscsiadm list target-param -v` コマンドによって次の情報が表示されます。

- ターゲットの認証設定
- ターゲットのログインパラメータのデフォルト設定
- 各ログインパラメータに構成されている値

`iscsiadm list target-param -v` コマンドでは、`/` 指定子の前にデフォルトのパラメータ値が表示され、`/` 指定子の後に構成後のパラメータ値が表示されます。パラメータをまったく構成していない場合は、構成後のパラメータの値がハイフン (-) で表示されます。詳細は、次の例を参照してください。

```
# iscsiadm list target-param -v eui.50060e8004275511 Target: eui.50060e8004275511
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
                Error Recovery Level: 0/-
                First Burst Length: 65536/-
```

```
                Immediate Data: yes/-
                Initial Ready To Transfer (R2T): yes/-
                Max Burst Length: 262144/-
                Max Outstanding R2T: 1/-
                Max Receive Data Segment Length: 65536/-
                Max Connections: 1/-
                Header Digest: NONE/-
                Data Digest: NONE/-
        Configured Sessions: 1
```

次の例では、ターゲットとイニシエータの間でネゴシエートされたあとのパラメータが出力されています。

```
# iscsiadm list target -v eui.50060e8004275511
Target: eui.50060e8004275511
        TPGT: 1
        ISID: 4000002a0000
        Connections: 1
                CID: 0
                  IP address (Local): 172.90.101.71:32813
                  IP address (Peer): 172.90.101.40:3260
                  Discovery Method: Static
                  Login Parameters (Negotiated):
                        Data Sequence In Order: yes
                        Data PDU In Order: yes
                        Default Time To Retain: 0
                        Default Time To Wait: 3
                        Error Recovery Level: 0
                        First Burst Length: 65536
                        Immediate Data: yes
                        Initial Ready To Transfer (R2T): yes
                        Max Burst Length: 262144
                        Max Outstanding R2T: 1
                        Max Receive Data Segment Length: 65536
                        Max Connections: 1
                        Header Digest: NONE
                        Data Digest: NONE
```

iSCSI イニシエータと iSCSI ターゲットのパラメータの変更

iSCSI イニシエータおよび iSCSI ターゲットデバイスのどちらでもパラメータは変更可能です。ただし、iSCSI イニシエータ上で変更可能なパラメータは、次のものだけです。

- iSCSI イニシエータのノード名 – イニシエータのノード名を別の名前に変更できます。イニシエータのノード名を変更した場合は、名前を変更したときの iSNS サーバー上の発見ドメインインストールの構成に応じて、iSNS によって発見されたターゲットがイニシエータのターゲットリストから削除されることがあります。詳細は、287 ページの「iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法」を参照してください。
- ヘッダーダイジェスト – NONE、デフォルト値、または CRC32。

- データダイジェスト – NONE、デフォルト値、または CRC32。
- 認証と CHAP シークレット – 認証の設定方法の詳細は、276 ページの「iSCSI イニシエータの CHAP 認証を構成する方法」を参照してください。
- 構成済みのセッション – 複数のセッションの構成方法の詳細は、291 ページの「ターゲットの複数の iSCSI セッションを有効にする方法」を参照してください。

iSCSI ドライバは、iSCSI イニシエータと iSCSI ターゲットデバイスのパラメータのデフォルト値を提供します。iSCSI イニシエータのパラメータを変更すると、iSCSI ターゲットデバイスにすでに別の値が設定されている場合以外、その変更後のパラメータが iSCSI ターゲットデバイスに継承されます。

注意 – 変更対象のパラメータがターゲットソフトウェアによってサポートされていることを確認してください。サポートされていない場合、iSCSI ターゲットデバイスにログインできない可能性があります。サポートされているパラメータの一覧については、使用するアレイのマニュアルを参照してください。

iSCSI パラメータの変更は、イニシエータとターゲット間の入出力が完了した時点で行うことをお勧めします。`iscsiadm modify` コマンドを使って変更が実行されると、iSCSI ドライバはセッションを接続し直します。

▼ iSCSI イニシエータと iSCSI ターゲットのパラメータを変更する方法

この手順の前半では、iSCSI イニシエータの変更されたパラメータが iSCSI ターゲットデバイスにどのように継承されるかを示します。この手順の後半では、iSCSI ターゲットデバイス上のパラメータを実際に変更する方法を示します。

この省略可能な手順では、ユーザーが現在ログインしているローカルシステム上で、iSCSI ターゲットデバイスへのアクセスがすでに構成されているものとします。

1. スーパーユーザーになります。
2. **iSCSI** イニシエータと **iSCSI** ターゲットデバイスの現在のパラメータを一覧表示します。

   a. **iSCSI** イニシエータの現在のパラメータを一覧表示します。次に例を示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
        Login Parameters (Default/Configured):
                Header Digest: NONE/-
                Data Digest: NONE/-
        Authentication Type: NONE
```

```
        RADIUS Server: NONE
        RADIUS access: unknown
        Configured Sessions: 1
```

b. **iSCSI** ターゲットデバイスの現在のパラメータを一覧表示します。次に例を示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
                Error Recovery Level: 0/-
                First Burst Length: 65536/-
                Immediate Data: yes/-
                Initial Ready To Transfer (R2T): yes/-
                Max Burst Length: 262144/-
                Max Outstanding R2T: 1/-
                Max Receive Data Segment Length: 65536/-
                Max Connections: 1/-
                Header Digest: NONE/-
                Data Digest: NONE/-
         Configured Sessions: 1
```

iSCSI イニシエータと iSCSI ターゲットデバイスの両方で、ヘッダーダイジェストとデータダイジェストのパラメータがどちらも現在 NONE に設定されています。

iSCSI ターゲットデバイスのデフォルトのパラメータを確認するには、Example 14–1 の 例 14–1 出力を参照してください。

3 **iSCSI** イニシエータのパラメータを変更します。

たとえば、ヘッダーダイジェストを CRC32 に設定します。

```
initiator# iscsiadm modify initiator-node -h CRC32
```

イニシエータのノード名を変更した場合は、新しい名前がターゲットと同じ発見ドメインに属していないと、iSNS によって発見されたターゲットがログアウトおよび削除されることがあります。ただし、ターゲットが使用中の場合は、削除されません。たとえば、これらのターゲット上でファイルが開いている場合またはファイルシステムがマウントされている場合は、それらのターゲットは削除されません。

これらのターゲットと新しいイニシエータノードの名前が同じ発見ドメインに属している場合は、名前の変更後に新しいターゲットも表示されることがあります。

4 パラメータが変更されたことを確認します。

a. **iSCSI** イニシエータの更新済みパラメータ情報を表示します。次に例を示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
```

```
        Login Parameters (Default/Configured):
                Header Digest: NONE/CRC32
                Data Digest: NONE/-
        Authentication Type: NONE
        RADIUS Server: NONE
        RADIUS access: unknown
        Configured Sessions: 1
```

ヘッダーダイジェストは CRC32 に設定されています。

b. iSCSI ターゲットデバイスの更新済みパラメータ情報を表示します。次に例を示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
                Error Recovery Level: 0/-
                First Burst Length: 65536/-
                Immediate Data: yes/-
                Initial Ready To Transfer (R2T): yes/-
                Max Burst Length: 262144/-
                Max Outstanding R2T: 1/-
                Max Receive Data Segment Length: 65536/-
                Max Connections: 1/-
                Header Digest: CRC32/-
                Data Digest: NONE/-
        Configured Sessions: 1
```

ヘッダーダイジェストは CRC32 に設定されています。

5 iSCSI イニシエータが iSCSI ターゲットに接続し直されたことを確認します。次に例を示します。

```
initiator# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        TPGT: 2
        ISID: 4000002a0000
        Connections: 1
                CID: 0
                  IP address (Local): nnn.nn.nn.nnn:64369
                  IP address (Peer): nnn.nn.nn.nnn:3260
                  Discovery Method: SendTargets
                  Login Parameters (Negotiated):
                        .
                        .
                        .
                        Header Digest: CRC32
                        Data Digest: NONE
```

6 **(省略可能) iSCSI イニシエータまたは iSCSI ターゲットデバイスのパラメータを設定解除します。**

`iscsiadm modify` コマンドを使用してデフォルトの設定に戻すことによってパラメータを設定解除できます。あるいは、`iscsiadm remove` コマンドを使ってすべてのターゲットプロパティーをデフォルト設定にリセットします。

`iscsiadm modify target-param` コマンドは、コマンド行で指定されたパラメータのみを変更します。

次の例では、ヘッダーダイジェストを none にリセットする方法を示します。

```
initiator# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

`iscsiadm remove target-param` コマンドについては、`iscsiadm(1M)` のマニュアルページを参照してください。

Solaris iSCSI マルチパスデバイスの設定

Solaris iSCSI マルチパス (MPxIO) デバイスを使用する場合には、次のガイドラインを考慮してください。

- **Solaris iSCSI** および **MPxIO** – MpxIO は、iSCSI イニシエータの MS/T (Multiple Sessions per Target) を設定する Solaris iSCSI 構成での、ターゲットポート集約と可用性をサポートします。
  - 複数の NIC を集約およびフェイルオーバーする場合は、IPMP を使用してください。
  - iSCSI ホストの基本的な構成は、iSCSI トラフィック専用の 2 つの NIC を備えたサーバーです。NIC は IPMP を使用して設定されます。パフォーマンスを最適化するために、iSCSI 以外のトラフィック用に追加の NIC が用意されます。
  - アクティブマルチパスは、Solaris iSCSI MS/T 機能、および IPMP 構成のフェイルオーバーと冗長性を使用する場合のみ実現できます。
    - IPMP 構成で 1 つの NIC に障害が発生すると、IPMP はフェイルオーバーの処理を行います。MPxIO ドライバでは障害は認識されません。IPMP 以外の構成では、MPxIO ドライバは障害状態になり、パスがオフラインになります。
    - IPMP 構成で 1 つのターゲットポートに障害が発生すると、MPxIO ドライバは障害を認識し、フェイルオーバーの処理を行います。IPMP 以外の構成では、MPxIO ドライバは障害を認識し、フェイルオーバーの処理を行います。

    Solaris iSCSI MS/T 機能を IPMP およびマルチパス化とともに使用する方法については、SunSolve Infodoc 207607「Understanding an iSCSI MS/T multi-path configuration」を参照してください。

    - MS/T の構成方法については、291 ページの「ターゲットの複数の iSCSI セッションを有効にする方法」を参照してください。IPMP の構成方法については、『Solaris のシステム管理 (IP サービス)』のパート VI「IPMP」を参照してください。
- **Solaris iSCSI**、ファイバチャネル **(FC)**、および **MPxIO** – より複雑な iSCSI/FC 構成では、MPxIO ドライバは次のように動作します。
    - FC SAN にデュアル iSCSI-FC ブリッジが存在する場合、iSCSI はターゲットパスを MPxIO に提示します。MPxIO は、一意の SCSI/LUN 識別子を照合し、それらが同一である場合は 1 つのパスを iSCSI ドライバに提示します。
    - iSCSI と FC の両方を使用してターゲットを接続する構成の場合、MPxIO ドライバは同じデバイスに対して異なるトランスポートを提供できます。この構成では、MPxIO は両方のパスを使用します。
    - iSCSI と FC を MPxIO と組み合わせて使用する場合は、`/kernel/drv/fp.conf` ファイルおよび `/kernel/drv/iscsi.conf` ファイルの MPxIO の設定が、サポートしたい MPxIO 構成と一致していることを確認します。たとえば、`fp.conf` では、MPxIO を HBA 全体で有効にするのか、ポート単位で有効にするのかを決定できます。
- 他社製ハードウェアに関する注意事項 – 他社製の HBA が Solaris iSCSI および MPxIO での使用に適しているかどうかを確認します。

  他社製の HBA を使用する場合は、その HBA ベンダーに `/kernel/drv/scsi_vhci.conf` ファイルの対称オプション情報を問い合わせる必要があります。

▼ ターゲットの複数の iSCSI セッションを有効にする方法

この手順を使用して、単一のターゲットに接続する複数の iSCSI セッションを作成できます。このシナリオは、ログインのリダイレクションをサポートするか、または同じターゲットポータルグループに含まれている複数のターゲットのポータルを使用する iSCSI ターゲットデバイスの場合に役に立ちます。ターゲットごとに複数の iSCSI セッションを使用するときは、Solaris SCSI Multipathing (MPxIO) を組み合わせて使用します。また、ホスト側の複数の NIC を使用して同じターゲット上の複数のポータルに接続すると、より広い帯域幅を実現できます。

MS/T 機能は、イニシエータのセッション ID (ISID) を変えることで、ターゲット上に 2 つ以上のセッションを作成します。この機能を有効にすると、ネットワークに 2 つの SCSI レイヤーパスが作成され、複数のターゲットを iSCSI レイヤーから Solaris I/O レイヤーまで公開できるようになります。MPxIO ドライバは、これらのパスに対する予約を処理します。

iSCSI と MPxIO パスの相互動作の仕組みについては、290 ページの「Solaris iSCSI マルチパスデバイスの設定」を参照してください。

iSCSI ターゲットの複数セッションを設定する前に、次の項目を確認してください。

- 通常の MS/T 構成には、2 つ以上の構成済みセッションがあります。

  ただし、ストレージが複数の TPGT をサポートしていて、ホストシステム上で SendTarget 発見を使用している場合には、構成済みセッションの数を 1 に設定できます。SendTarget 発見によって、複数のパスが存在することが自動的に検出されて複数のターゲットセッションが作成されます。
- `/kernel/drv/iscsi.conf` ファイルで `mxpio` 構成パラメータが有効になっていることを確認します。

  ```
  # cd /kernel/drv
  # grep mpxio iscsi.conf
  iscsi.conf:mpxio-disable="no";
  ```
- IPMP を使用して複数のネットワーク接続が構成されていることを確認します。
- 複数のネットワーク接続が使用可能であることを確認します。

  ```
  # ifconfig -a
  ```

1 スーパーユーザーになります。

2 **iSCSI** イニシエータと **iSCSI** ターゲットの現在のパラメータを一覧表示します。

   a. **iSCSI** イニシエータの現在のパラメータを一覧表示します。次に例を示します。

   ```
   initiator# iscsiadm list initiator-node
   Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
   Initiator node alias: zzr1200
           .
           .
           .
           Configured Sessions: 1
   ```

   b. **iSCSI** ターゲットデバイスの現在のパラメータを一覧表示します。次に例を示します。

   ```
   initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
   Target: iqn.1992-08.com.abcstorage:sn.84186266
           Alias: -
           .
           .
           .
           Configured Sessions: 1
   ```

   Configured Sessions (構成済みのセッション) の値は、ターゲットポータルグループ内の各ターゲット名用に作成される構成済みの iSCSI セッションの数です。

3 次のいずれかを選択し、構成済みのセッションの数をイニシエータノードで変更してすべてのターゲットに適用するか、ターゲットレベルで変更して特定のターゲットに適用します。

   ターゲットのセッション数は 1 から 4 の間である必要があります。

   - iSCSI イニシエータノードにパラメータを適用します。

次に例を示します。

```
initiator# iscsiadm modify initiator-node -c 2
```

- iSCSI ターゲットにパラメータを適用します。

  次に例を示します。

```
initiator# iscsiadm modify target-param -c 2  iqn.1992-08.com.abcstorage:sn.84186266
```

- 構成済みセッションを 1 つ以上のローカル IP アドレスにバインドします。

  構成済みのセッションは、特定のローカル IP アドレスにバインドすることもできます。この方法を使用する場合は、コンマ区切りのリストで 1 つ以上のローカル IP アドレスを指定します。各 IP アドレスは iSCSI セッションを表します。この方法は、イニシエータノードまたはターゲットパラメータのレベルでも実行できます。次に例を示します。

```
initiator# iscsiadm modify initiator-node -c 10.0.0.1,10.0.0.2
```

注 – 指定した IP アドレスがルーティング可能ではない場合、アドレスが無視され、デフォルトの Solaris ルートおよび IP アドレスがこのセッションで使用されます。

---

4 パラメータが変更されたことを確認します。

a. イニシエータノードの更新された情報を表示します。次に例を示します。

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
        .
        .
        .
        Configured Sessions: 2
```

b. ターゲットノードの更新された情報を表示します。次に例を示します。

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        Alias: -
        .
        .
        .
        Configured Sessions: 2
```

5 `mpathadm list lu` コマンドで複数パスを一覧表示して、**OS** デバイス名が `iscsiadm list` の出力に一致していることと、パスの数が **2** 以上であることを確認します。

iSCSI 構成に関する問題の障害追跡

一般的な iSCSI 構成に関する問題の障害追跡を行う際に利用可能なツールを、次に示します。

- `snoop` – このツールは更新され、iSCSI パケットをサポートするようになりました。
- `wireshark` – この製品は `http://www.wireshark.org/` から入手できます。

どちらのツールも、ポート 3260 上の iSCSI パケットをフィルタできます。

次の各節では、iSCSI のさまざまな障害追跡やエラーメッセージ解決シナリオについて説明します。

ローカルシステムから iSCSI ターゲットに接続できない

▼ iSCSI の接続に関する問題の障害追跡を行う方法

1. スーパーユーザーになります。
2. iSCSI ターゲットの情報を一覧表示します。

   次に例を示します。

   ```
   initiator# iscsiadm list target
   Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
           TPGT: default
           ISID: 4000002a0000
           Connections: 0
   ```

3. `iscsiadm list target` の出力結果に接続が表示されていない場合は、接続が失敗したことについて、`/var/adm/messages` ファイル内で考えられる理由を調査します。

   また、`ping` コマンドを使用するか、または、`telnet` コマンドを使ってストレージデバイスの iSCSI ポートに接続し、iSCSI サービスが利用可能かどうかを確認することによって、接続がアクセス可能かどうかを確認することもできます。デフォルトのポートは 3260 です。

   さらに、ストレージデバイスのログファイルでエラーを確認します。

4 `iscsiadm list target` の出力結果に目的のターゲットが表示されていない場合は、`/var/adm/messages` ファイル内でその考えられる原因を調査します。

SendTargets を発見方式として使用している場合は、`-v` オプションを使って *discovery-address* の一覧表示を試みると、ホストが目的のターゲットを認識しているかどうかがわかります。次に例を示します。

```
initiator# iscsiadm list discovery-address -v 10.0.0.1
Discovery Address: 10.0.0.1:3260
    Target name: eui.210000203787dfc0
        Target address:        10.0.0.1:11824
    Target name: eui.210000203787e07b
        Target address:        10.0.0.1:11824
```

発見方式として iSNS を使用している場合は、iSNS 発見方式を有効にし、*v* オプションを使用して `-isns-server` を一覧表示することで、期待されるターゲットがホストに認識されることを確認します。次に例を示します。

```
initiator# iscsiadm list isns-server -v
iSNS Server IP Address: 10.20.56.56:3205
        Target name: iqn.1992-08.com.xyz:sn.1234566
                Target address:   10.20.57.161:3260, 1
        Target name: iqn.2003-10.com.abc:group-0:154:abc-65-01
                Target address:   10.20.56.206:3260, 1
        Target name: iqn.2003-10.com.abc:group-0:154:abc-65-02
                Target address:   10.20.56.206:3260, 1
.
.
.
```

ローカルシステム上で iSCSI デバイスまたは iSCSI ディスクが利用できない

▼ iSCSI デバイスまたは iSCSI ディスクが利用できない問題の障害追跡を行う方法

1 スーパーユーザーになります。

2 列挙中にこのターゲット上で発見された LUN を確認します。

次に例を示します。

```
# iscsiadm list target -S
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
        TPGT: default
        ISID: 4000002a0000
        Connections: 1
        LUN: 0
            Vendor:  ABCSTOR
            Product: 0010
            OS Device Name: /dev/rdsk/c3t34d0s2
```

-S オプションを指定すると、列挙中にこのターゲット上で発見された LUN が表示されます。表示されるはずの LUN が表示されない場合は、/var/adm/messages ファイル内でエラーが報告されていないか確認します。ストレージデバイスのログファイルにエラーが記録されていないか確認します。また、ストレージデバイスのすべての LUN マスクが正しく構成されていることも確認します。

iSNS 発見方式を使用する場合の LUN マスクの使用

特定のイニシエータに対するストレージの認証を制御する手段として iSNS 発見ドメインを使用しないでください。承認されたイニシエータのみが LUN にアクセスできることを確認する場合は、代わりに LUN マスクを使用します。

ターゲットが使用中のときに発見ドメインからターゲットを削除しようとした場合、iSCSI イニシエータはこのターゲットからログアウトしません。このイニシエータがこのターゲット(および関連付けられた LUN) にアクセスしないようにする場合は、LUN マスクを使用する必要があります。ターゲットを発見ドメインから削除するだけでは不十分です。

iSCSI の一般的なエラーメッセージ

この節では、/var/adm/messages ファイルに見つかる可能性のある iSCSI メッセージと、回復のために適用できる解決方法について説明します。

メッセージの形式は次のとおりです。

iscsi *TYPE* (*OID*) *STRING* (*STATUS-CLASS#/STATUS-DETAIL#*)

TYPE	接続 (connection)、セッション (session) のいずれかです。
OID	接続またはセッションのオブジェクト ID です。この ID は、OS インスタンスに一意です。
STRING	状態の説明です。
STATUS-CLASS#/STATUS-DETAIL#	これらの値は、RFC 3720 で定義されている iSCSI ログイン応答として返されます。

iscsi connection(*OID*) login failed - Miscellaneous iSCSI initiator errors.
原因:何らかのイニシエータエラーにより、デバイスログインが失敗しました。

iscsi connection(*OID*) login failed - Initiator could not be successfully authenticated.
原因:デバイスによるイニシエータの認証が成功しませんでした。

対処方法: 該当する場合には、CHAP 名、CHAP パスワード、または RADIUS サーバーの設定が正しいことを確認します。

`iscsi connection(`*OID*`) login failed - Initiator is not allowed access to the given target.`

原因: イニシエータが iSCSI ターゲットデバイスにアクセスすることを、デバイスが許可できません。

対処方法: イニシエータ名を確認するとともに、その名前がストレージデバイスによって正しくマスクまたはプロビジョニングされていることを確認します。

`iscsi connection(`*OID*`) login failed - Requested ITN does not exist at this address.`

原因: ユーザーが要求する iSCSI ターゲット名 (ITN) へのアクセスを、デバイスが提供しません。

対処方法: イニシエータの発見情報が正しく指定されており、ストレージデバイスが正しく構成されていることを確認します。

`iscsi connection(`*OID*`) login failed - Requested ITN has been removed and no forwarding address is provided.`

原因: ユーザーが要求する iSCSI ターゲット名 (ITN) へのアクセスを、デバイスが提供しなくなりました。

対処方法: イニシエータの発見情報が正しく入力されており、ストレージデバイスが正しく構成されていることを確認します。

`iscsi connection(`*OID*`) login failed - Requested iSCSI version range is not supported by the target.`

原因: イニシエータの iSCSI バージョンがストレージデバイスによってサポートされていません。

`iscsi connection(`*OID*`) login failed - No more connections can be accepted on this Session ID (SSID).`

原因: ストレージデバイスは、このイニシエータノードから iSCSI ターゲットデバイスへの接続をこれ以上受け入れることができません。

`iscsi connection(`*OID*`) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).`

原因: ストレージデバイスが、イニシエータ名またはターゲット名が正しく指定されていないと報告しています。

対処方法: iSCSI のイニシエータ名またはターゲット名を正しく指定します。

`iscsi connection(`*OID*`) login failed - Target hardware or software error.`
原因: ストレージデバイスでハードウェアエラーまたはソフトウェアエラーが発生しました。

対処方法: ストレージのマニュアルを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

`iscsi connection(`*OID*`) login failed - iSCSI service or target is not currently operational.`
原因: ストレージデバイスが現在動作していません。

対処方法: ストレージのマニュアルを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

`iscsi connection(`*OID*`) login failed - Target has insufficient session, connection or other resources.`
原因: ストレージデバイスのリソースが不足しています。

対処方法: ストレージのマニュアルを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

`iscsi connection(`*OID*`) login failed - unable to initialize authentication`

`iscsi connection(`*OID*`) login failed - unable to set authentication`

`iscsi connection(`*OID*`) login failed - unable to set username`

`iscsi connection(`*OID*`) login failed - unable to set password`

`iscsi connection(`*OID*`) login failed - unable to set ipsec`

`iscsi connection(`*OID*`) login failed - unable to set remote authentication`
原因: イニシエータが認証の初期化または設定を正しく行えませんでした。

対処方法: イニシエータの認証が正しく設定されていることを確認します。

`iscsi connection(`*OID*`) login failed - unable to make login pdu`
原因: イニシエータが、イニシエータまたはストレージデバイスの設定に基づいてログインのペイロードデータユニット (PDU) を作成できませんでした。

対処方法: 任意のターゲットログインパラメータやその他のデフォルト以外の設定値のリセットを試みます。

`iscsi connection(`*OID*`) login failed - failed to transfer login`

`iscsi connection(`*OID*`) login failed - failed to receive login response`
原因: イニシエータが、ログインのペイロードデータユニット (PDU) をネットワーク接続経由で転送または受信できませんでした。

対処方法: ネットワーク接続が到達可能であることを確認します。

`iscsi connection(`*OID*`) login failed - received invalid login response (`*OP CODE*`)`
原因: ストレージデバイスがログインに対して予想外の応答を返しました。

`iscsi connection(`*OID*`) login failed - login failed to authenticate with target`
原因: イニシエータがストレージデバイスを認証できませんでした。

対処方法: イニシエータの認証が正しく設定されていることを確認します。

`iscsi connection(`*OID*`) login failed - initiator name is required`
原因: どのアクションを実行する場合も、イニシエータ名が構成されている必要があります。

対処方法: イニシエータ名が構成されていることを確認します。

`iscsi connection(`*OID*`) login failed - authentication receive failed`
`iscsi connection(`*OID*`) login failed - authentication transmit failed`
原因: イニシエータが認証情報を転送または受信できませんでした。

対処方法: 状況に応じて、ストレージデバイスまたは RADIUS サーバーとのネットワーク接続を確認します。

`iscsi connection(`*OID*`) login failed - login redirection invalid`
原因: ストレージデバイスが、イニシエータを無効な宛先にリダイレクトしようとしました。

対処方法: ストレージのマニュアルを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

`iscsi connection(`*OID*`) login failed - target protocol group tag mismatch, expected <TPGT>, received <TPGT>`
原因: イニシエータとターゲットの TPGT (ターゲットポータルグループタグ、Target Portal Group Tag) が一致しません。

対処方法: イニシエータ上またはストレージデバイス上の TPGT 発見設定を確認します。

`iscsi connection(`*OID*`) login failed - can't accept `*PARAMETER*` in security stage`
原因: ログインのセキュリティーフェーズで、デバイスがサポートされていないログインパラメータで応答しました。

対処方法: パラメータ名が参考のため記載されています。ストレージのマニュアルを参照するか、ストレージのベンダーに連絡して必要なサポートを受けます。

`iscsi connection(`*OID*`) login failed - HeaderDigest=CRC32 is required, can't accept` *VALUE*

`iscsi connection(`*OID*`) login failed - DataDigest=CRC32 is required, can't accept` *VALUE*

原因: このターゲットに対して、`CRC32` に設定された `HeaderDigest` または `DataDigest` のみを受け入れるようにイニシエータが構成されています。デバイスは値 *VALUE* を返しました。

対処方法: イニシエータとデバイスのダイジェスト設定に互換性があることを確認します。

`iscsi connection(`*OID*`) login failed - HeaderDigest=None is required, can't accept` *VALUE*

`iscsi connection(`*OID*`) login failed - DataDigest=None is required, can't accept` *VALUE*

原因: このターゲットに対して、NONE に設定された `HeaderDigest` または `DataDigest` のみを受け入れるようにイニシエータが構成されています。デバイスは値 *VALUE* を返しました。

対処方法: イニシエータとデバイスのダイジェスト設定に互換性があることを確認します。

`iscsi connection(`*OID*`) login failed - can't accept` *PARAMETER*

原因: イニシエータはこのパラメータをサポートしません。

`iscsi connection(`*OID*`) login failed - can't accept MaxOutstandingR2T` *VALUE*

原因: イニシエータは、記載された *VALUE* の `MaxOutstandingR2T` を受け入れません。

`iscsi connection(`*OID*`) login failed - can't accept MaxConnections` *VALUE*

原因: イニシエータは、記載された *VALUE* の最大接続数を受け入れません。

`iscsi connection(`*OID*`) login failed - can't accept ErrorRecoveryLevel` *VALUE*

原因: イニシエータは、記載された *VALUE* のエラー回復レベルを受け入れません。

`iscsi session(`*OID*`)` *NAME* `offline`

原因: このターゲット *NAME* のすべての接続が、削除されたか、または失敗しました。

`iscsi connection(`*OID*`) failure - unable to schedule enumeration`

原因: イニシエータがこのターゲットの LUN を列挙できませんでした。

対処方法: LUN の列挙を強制実行するには、`devfsadm -i iscsi` コマンドを実行します。詳細は、`devfsadm(1M)` のマニュアルページを参照してください。

`iscsi connection(`*OID*`) unable to connect to target` *NAME* `(errno:` *ERRNO* `)`
原因: イニシエータによるネットワーク接続の確立が失敗しました。

対処方法: 接続エラーに関する特定の *ERRNO* については、`/usr/include/sys/errno.h` ファイルを参照してください。

◆ ◆ ◆ 第 15 章

15 formatユーティリティー(参照情報)

この章では、formatユーティリティー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。

- 303 ページの「formatユーティリティーを使用する上での推奨事項および要件」
- 304 ページの「formatのメニューとコマンドの説明」
- 310 ページの「format.datファイル」
- 315 ページの「formatコマンドへの入力規則」
- 317 ページの「formatユーティリティーのヘルプを利用する」

formatユーティリティーの使用方法の概要については、207 ページの「formatユーティリティー」を参照してください。

formatユーティリティーを使用する上での推奨事項および要件

formatユーティリティーを使用するには、スーパーユーザーになるか、同等の役割を引き受ける必要があります。そうしないと、formatユーティリティーを使用しようとしたときに次のようなエラーメッセージが表示されます。

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

既存のデータを維持しつつformatユーティリティーを使用する場合、次の指針に従って操作を行なってください。

- ディスクドライブ上のすべてのファイルのバックアップを作成します。
- formatユーティリティーのdumpコマンドを使用して、すべての欠陥領域リストをファイルに保存します。ファイル名には、ドライブタイプ、モデル番号、およびシリアル番号を含めるべきです。

- 製造元から出荷時にドライブといっしょに提供された、欠陥領域リストを保管します。

format のメニューとコマンドの説明

次のような format のメインメニューが表示されます。

```
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
        fdisk      - run the fdisk program (x86 only)
        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
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

次の表に、format ユーティリティーのメインメニューの項目を示します。

表 15–1 format ユーティリティーのメインメニュー項目の説明

メニュー項目	コマンド/メニュー	説明
disk	コマンド	システムのドライブをすべて表示します。あとの操作で使用するディスクを選択することもできます。このディスクは、「現在のディスク」と呼ばれます。
type	コマンド	現在のディスクの製造元とモデルを表示します。認識されているドライブタイプのリストも表示します。SCSI-2 対応ディスクドライブの場合は必ず Auto configure オプションを選択します。
partition	メニュー	スライスの作成および変更を行います。詳細は、306 ページの「partition メニュー」を参照してください。
current	コマンド	現在のディスクに関する次の情報を表示します。 ■ デバイス名とデバイスタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ 物理デバイス名

表 15–1 format ユーティリティーのメインメニュー項目の説明 (続き)

メニュー項目	コマンド/メニュー	説明
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します。 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 エントリが存在しない場合に、プロンプトへの応答として入力する情報 このコマンドは、IDE ディスクには適用できません。IDE ディスクは、あらかじめ製造元でフォーマットされます。
fdisk	メニュー	x86 プラットフォームのみ: fdisk プログラムを実行し、Solaris fdisk パーティションを作成します。 fdisk コマンドは、1T バイトを超えるサイズの EFI ラベル付きディスクで使用することはできません。
repair	コマンド	現在のディスク上で特定のブロックを修復します。
label	コマンド	現在のディスクに新しいラベルを書き込みます。
analyze	メニュー	読み取り、書き込み、および比較テストを実行します。詳細は、308 ページの「analyze メニュー」を参照してください。
defect	メニュー	欠陥リストを検索して表示します。詳細は、309 ページの「defect メニュー」を参照してください。この機能は、IDE ディスクには適用できません。IDE ディスクは欠陥の検出を自動的に行います。
backup	コマンド	**VTOC** – バックアップラベルを検索します。 **EFI** – サポートされません。
verify	コマンド	現在のディスクに関する次の情報を表示します。 ■ デバイス名とデバイスタイプ ■ シリンダ数、代替シリンダ数、ヘッド数、セクター数 ■ パーティションテーブル
save	コマンド	**VTOC** – 新しいディスク情報およびパーティション情報を保存します。 **EFI** – 適用できません。
inquiry	コマンド	**SCSI** ディスクのみ – 現在のドライブのベンダー、製品名、リビジョンレベルを表示します。
volname	コマンド	8 文字のボリューム名を新規に指定して、ディスクラベルを作成します。

表 15–1 format ユーティリティーのメインメニュー項目の説明 (続き)

メニュー項目	コマンド/メニュー	説明
quit	コマンド	format メニューを終了します。

partition メニュー

partition メニューは次のようになっています。

```
format> partition
PARTITION MENU:
        0      - change ‘0’ partition
        1      - change ‘1’ partition
        2      - change ‘2’ partition
        3      - change ‘3’ partition
        4      - change ‘4’ partition
        5      - change ‘5’ partition
        6      - change ‘6’ partition
        7      - change ‘7’ partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        quit
partition>
```

次の表に、partition メニューの項目を示します。

表 15–2 partition メニュー項目の説明

サブコマンド	説明
change ‘*n*’ partition	新しいパーティションに次の情報を指定できます。 ■ 識別タグ ■ アクセス権フラグ ■ 開始シリンダ ■ サイズ
select	事前定義済みのパーティションテーブルを選択できます。
modify	パーティションテーブル内のすべてのスライスを変更できます。個々のスライスに対して実行する change ‘*x*’ partition コマンドよりも、このコマンドが多く使用されます。
name	現在のパーティションテーブルの名前を指定できます。
print	現在のパーティションテーブルを表示します。
label	パーティションマップとラベルを現在のディスクに書き込みます。

表 15–2 partition メニュー項目の説明 (続き)

サブコマンド	説明
quit	partition メニューを終了します。

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 14169 cylinders
            Cylinder size is 2510 (512 byte) blocks

                                             Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1       Active    x86 Boot          1     9       9      0
          2                 Solaris2         10  14168    14159    100

SELECT ONE OF THE FOLLOWING:

  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
  Enter Selection:
```

次の表に、fdisk メニューの項目を示します。

表 15–3 x86: fdisk メニュー項目の説明

メニュー項目	説明
Create a partition	fdisk パーティションを作成します。Solaris や DOS など、OS ごとに別々のパーティションを作成しなければなりません。1 台のディスクの最大パーティション数は 4 です。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ます。
Specify the active partition	ブートに使用するパーティションを指定できます。このメニュー項目により、第 1 段階のブートプログラムが第 2 段階のブートプログラムを検索する場所を指定します。
Delete a partition	以前に作成したパーティションを削除します。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ます。
Change between Solaris and Solaris2 Partition IDs	パーティション識別子を 130 (0x82) から 191 (0xbf) に、または 191 (0xbf) から 130 (0x82) に変更します。

表 15–3 x86: fdisk メニュー項目の説明 (続き)

メニュー項目	説明
Exit (update disk configuration and exit)	新しいパーティションテーブルを書き込んで fdisk メニューを終了します。
Cancel (exit without updating disk configuration)	パーティションテーブルを変更せずに fdisk メニューを終了します。

analyze メニュー

analyze メニューは次のようになっています。

```
format> analyze

ANALYZE MENU:
    read     - read only test   (doesn't harm SunOS)
    refresh  - read then write  (doesn't harm data)
    test     - pattern testing  (doesn't harm data)
    write    - write then read      (corrupts data)
    compare  - write, read, compare (corrupts data)
    purge    - write, read, write   (corrupts data)
    verify   - write entire disk, then verify (corrupts data)
    print    - display data buffer
    setup    - set analysis parameters
    config   - show analysis parameters
    quit
analyze>
```

次の表に、analyze メニューの項目を示します。

表 15–4 analyze メニュー項目の説明

サブコマンド	説明
read	現在のディスクの各セクターを読み込みます。デフォルトで欠陥ブロックを修復します。
refresh	データを損なわずに、現在のディスク上で読み取りおよび書き込みを実行します。デフォルトで欠陥ブロックを修復します。
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みます。デフォルトで欠陥ブロックを修復します。
write	一連のパターンをディスクに書き込んでから、そのデータをディスクから読み込みます。ディスク上の既存のデータは破壊されます。デフォルトで欠陥ブロックを修復します。
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、書き込みバッファー内のデータと比較します。ディスク上の既存のデータは破壊されます。デフォルトで欠陥ブロックを修復します。

表 15–4 analyze メニュー項目の説明 (続き)

サブコマンド	説明
purge	ディスク上のデータをすべて削除し、いかなる手段でも取り出せないようにします。ディスク全体 (またはディスクのセクション) に 3 種類のパターンを書き込むことにより、データを削除します。検査に合格すると16進のビットパターンがディスク全体 (またはディスクのセクション) に上書きされます。 デフォルトで欠陥ブロックを修復します。
verify	最初の検査に合格すると、ディスク全体の各ブロックに固有のデータを書き込みます。次の検査に合格すると、データを読み取り、検証します。ディスク上の既存のデータは破壊されます。デフォルトで欠陥ブロックを修復します。
print	読み取り / 書き込みバッファー内のデータを表示します。
setup	次の解析パラメータを指定できます。 Analyze entire disk? yes Starting block number: *depends on drive* Ending block number: *depends on drive* 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*/*nn*) Verify media after formatting? yes Enable extended messages? no Restore defect list? yes Restore disk label? yes
config	現在の解析パラメータを表示します。
quit	analyze メニューを終了します。

defect メニュー

defect メニューは次のようになっています。

```
format> defect

DEFECT MENU:
        primary  - extract manufacturer's defect list
        grown    - extract manufacturer's and repaired defects lists
        both     - extract both primary and grown defects lists
        print    - display working list
        dump     - dump working list to file
        quit
defect>
```

次の表に、defect メニューの項目を示します。

表 15–5 defect メニュー項目の説明

サブコマンド	説明
主ノード	製造元の欠陥リストをディスクドライブから読み込み、メモリー内の欠陥リストを更新します。
grown	増分の欠陥リストを読み取り、メモリー内の欠陥リストを更新します。「増分の欠陥」とは、解析中に検出された欠陥のことです。
both	製造元の欠陥リストと増分の欠陥リストの両方を読み取ります。その後、メモリー内欠陥リストを更新します。
print	メモリー内の欠陥リストを表示します。
dump	メモリー内の欠陥リストをファイルに保存します。
quit	defect メニューを終了します。

format.dat ファイル

Oracle Solaris OS と一緒に出荷される format.dat ファイルでは、多数の標準的なディスクがサポートされます。使用しているディスクドライブが format.dat ファイルに記載されていない場合は、次のようにします。

- format.dat ファイルに使用しているディスクのエントリを追加します。
- format ユーティリティーで、type コマンドと other オプションを選択して、エントリを追加します。

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 ファイルをほかのシステム上で使用する場合は、format.dat ファイルに追加する特定のディスクドライブを使用するシステムごとに、このファイルをコピーしてください。

次のいずれかの場合には、システムの /etc/format.dat ファイルを変更する必要があります。

- ディスクが Oracle Solaris OS でサポートされない場合
- パーティションテーブルが入っているディスクが、Oracle Solaris OS のデフォルト構成とは異なる場合

注 – /etc/format.dat ファイルのデフォルトエントリは変更しないでください。デフォルトエントリを変更する場合は、混乱を避けるために、そのエントリをコピーし、別の名前を付けてから変更します。

/etc/format.dat は、EFI ラベル付きディスクには適用できません。

format.dat ファイルの内容

format.dat ファイルには、format ユーティリティーに使用されるディスクドライブ情報が入っています。format.dat ファイル内では、次の 3 つの項目が定義されています。

- 検索パス
- ディスクタイプ
- スライステーブル

format.dat ファイルの構文

/etc/format.dat ファイルには、次の構文規則が適用されます。

- ポンド記号 (#) はコメント文字です。ポンド記号に続く 1 行のテキストは、format ユーティリティーでは解釈されません。
- format.dat ファイル内の各定義は、1 つの論理行で評価されます。定義が長すぎて 1 行に収まらない場合は、定義の最終行を除くすべての行末にバックスラッシュ (\) を付けなければなりません。
- 定義は、左辺に識別子、右辺に 1 つまたは複数の値を持つ一連の代入式から成ります。代入演算子は等号 (=) です。定義内の代入式はコロン (:) で区切らなければなりません。
- format ユーティリティーは、空白を無視します。代入値に空白を含める場合は、値全体を二重引用符 (") で囲みます。この構文により、引用符の内側の空白は代入値の一部として保持されます。
- 代入式によっては、右辺に複数の値を指定できるものがあります。値はコンマ (,) で区切ります。

format.dat ファイル中のキーワード

format.dat ファイルには、format ユーティリティーが起動時に読み込むディスク定義が含まれます。各定義の先頭には、 キーワード disk_type または partition が付きます。これらのキーワードについて、次の表で説明します。

表 15–6 format.dat ファイルのキーワードの説明

キーワード	説明
disk_type	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、Oracle Solaris OS でサポートされるコントローラとディスクの定義が入っています。 サポートされないディスクを使用する場合にのみ、新しい disk_type 定義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。
partition	特定のディスクタイプのパーティションテーブルを定義します。パーティションテーブルには、パーティション情報だけでなく、format ユーティリティー内で参照可能な名前が入っています。デフォルトの format.dat ファイルには、数種類のディスクドライブに対応するデフォルトのパーティション定義が含まれます。システムのディスク上にパーティションを作成し直した場合は、パーティション定義を追加します。必要に応じて、パーティション情報をデータファイルにいくつ追加してもかまいません。

ディスクタイプ (format.dat)

format.dat ファイル内の disk_type キーワードは、コントローラとディスクのモデルを定義します。各 disk_type 定義には、ディスクの物理ジオメトリに関する情報が含まれます。デフォルトの format.dat ファイルには、Oracle Solaris OSでサポートされるコントローラとディスクの定義が入っています。サポートされないディスクを使用する場合にのみ、新しい disk_type を追加する必要があります。必要に応じて、disk_type 定義をデータファイルにいくつ追加してもかまいません。

キーワード自体が、ディスクタイプ名になります。この名前は、ディスクのラベルの一部になり、format ユーティリティーの実行時にディスクタイプを識別するために使用されます。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべての disk_type 定義でキーワードのほかに割り当てなければならない識別子を示します。

表 15–7 必須の disk_type 識別子 (format.dat)

識別子	説明
ctlr	ディスクタイプで有効なコントローラのタイプ。現在、有効な値は SCSI と ATA です。
ncyl	ディスクタイプ内のデータシリンダ数。この数によって、システムがアクセスできるディスクの論理シリンダ数が決まります。
acyl	ディスクタイプ内の代替シリンダ数。format ユーティリティーは、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します。代替シリンダとして、常に 2 つ以上のシリンダを残しておく必要があります。

表 15–7 必須の disk_type 識別子 (format.dat) (続き)

識別子	説明
pcyl	ディスクタイプ内の物理シリンダ数。この数値は、ディスクメディアの境界を計算するために使用されます。通常、この数値は ncyl と acyl の合計に等しくなります。
nhead	ディスクタイプ内のヘッド数。この数値は、ディスクメディアの境界を計算するために使用されます。
nsect	ディスクタイプ内の 1 トラック当たりのデータセクター数。この数値は、ディスクメディアの境界を計算するために使用されます。この数値にはデータセクター以外は含まれません。スペアは、各トラックのデータセクション数には含まれません。
rpm	ディスクタイプの 1 分当たりの回転数。この情報はラベルに書き込まれ、あと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ます。

コントローラによっては、ほかの識別子が必要な場合があります。次の表に、SCSI コントローラに必要な識別子を示します。

表 15–8 SCSI コントローラに必要な disk_type 識別子 (format.dat)

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を指定します。詳細は、コントローラのマニュアルを参照してください。
cache	format ユーティリティー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を指定します。詳細は、コントローラのマニュアルを参照してください。
trks_zone	代替セクターのマッピング内で使用される 1 つの欠陥ゾーン当たりのトラック数を指定した数値を指定します。詳細は、コントローラのマニュアルを参照してください。
asect	所定の欠陥ゾーン内で代替マッピングに利用可能なセクター数を指定します。詳細は、コントローラのマニュアルを参照してください。

例 15–1 SCSI コントローラに必要な disk_type 識別子 (format.dat)

次に、disk_type 定義の例を示します。

```
disk_type = "SUN1.3G" \
        : ctlr = SCSI : fmt_time = 4 \
        : trks_zone = 17 : asect = 6 : atrks = 17 \
        : ncyl = 1965 : acyl = 2 : pcyl = 3500 : nhead = 17 : nsect = 80 \
        : rpm = 5400 : bpt = 44823

disk_type = "SUN2.1G" \
        : ctlr = SCSI : fmt_time = 4 \
        : ncyl = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \
        : rpm = 5400 : bpt = 44823
```

例 15–1　SCSI コントローラに必要な disk_type 識別子 (format.dat)　(続き)

```
disk_type = "SUN2.9G" \
        : ctlr = SCSI : fmt_time = 4 \
        : ncyl = 2734 : acyl = 2 : pcyl = 3500 : nhead = 21 : nsect = 99 \
        : rpm = 5400
```

パーティションテーブル (format.dat)

format.dat ファイル内のパーティションテーブルに、特定のディスクタイプのスライステーブルが定義されています。

format.dat ファイル内の partition キーワードが、パーティションテーブル名になります。空白が含まれている名前は、二重引用符で囲んでください。次の表に、すべてのパーティション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 15–9　パーティションテーブルに必須の識別子 (format.dat)

識別子	説明
disk	このパーティションテーブルが定義されている disk_type の名前。この名前は disk_type 内で使用されるとおりに指定しなければなりません。
ctlr	このパーティションテーブルを接続できるディスクコントローラタイプ。現在、有効な値は ATA コントローラを表す ATA と SCSI コントローラを表す SCSI です。ここで指定したコントローラタイプは、disk_type 定義で選択した disk_type にも定義する必要があります。

スライス定義内のほかの識別子では、実際のパーティション情報を記述します。識別子は 0 - 7 の番号です。これらの識別子は省略可能です。明示的に割り当てられていないパーティションは、長さ 0 に設定されます。これらの識別子の値は、それぞれコンマで区切られた数値のペアになります。最初の数値は、パーティションの開始シリンダを表します。2 番目の数値は、スライス内のセクター数を表します。

例 15–2　パーティションテーブルに必要な識別子 (format.dat)

次に、スライス定義の例を示します。

```
partition = "SUN1.3G" \
        : disk = "SUN1.3G" : ctlr = SCSI \
        : 0 = 0, 34000 : 1 = 25, 133280 : 2 = 0, 2672400 : 6 = 123, 2505120

partition = "SUN2.1G" \
        : disk = "SUN2.1G" : ctlr = SCSI \
        : 0 = 0, 62320 : 1 = 41, 197600 : 2 = 0, 4154160 : 6 = 171, 3894240

partition = "SUN2.9G" \
        : disk = "SUN2.9G" : ctlr = SCSI \
```

例 15–2　パーティションテーブルに必要な識別子 (format.dat)　(続き)

```
: 0 = 0, 195426 : 1 = 94, 390852 : 2 = 0, 5683986 : 6 = 282, 5097708
```

format ユーティリティーの代替データファイルを指定する

format ユーティリティーは、次の順番で、代替ファイルの位置を認識します。

1. format -x オプションでファイル名を指定した場合、ファイルは常にデータファイルとして使用されます。
2. -x オプションを指定しない場合、format ユーティリティー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い場合、format ユーティリティー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 Oracle Solaris OS と一緒に出荷されるので、必ず存在します。

format コマンドへの入力規則

format ユーティリティーを使用する場合は、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。データ指定時に format のヘルプ機能を使用する方法については、317 ページの「format ユーティリティーのヘルプを利用する」を参照してください。

format コマンドへ番号を指定する

format ユーティリティーを使用する際、数値を入力する必要があります。入力方法には、適切なデータを指定する方法と、選択肢のリストから番号を選択する方法があります。どちらの場合も、ヘルプ機能を使用すると、format は期待する数値の上限と下限を表示します。適切な数値を入力するだけで済みます。数値は、その一部として底を明示的に指定しない限り (16 進数を表す 0x など)、10 進数と見なされます。

次の例は、整数の入力を示しています。

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

format コマンドへブロック番号を指定する

ディスクブロック番号を指定する必要がある場合、次の 2 つの方法のいずれかを選択できます。

- ブロック番号を表す整数を指定します
- シリンダ / ヘッド / セクター書式でブロック番号を指定します

この情報は、論理ブロック番号を表す整数として指定できます。任意の底の数値を指定できますが、デフォルトは 10 進です。また、ここで最大演算子 (ドル記号 $) を使用して、format ユーティリティー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。

シリンダ / ヘッド / セクター書式を使ってブロック番号を指定する方法もあります。この方法では、ブロック番号の 3 つの論理構成要素である、シリンダ、ヘッド、およびセクターの値を明示的に指定しなければなりません。 これらの値も論理値です。ただし、メディアのレイアウトに関連したディスク領域を定義できます。

シリンダ / ヘッド / セクター番号を指定しない場合、値は 0 であると見なされます。数値の代わりに、最大演算子を使用してもかまいません。すると、format ユーティリティーにより、適切な値が選択されます。シリンダ、ヘッド、セクターの値の例は次のとおりです。

```
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 ユーティリティーでは、ブロック番号は常に上記の両方の書式で出力されます。また、ヘルプ機能によって、期待されるブロック番号の上限と下限が両方の書式で表示されます。

format のコマンド名を指定する

format ユーティリティーでメニュープロンプトを表示する際、入力としてコマンド名が必要になります。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 (artition) を使用して format メニューから partition メニューにアクセスできます。次に、p(rint) を使用して現在のスライステーブルを表示できます。

```
format> p
PARTITION MENU:
        0      - change ‘0’ partition
        1      - change ‘1’ partition
        2      - change ‘2’ partition
        3      - change ‘3’ partition
        4      - change ‘4’ partition
        5      - change ‘5’ partition
        6      - change ‘6’ partition
        7      - change ‘7’ partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        quit
partition> p
```

format コマンドへディスク名を指定する

format ユーティリティーでは、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、名前に使用する文字列を自由に指定できます。空白を含む名前は、二重引用符 (") で囲まなければなりません。二重引用符で囲まなければ、名前の最初の語だけが使用されます。

たとえば、ディスクの特定のパーティションテーブルを指定する場合、partition メニューの name サブコマンドを使用できます。

```
partition> name
Enter table name (remember quotes): "new disk3"
```

format ユーティリティーのヘルプを利用する

format ユーティリティーにはヘルプ機能が組み込まれており、format ユーティリティーが入力待ちの状態であればいつでも使用できます。疑問符 (?) を入力するだけで必要な入力に関するヘルプが表示されます。format ユーティリティーでは、どんなタイプの入力が必要かについて簡潔な説明が表示されます。

メニュープロンプトから ? と入力すると、利用できるコマンドのリストが表示されます。

format ユーティリティーに関連するマニュアルページには、次が含まれます。

- format(1M) – format ユーティリティーの基本機能およびコマンド行で使用可能なすべての変数について説明します。
- format.dat(4) – format ユーティリティーで使用するディスクドライブ構成に関する情報を提供します。

◆ ◆ ◆ 第 16 章

ファイルシステムの管理 (概要)

ファイルシステムの管理は、もっとも重要なシステム管理作業の 1 つです。

この章の内容は以下のとおりです。

- 319 ページの「ファイルシステムの新機能」
- 327 ページの「ファイルシステム管理作業についての参照先」
- 328 ページの「ファイルシステムの概要」
- 328 ページの「ファイルシステムのタイプ」
- 336 ページの「UFS ファイルシステム管理用のコマンド」
- 338 ページの「デフォルトの Solaris ファイルシステム」
- 335 ページの「スワップ空間」
- 339 ページの「UFS ファイルシステム」
- 346 ページの「ファイルシステムのマウントとマウント解除」
- 352 ページの「ファイルシステムのタイプを調べる」

ファイルシステムの新機能

この節では、Solaris リリースにおけるファイルシステムの新機能について説明します。

- 320 ページの「ファイルシステム監視ツール (`fsstat`) 」
- 320 ページの「Oracle Solaris ZFS ファイルシステム」
- 321 ページの「UFS ファイルシステムユーティリティー (`fsck`、`mkfs`、および `newfs`) の拡張機能」

Solaris の新機能の一覧および Solaris リリースについての説明は、『Oracle Solaris 10 9/10 の新機能』を参照してください。

ファイルシステム監視ツール (fsstat)

Solaris 10 6/06: 新しいファイルシステム監視ツール `fsstat` を使用してファイルシステムの操作を報告できます。いくつかのオプションを使用して、マウントポイントごとまたはファイルシステムのタイプごとなどの活動を報告できます。

たとえば、次の `fsstat` コマンドは、ZFS モジュールがロードされてからのすべての ZFS ファイルシステムの操作を表示します。

```
$ fsstat zfs
 new  name   name  attr  attr lookup rddir  read read  write write
 file remov  chng   get   set    ops   ops   ops bytes   ops bytes
 268K  145K 93.6K 28.0M 71.1K   186M 2.74M 12.9M 56.2G 1.61M 9.46G zfs
```

たとえば、次の `fsstat` コマンドは、`/export/ws` ファイルシステムがマウントされてからのすべてのファイルシステムの操作を表示します。

```
$ fsstat /export/ws
new  name   name  attr  attr lookup rddir  read read  write write
file remov  chng   get   set    ops   ops   ops bytes   ops bytes
   0     0     0 18.1K     0  12.6M    52     0     0     0     0 /export/ws
```

デフォルトのフォームは、G バイト、K バイト、M バイトなど、理解しやすい値を使用して統計情報を報告します。

詳細は、`fsstat(1M)` のマニュアルページを参照してください。

Oracle Solaris ZFS ファイルシステム

Solaris 10 6/06: 革新的な新しいファイルシステムである Oracle Solaris ZFS は、管理の簡素化、トランザクションのセマンティクス、エンドツーエンドのデータの整合性、および大きなスケーラビリティーを提供します。さらに、ZFS は次の管理機能を提供します。

- バックアップと復元の機能
- デバイス管理のサポート
- GUI 管理ツール
- 持続的なスナップショットの作成と複製の機能
- ファイルシステムの割り当て制限の設定機能
- RBAC ベースのアクセス制御機能
- ストレージプールによるファイルシステムの容量の予約機能
- ゾーンがインストールされた Solaris システムのサポート

ZFS の使用方法の詳細は、『Oracle Solaris ZFS 管理ガイド』を参照してください。

UFS ファイルシステムユーティリティー (fsck、mkfs、および newfs) の拡張機能

Solaris 10 6/06: ファイルシステム検査ユーティリティー fsck は、FreeBSD 4.9 バージョンの fsck プログラム内の機能やその他の拡張機能が含まれるように拡張されています。

Solaris リリースの fsck ユーティリティーには次の改善が加えられています。

- ファイルシステムをより詳しく検査して修復し、改善されたエラーメッセージを表示します。たとえばいくつかのシナリオにおいて、fsck は失われている構造を特定し、それらを適切に置き換えます。
- バックアップスーパーブロックを自動的に検索します。
- fsck を再実行する必要があるタイミングを報告します。
- ディレクトリが消去された場合は、fsck が直ちにディレクトリの内容の復元を試みるので、このユーティリティーの再実行時間が短縮されます。
- fsck が重複するブロックを検出し、重複するブロックを参照するすべてのファイルがクリアーされなかった場合、fsck は、fsck の実行の最後に i ノード番号を報告します。そのあとで、find コマンドを使用して破損した i ノードを確認できます。
- デバイスファイルや ACL エントリなどの拡張属性およびほかの特別なファイルのステータスに関連する改善されたエラーメッセージが含まれています。
- より詳細なメッセージを使用できるようにする -v オプションが用意されています。

さらに、newfs コマンドと mkfs コマンドが更新され、ファイルシステムのスーパーブロック情報をテキストで表示したり、スーパーブロック情報をバイナリ形式でダンプしたりするための新しいオプションが追加されました。

`newfs [ -S or -B ]` */dev/rdsk/...*

`-S` ファイルシステムのスーパーブロックをテキスト形式で表示します。

`-B` ファイルシステムのスーパーブロックをバイナリ形式でダンプします。

`mkfs [ -o calcsb or -o calcbinsb ]` */dev/rdsk/... size*

`-o calcsb` ファイルシステムのスーパーブロックをテキスト形式で表示します。

`-o calcbinsb` ファイルシステムのスーパーブロックをバイナリ形式でダンプします。

fsck ユーティリティーは、このスーパーブロック情報を使用して、バックアップスーパーブロックを検索します。

次の節では、具体的な fsck の機能拡張および対応するエラーメッセージについて説明します。fsck ユーティリティーを使用して破損したスーパーブロックを修復する手順については、446 ページの「不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース)」を参照してください。

バックアップスーパーブロックの自動検索

次の fsck エラーメッセージの例は、バックアップスーパーブロックの自動発見の失敗を示しています。

注意 – ファイルシステムに破損したスーパーブロックが含まれ、ntrack や nsect などの newfs または mkfs のカスタマイズされたパラメータを使用してファイルシステムが作成されている場合、修復処理のために自動的に発見されたスーパーブロックがファイルシステムを破損させる可能性があります。

カスタマイズされたパラメータを使用して作成されたファイルシステムで、ファイルシステムに不正なスーパーブロックが含まれている場合、fsck は、fsck セッションを取り消すための次のようなプロンプトを表示します。

```
CANCEL FILESYSTEM CHECK?
```

- ファイルシステムが newfs コマンドを使用して作成され、fsck がプライマリスーパーブロックのみが破損していることを報告する場合、fsck でスーパーブロックを復元することを検討してください。

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS

USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING NEWFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
```

```
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y


UPDATE STANDARD SUPERBLOCK? y

81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
```

- ファイルシステムが mkfs コマンドを使用して作成され、fsck がプライマリスーパーブロックのみが破損していることを報告する場合、fsck でスーパーブロックを復元することを検討してください。

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS

USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y


UPDATE STANDARD SUPERBLOCK? y

81 files, 3609 used, 243605 free (117 frags, 30436 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
```

次の例は、破損したスーパーブロックのシナリオで fsck の -y オプションを指定した場合に何が起こるかを示しています。fsck セッションが自動的に終了します。代替のスーパーブロックを使用してセッションを再実行するためのメッセージが表示されます。

```
# fsck -y /dev/dsk/c1t2d0s0
#
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS?  yes


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS?  yes

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS?  yes

CALCULATED GENERIC SUPERBLOCK WITH MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK?  yes

Please verify that the indicated block contains a proper
superblock for the filesystem (see fsdb(1M)).

FSCK was running in YES mode.  If you wish to run in that mode using
the alternate superblock, run 'fsck -y -o b=453920 /dev/rdsk/c1t2d0s0'.
```

次の fsck エラーメッセージのシナリオは、バックアップスーパーブロックの新しいプロンプトを示していますが、この例では、fsck の実行は取り消されていません。ファイルシステムがカスタマイズされたパラメータを使用して作成されている場合またはこのシステム上での fsck の実行に関する他の心配がある場合は、fsck セッションの取り消しが適切な応答です。

次のように、さまざまなスーパーブロックのエラー状況が斜体で表示されます。

```
# fsck /dev/rdsk/c0t1d0s0

** /dev/rdsk/c0t1d0s0

BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DATA BLOCKS OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: INODES PER GROUP OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG
BAD SUPERBLOCK AT BLOCK 16: BAD VALUES IN SUPER BLOCK
BAD SUPERBLOCK AT BLOCK 16: NCG OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: CPG OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: NCYL IS INCONSISTENT WITH NCG*CPG
BAD SUPERBLOCK AT BLOCK 16: SIZE OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DIRECTORIES OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
BAD SUPERBLOCK AT BLOCK 16: INOPB NONSENSICAL RELATIVE TO BSIZE

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS
```

```
USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2a - Check Duplicated Names
** Phase 2b - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups

SALVAGE? yes


UPDATE STANDARD SUPERBLOCK? yes

82 files, 3649 used, 244894 free (6 frags, 30611 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
```

fsck による再実行が必要なタイミングの報告

fsck によって再実行が必要なタイミングの報告が改善されるので、特に実行に時間がかかる大きなファイルシステムの場合などに、このコマンドを何度も実行するための時間が節約され、実行する必要性も少なくなります。

次の新しいメッセージは、エラーシナリオの最後に fsck ユーティリティーを再実行するように要求します。

```
***** PLEASE RERUN FSCK *****
```

または

```
Please rerun fsck(1M) to correct this.
```

これらの新しいプロンプトは、 fsck を再実行すべきかどうかの判断が難しいという以前の問題を解決します。

上のメッセージのように fsck の再実行を要求されない場合は、次のようなメッセージが表示されても fsck を再実行する必要はありません。

```
***** FILE SYSTEM WAS MODIFIED *****
```

ただし、このメッセージのあとに fsck を再実行してもファイルシステムは損傷しません。このメッセージは、fsck の修正処置に関する情報を示すだけのものです。

拡張属性に関連する新しい fsck メッセージ

拡張属性を含むファイルを報告および修復する新しい fsck メッセージが用意されています。次に例を示します。

```
BAD ATTRIBUTE REFERENCE TO I=1 FROM I=96

Attribute directory I=97 not attached to file I=96
  I=96  OWNER=root MODE=40755
SIZE=512 MTIME=Jul 12 10:55:10 2010
DIR= <xattr>

FIX?  yes

ZERO LENGTH ATTR DIR I=12  OWNER=root MODE=160755
SIZE=0 MTIME=Jul 12 10:56:10 2010
CLEAR?  yes

File should BE marked as extended attribute
  I=22  OWNER=root MODE=100644
SIZE=0 MTIME=Jul 12 10:57:10 2010
FILE=  <xattr>

FIX?  yes

UNREF ATTR DIR  I=106  OWNER=root MODE=160755
SIZE=512 MTIME=Jul 12 10:58:10 2010
RECONNECT?  yes

File I=107 should NOT be marked as extended attribute
  I=107  OWNER=root MODE=100644
SIZE=0 MTIME=Jul 12 10:59:10 2010
FILE=?/attfsdir-7-att

FIX?  yes
DIR I=106 CONNECTED.
```

重複するブロックまたはフラグメントの処理の改善

fsck エラーメッセージは、ブロック、フラグメント、または LFN に関する情報を報告するようになりました。LFN は、ファイルの先頭からの論理フラグメント番号です。たとえば、次のような出力が表示されます。

```
** Phase 1 - Check Blocks and Sizes
FRAGMENT 784 DUP I=38 LFN 0
FRAGMENT 785 DUP I=38 LFN 1
FRAGMENT 786 DUP I=38 LFN 2
.
```

.
.

fsck はオブジェクトをフラグメントとして処理しますが、以前の Solaris リリースでは、オブジェクト情報をブロックとして報告するだけでした。現在は、フラグメントとして正しく報告します。

fsck が重複するブロックまたはフラグメントに関連するエラーを検出した場合、fsck は、fsck の出力の最後にクリアされないファイルを表示します。たとえば、次のような出力が表示されます。

```
LIST REMAINING DUPS?  yes

Some blocks that were found to be in multiple files are still
assigned to file(s).
Fragments sorted by inode and logical offsets:
Inode 38:
  Logical Offset  0x00000000              Physical Fragment  784
  Logical Offset  0x00000800              Physical Fragment  786
  Logical Offset  0x00001000              Physical Fragment  788
  Logical Offset  0x00001800              Physical Fragment  790
```

この例では、そのあとで、find -i *inode-number* コマンドを使用して i ノード 38 を識別します。

ファイルシステム管理作業についての参照先

ファイルシステム管理の手順については、次を参照してください。

ファイルシステム管理作業	参照先
新しいファイルシステムを作成します。	第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」および第 19 章「CacheFS ファイルシステムの使用 (手順)」
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します。	第 18 章「ファイルシステムのマウントとマウント解除 (手順)」
新しいディスクデバイスを接続して構成します。	第 10 章「ディスクの管理 (概要)」
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します。	第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」
ファイルシステムの矛盾を検査して訂正します。	第 21 章「UFS ファイルシステムの整合性検査 (手順)」

ファイルシステムの概要

ファイルシステムは、ファイルを編成して格納するためのディレクトリ構造です。「ファイルシステム」という用語は、次のような場合に使用されます。

- 特定の種類のファイルシステム (ディスクベース、ネットワークベース、または仮想) を指す場合
- ルートディレクトリ (/) から始まるファイルツリー全体を指す場合
- ディスクスライスやほかの記憶メディアデバイスのデータ構造を指す場合
- ファイルツリー構造のうち、ファイルがアクセスできるように主なファイルツリー上のマウントポイントに接続されている部分を指す場合

通常、その意味は状況に応じて判断できます。

Solaris オペレーティングシステムは、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) アーキテクチャーを使用します。VFS アーキテクチャーによって、カーネルはファイルの読み取り、書き込み、一覧表示などの基本操作を処理できます。また、VFS アーキテクチャーにより新しいファイルシステムの追加も簡単になります。

ファイルシステムのタイプ

Oracle Solaris OS では、次の 3 種類のファイルシステムがサポートされます。

- ディスクベースのファイルシステム
- ネットワークベースのファイルシステム
- 仮想ファイルシステム

ファイルシステムのタイプを確認するには、352 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、DVD、フロッピーディスクなどの物理メディアに格納されます。ディスクベースのファイルシステムは、さまざまな形式で作成できます。次の表で、作成できる形式について説明します。

ディスクベースのファイルシステム	形式の説明
UFS	UNIX ファイルシステム (4.3 Tahoe リリースに組み込まれていた BSD Fat Fast ファイルシステム)。UFS は、Oracle Solaris OS のデフォルトのディスクベースファイルシステムです。 UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。ディスクのフォーマットとスライスへの分割方法については、第 10 章「ディスクの管理 (概要)」を参照してください。
ZFS	ZFS ファイルシステムは、Solaris 10 6/06 リリースで使用できます。詳細については、『Oracle Solaris ZFS 管理ガイド』を参照してください。
HSFS	High Sierra、Rock Ridge、および ISO 9660 のファイルシステム。High Sierra は、はじめての 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 の Rock Ridge 拡張がサポートされます。これらの拡張が CD-ROM 上にあるときは、書き込みとハードリンクを除いて、UFS ファイルシステムのすべての機能とファイルタイプが提供されます。
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向けに作成された DOS フォーマットディスク上のデータおよびプログラムに対して、読み取りアクセスと書き込みアクセスを行うことができます。
UDFS	UDFS (Universal Disk Format) ファイルシステム。DVD (Digital Versatile Disc または Digital Video Disc) と呼ばれる光学式メディアテクノロジに情報を格納するための業界標準形式です。

ディスクベースの各種ファイルシステムは、次のように特定のメディアのタイプに対応しています。

- UFS とハードディスク
- HSFS と CD-ROM
- PCFS とフロッピーディスク
- UDF と DVD

ただし、これらの対応関係は制限的なものではありません。たとえば、CD-ROM やフロッピーディスクにも、UFS ファイルシステムを格納できます。

UDFS (Universal Disk Format) ファイルシステム

リムーバブルメディア上で UDFS ファイルシステムを作成する方法については、38 ページの「リムーバブルメディア上にファイルシステムを作成する方法」を参照してください。

UDF ファイルシステムは、「DVD」 (Digital Versatile Disc または Digital Video Disc) 光学式メディアに情報を格納するための業界標準形式です。

UDF ファイルシステムは、SPARC と x86 の両方のプラットフォームにおいて、動的に読み込み可能な 32 ビットと 64 ビットのモジュールとして提供されます。また、ファイルシステムを作成、マウント、および検査するシステム管理ユーティリティーも同時に提供されます。Solaris の UDF ファイルシステムは、サポートされている ATAPI と SCSI の DVD ドライブ、CD-ROM デバイス、ハードディスク、およびフロッピーディスクドライブで機能します。さらに、Solaris の UDF ファイルシステムは UDF 1.50 仕様に完全に準拠しています。

UDF ファイルシステムには次のような機能があります。

- UDF ファイルシステムが入っている業界標準の CD-ROM や DVD-ROM のメディアにアクセスできます
- さまざまなプラットフォームやオペレーティングシステムで情報を交換できます
- UDF 形式に基づく DVD ビデオ仕様を使用することで、放送品質並みの映像、高品質のサウンド、すぐれた対話性という特長を備えた新しいアプリケーションを実装できます

次の機能は、UDF ファイルシステムにはありません。

- 書き込み可能なメディア (CD-RW) へのディスクアットワンス (Disk At Once) 記録、およびインクリメンタル記録
- UDF 1.50 仕様の一部ではないディスク割り当て、ACL、トランザクションのロギング、ファイルシステムのロック、およびファイルシステムのスレッドなど UFS 構成要素

次に、UDF ファイルシステムの要件を示します。

- Solaris 7 11/99 以降のリリースが動作していること
- SPARC または x86 プラットフォームがサポートされていること
- CD-ROM または DVD-ROM ドライブがサポートされていること

Solaris で実装された UDF ファイルシステムには、次のような互換性があります。

- 業界標準の読み取り / 書き込み UDF Version 1.50 のサポート
- 完全に国際化されたファイルシステムのユーティリティー

ネットワークベースのファイルシステム

「ネットワークベースのファイルシステム」は、ネットワークからアクセスされるファイルシステムです。ネットワークベースのファイルシステムは通常、1 つのシステム (通常はサーバー) 上にあり、ほか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 で分散された「リソース」(ファイルやディレクトリ) を管理するには、サーバーからそれらのリソースをエクスポートして個々のクライアントシステムでマウントします。詳細は、350 ページの「NFS 環境」を参照してください。

仮想ファイルシステム

「仮想ファイルシステム」は、特殊なカーネル情報と機能へのアクセスを提供するメモリーベースのファイルシステムです。ほとんどの仮想ファイルシステムは、ディスク領域を使用しません。ただし、CacheFS ファイルシステムは、ディスク上のファイルシステムを使用してキャッシュを保持します。また、一時ファイルシステム (TMPFS) などの一部の仮想化ファイルシステムは、ディスク上のスワップ空間を使用します。

CacheFS ファイルシステム

CacheFS ファイルシステムを使用すると、リモートファイルシステムや、CD-ROM ドライブのような低速デバイスのパフォーマンスを改善できます。ファイルシステムをキャッシュすると、リモートファイルシステムや CD-ROM から読み込まれたデータは、ローカルシステム上のキャッシュに格納されます。

NFS や CD-ROM ファイルシステムのパフォーマンスとスケーラビリティーを向上させるには、CacheFS ファイルシステムを使用してください。CacheFS ソフトウェアは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティーを改善する汎用ファイルシステムキャッシュメカニズムです。

CacheFS ソフトウェアは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ソフトウェアはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。また、CacheFS ファイルシステムと autofs サービスを組み合わせると、パフォーマンスとスケーラビリティーをさらに向上させることができます。

CacheFS ファイルシステムの詳細は、第 19 章「CacheFS ファイルシステムの使用 (手順)」を参照してください。

NFS Version 4 と CacheFS の互換性の問題

CacheFS クライアントと CacheFS サーバーの両方で NFS version 4 が実行されている場合、ファイルはフロントファイルシステムにキャッシュされません。すべてのファイルアクセスは、バックファイルシステムから提供されます。また、ファイルはフロントファイルシステムにキャッシュされていないため、フロントファイルシステムに反映する CacheFS 固有のマウントオプションは無視されます。CacheFS 固有のマウントオプションはバックファイルシステムに適用しません。

注 – 初めてシステムを NFS version 4 に構成すると、キャッシュが動作しないことを示す警告がコンソールに表示されます。

以前の Solaris リリースのように CacheFS マウントを実装する場合は、CacheFS `mount` コマンドで NFS version 3 を指定してください。次に例を示します。

```
mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,vers=3
starbug:/docs /docs
```

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。通常、ファイルシステムの読み取りと書き込みには、UFS ファイルシステムよりもメモリーを使用したほうが効率的です。TMPFS を使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバヘッドを軽減でき、システムパフォーマンスを改善できます。一時ファイルは、たとえば、プログラムのコンパイル時に作成されます。OS は、一時ファイルを操作しているとき、多くのディスク処理またはネットワーク処理を行います。TMPFS を使ってこれらの一時ファイルを管理することで、それらの作成、操作、および削除の効率を大幅に向上できます。

TMPFS ファイルシステムのファイルは、永続的に保存されるわけではありません。これらのファイルは、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削除されます。

TMPFS は、Oracle Solaris OS 内の `/tmp` ディレクトリのデフォルトのファイルシステムです。UFS ファイルシステムの場合と同様に、`/tmp` ディレクトリとの間でファイルをコピーまたは移動できます。

TMPFS ファイルシステムは、一時的な退避場所としてスワップ空間を使用します。TMPFS ファイルシステムがマウントされたシステムのスワップ空間が足りないと、次の 2 つの問題が発生する可能性があります。

- TMPFS ファイルシステムは、通常のファイルシステムと同様に容量不足になる可能性があります。
- TMPFS はスワップ空間を割り当ててファイルのデータを保存するので (必要な場合)、一部のプログラムがスワップ空間不足のために実行できなくなる可能性があります。

TMPFS ファイルシステムの作成方法については、第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」を参照してください。スワップ空間を追加する方法については、第 20 章「追加スワップ空間の構成 (手順)」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、 /tmp/newroot 上にルート (/) ディレクトリのループバックマウントを作成できます。このループバックマウントでは、NFS サーバーからマウントされたファイルシステムを含むファイルシステム階層全体が、 /tmp/newroot の下に複製されたように見えます。どのファイルにも、ルート (/) で始まるパス名または /tmp/newroot で始まるパス名を使用してアクセスできます。

LOFS ファイルシステムを作成する手順については、第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」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) はメモリー内に存在し、/proc ディレクトリ内にアクティブなプロセスのプロセス番号別リストが格納されます。/proc ディレクトリの内容は、ps などのコマンドで使用されます。デバッガやほかの開発ツールも、ファイルシステムコールを使用して、プロセスのアドレス空間にアクセスできます。

注意 – /proc ディレクトリ内のファイルは削除しないでください。/proc ディレクトリからプロセスを削除しても、そのプロセスは強制終了されません。/proc ファイルはディスク容量を消費しないため、このディレクトリからファイルを削除しても無意味です。

/proc ディレクトリは、管理が不要です。

その他の仮想ファイルシステム

次のタイプの仮想ファイルシステムは、参考のために掲載してあります。管理は不要です。

仮想ファイルシステム	説明
CTFS	CTFS (契約ファイルシステム) は、契約の作成、制御、および監視のためのインタフェースです。契約は、豊富なエラー報告機能とリソースの削除を延期する手段 (オプション) を提供することにより、プロセスと、このプロセスが依存するシステムとの関係を拡張します。 サービス管理機能 (SMF) は、プロセス契約 (契約の一種) を使用して、サービスを構成するプロセス群を追跡します。このため、マルチプロセスサービスの一部分での障害をそのサービスの障害として識別できます。
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、ファイル記述子を使用して名前を明示的に与えます。
MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供します。
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 に使用されます。
OBJFS	OBJFS (オブジェクト) ファイルシステムは、現在カーネルによってロードされているすべてのモジュールの状態を説明します。デバッガはこのファイルシステムにアクセスすることで、カーネルに直接アクセスしなくてもカーネルシンボルに関する情報を入手できます。
SHAREFS	ローカルシステムの共有ファイルシステムのテーブルに対する読み取り専用アクセス権を提供します。
SPECFS (特殊)	キャラクタ型特殊デバイスとブロック型デバイスへのアクセスを提供します。
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。

libc_hwcap

x86 システムでのマウント出力には、`libc` のハードウェア最適化の実装である、`libc_hwcap` ライブラリのループバックマウントが含まれる場合があります。この `libc` の実装は、32 ビットアプリケーションのパフォーマンスを最適化するためのものです。

このループバックマウントは、管理の必要がなく、ディスク容量を消費しません。

拡張ファイル属性

UFS、NFS、およびTMPFSファイルシステムは、拡張ファイル属性を含むように機能拡張されました。アプリケーション開発者は、拡張ファイル属性を使って、ファイルに特定の属性を関連付けることができます。たとえば、ウィンドウシステムの管理アプリケーションの開発者は、表示アイコンとファイルを関連付けることができます。拡張ファイル属性は、論理的には、ターゲットファイルに関連付けられている隠しディレクトリ内のファイルとして表されます。

属性を追加し、拡張属性の名前空間内に入っているシェルコマンドを実行するには、`runat` コマンドを使用します。拡張属性の名前空間とは、特定のファイルに関連付けられた、非表示の属性ディレクトリです。

`runat` コマンドを使用して属性をファイルに追加するには、最初に属性ファイルを作成する必要があります。

```
$ runat filea cp /tmp/attrdata attr.1
```

次に、`runat` コマンドを使用して、ファイルの属性をリストに表示します。

```
$ runat filea ls -l
```

詳細は、`runat(1)` のマニュアルページを参照してください。

属性認識オプションの追加により、多くのSolarisファイルシステムコマンドがファイルシステム属性をサポートするようになりました。属性認識オプションを使って、ファイル属性を照会したり、コピーしたり、検索したりできます。詳細は、各ファイルシステムコマンドのマニュアルページを参照してください。

スワップ空間

Oracle Solaris OS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。スワップ空間は、現在のプロセスを処理するだけの十分な物理メモリーがシステムにない場合に、仮想メモリー記憶域として使用されます。

多くのアプリケーションは十分なスワップ空間が使用できることを前提に作成されているため、スワップ空間を割り当て、その使われ方を監視して、必要に応じてスワップ空間を追加する方法を知っておく必要があります。スワップ空間の概要とスワップ空間の追加方法については、第20章「追加スワップ空間の構成(手順)」を参照してください。

UFS ファイルシステム管理用のコマンド

ほとんどのファイルシステム管理コマンドには、汎用コンポーネントとファイルシステム固有のコンポーネントがあります。可能な場合には、常に汎用コマンドを使用してください。汎用コマンドは、ファイルシステム固有のコマンドを呼び出します。次の表に、ファイルシステム管理の汎用コマンドを一覧表示します。これらのコマンドは、/usr/sbin ディレクトリにあります。

表 16–1 ファイルシステム管理用の汎用コマンド

コマンド	説明	マニュアルページ
clri	i ノードをクリアします	clri(1M)
df	空きディスクブロック数とファイル数を表示します。	df(1M)
ff	ファイルシステムのファイル名と統計情報を表示します	ff(1M)
fsck	ファイルシステムの整合性を検査し、検出された損傷を修復します	fsck(1m)
fsdb	ファイルシステムをデバッグします	fsdb(1M)
fstyp	ファイルシステムのタイプを調べます	fstyp(1M)
labelit	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します (volcopy コマンド専用)	labelit(1M)
mkfs	新しいファイルシステムを作成します	mkfs(1M)
mount	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します	mount(1M)
mountall	virtual file system table (/etc/vfstab) に指定されているすべてのファイルシステムをマウントします	mountall(1M)
ncheck	パス名とその i ノード番号のリストを生成します	ncheck(1M)
umount	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します	mount(1M)
umountall	virtual file system table (/etc/vfstab) に指定されているすべてのファイルシステムをマウント解除します	mountall(1M)
volcopy	ファイルシステムのイメージコピーを作成します	volcopy(1M)

ファイルシステムコマンドによるファイルシステムタイプの判断

汎用ファイルシステムコマンドは、次の順序でファイルシステムのタイプを判断します。

1. 指定されている場合は、`-F` オプションで指定されているファイルシステムのタイプから判断します。
2. 特殊デバイスを `/etc/vfstab` ファイルのエントリと突き合わせて判断します (*special* デバイス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、*special* デバイスフィールドと突き合わせて検査します。
3. ローカルファイルシステムの場合は `/etc/default/fs` ファイル内に指定されたデフォルトを使用し、リモートファイルシステムの場合は`/etc/dfs/fstypes` ファイル内に指定されたデフォルトを使用して判断します。

汎用ファイルシステムコマンドと専用ファイルシステム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、『SunOS リファレンスマニュアル 1M : システム管理コマンド』を参照してください。 ファイルシステムの汎用コマンドのマニュアルページには、汎用コマンドオプションに関する情報だけが記載されています。専用コマンドのマニュアルページには、該当するファイルシステムのオプション情報が記載されています。特定のファイルシステムのマニュアルページを見つけるには、汎用コマンド名の末尾にアンダースコアとファイルシステムタイプの略称を追加してください。

たとえば、UFS ファイルシステムのマウントについてのマニュアルページを参照するには、次のように入力します。

```
$ man mount_ufs
```

たとえば、ZFS ファイルシステムのマウントについてのマニュアルページを参照するには、次のように入力します。

```
$ man zfs
```

デフォルトの Solaris ファイルシステム

ZFS や UFS ファイルシステムは階層構造になっており、ルートディレクトリ (/) から始まり、下位に多数のディレクトリが形成されています。Solaris のインストールプロセスは、デフォルトのディレクトリセットをインストールし、一連の規則を適用して類似するタイプのファイルをグループ化します。

Solaris のファイルシステムおよびディレクトリの内容については、`filesystem(5)` のマニュアルページを参照してください。

次の表で、デフォルトの Solaris ファイルシステムの概要について説明します。

表 16–2 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 または ZFS	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ます。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ます。
`/usr`	UFS または ZFS	ほかのユーザーと共有できるシステムファイルとディレクトリ。特定のタイプのシステム上でのみ実行できるファイルは、`/usr` ファイルシステムまたはディレクトリに入っています (SPARC 実行可能ファイルなど)。どのタイプのシステム上でも使用できるファイル (マニュアルページなど) は、`/usr/share` ディレクトリに配置できます。
`/export/home` または `/home`	NFS、UFS、または ZFS	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ます。デフォルトでは、`/home` ディレクトリは自動マウントされるファイルシステムです。スタンドアロンシステムでは、`/home` ディレクトリが UFS ファイルシステムまたは ZFS ファイルシステムである場合があります。
`/var`	UFS または ZFS	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` や `ex` のバックアップファイルなど) が含まれます。
`/opt`	NFS、UFS、または ZFS	オプションの他社製のソフトウェア製品のマウントポイント。一部のシステムでは、`/opt` ディレクトリが UFS ファイルシステムまたは ZFS ファイルシステムである場合があります。

表 16–2 デフォルトの Solaris ファイルシステム (続き)

ファイルシステムまたはディレクトリ	ファイルシステムのタイプ	説明
/tmp	TMPFS	システムがブートされるか、/tmp ファイルシステムがマウント解除されるたびに削除される一時ファイル。
/proc	PROCFS	アクティブなプロセスのプロセス番号別リスト。
/etc/mnttab	MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供する仮想ファイルシステム。
/var/run	TMPFS	システムのブート後は不要になる一時ファイルを格納するメモリーベースのファイルシステム。
/system/contract	CTFS	契約情報を保持する仮想ファイルシステム。
/system/object	OBJFS	カーネルに直接アクセスすることなくカーネルシンボルの情報にアクセスする際にデバッガによって使用される仮想ファイルシステム。

システムを動作させるには、ルート (/) と /usr のファイルシステムが必要です。mount など、/usr ファイルシステム内のもっとも基本的なコマンドの一部は、ルート (/) ファイルシステムにも含まれています。したがって、これらのコマンドは、システムのブート時やシングルユーザーモードでの使用時のほか、/usr がマウントされていない場合でも使用可能です。ルート (/) および /usr ファイルシステムのデフォルトディレクトリについては、第 22 章「UFS ファイルシステム (参照情報)」を参照してください。

UFS ファイルシステム

UFS ファイルシステムの詳細は、次の節を参照してください。

- 339 ページの「UFS ファイルシステムの機能」
- 340 ページの「UFS ファイルシステムの計画」
- 341 ページの「64 ビット: マルチテラバイトの UFS ファイルシステムのサポート」
- 344 ページの「UFS ロギング」
- 345 ページの「UFS スナップショット」
- 346 ページの「UFS 直接入出力」

UFS ファイルシステムの機能

UFS は、Solaris オペレーティングシステム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 ファイルシステムを管理していることになります。UFS が提供する機能は次のとおりです。

UFS の機能	説明
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を提供します。
大規模ファイルシステム	最大 16T バイトまで増やせるファイルシステムに、およそ 1T バイトの大きさのファイルを指定できます。EFI ディスクラベルを使ってディスク上にマルチテラバイトの UFS ファイルシステムを作成できます。
ログ	UFS ロギングは、完全な UFS 操作による複数のメタデータ変更を、1 つのトランザクションにまとめます。一連のトランザクションは、ディスク上のログに記録されたあと、実際の UFS ファイルシステムのメタデータに適用されます。
マルチテラバイトファイルシステム	マルチテラバイトファイルシステムでは、およそ 16T バイトの最大使用可能空間からおよそ 1 パーセントのオーバーヘッドを引いたサイズの UFS ファイルシステムを作成できます。
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、または不明として表示します。これらのフラグによって、不要なファイルシステム検査を行うことがなくなります。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検査は実行されません。

UFS ファイルシステム構造の詳細は、第 22 章「UFS ファイルシステム (参照情報)」を参照してください。

UFS ファイルシステムの計画

ファイルシステムの配置を決めるときには、要求が競合する可能性があることを考えなければなりません。次にいくつかの推奨事項を示します。

- 作業負荷を異なる入出力システムやディスクドライブ間でできるだけ均等に分散します。`/export/home` ファイルシステムやスワップ空間を異なるディスク間で均等に分散させます。
- プロジェクトの個々の部分やグループのメンバーを同じファイルシステム内に入れます。
- 1 ディスク当たりのファイルシステム数をできるだけ少なくします。通常、システム (またはブート) ディスク上には、ルート (`/`)、`/usr`、スワップ空間の 3 つのファイルシステムがあります。その他のディスク上で、1 つまたは多くても 2 つのファイルシステムを作成し、追加のスワップ空間になるファイルシステムを 1 つ持たせます。多数の小規模なファイルシステムに分割しすぎるよりもファイルシステム数を少なくして余地を設ける方が、ファイルがフラグメントに分割され

る可能性が小さくなります。容量の大きいテープドライブを使用し、ufsdump コマンドで複数のボリュームを処理できるようにしておけば、大規模なファイルシステムでも簡単にバックアップをとることができます。

- きわめて小さいファイルを絶えず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、類似するタイプのユーザーファイルを同じファイルシステム内で保管する必要はありません。

デフォルトのファイルシステムパラメータの情報と、新しい UFS ファイルシステムの作成手順については、第 17 章「ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)」を参照してください。

64 ビット: マルチテラバイトの UFS ファイルシステムのサポート

この Solaris リリースでは、64 ビットの Solaris カーネルを実行しているシステムで、マルチテラバイトの UFS ファイルシステムをサポートします。

以前は、64 ビットと 32 ビットのどちらのシステムでも、UFS ファイルシステムはおよそ 1T バイトに制限されていました。今回、マルチテラバイト UFS ファイルシステムをサポートするため、すべての UFS ファイルシステムコマンドとユーティリティーが更新されました。

たとえば、ufsdump コマンドは大規模な UFS ファイルシステムをダンプできるようにブロックサイズが大きくなりました。

```
# ufsdump 0f /dev/md/rdsk/d97 /dev/md/rdsk/d98
    DUMP: Date of this level 0 dump: Mon Jul 12 10:51:10 2010
    DUMP: Date of last level 0 dump: the epoch
    DUMP: Dumping /dev/md/rdsk/d97 to /dev/md/rdsk/d98
    DUMP: Mapping (Pass I) [regular files]
    DUMP: Mapping (Pass II) [directories]
    DUMP: Writing 32 Kilobyte records
    DUMP: Estimated 17439410 blocks (8515.34MB).
    DUMP: Dumping (Pass III) [directories]
    DUMP: Dumping (Pass IV) [regular files]
```

1T バイト未満の UFS ファイルシステムは、以前と同じように管理されます。1T バイト未満の UFS ファイルシステムと 1T バイトを超えるファイルシステムとの間に管理面での違いはありません。

最初に newfs -T オプションを使用していれば、1T バイト未満の UFS ファイルシステムを作成し、最終的にマルチテラバイトのファイルシステムに拡張されるように指定できます。このオプションを使用すると、マルチテラバイトのファイルシステムに適した値に拡張が可能であるように i ノードとフラグメントの密度が設定されます。

32 ビットのカーネルを実行しているシステムで 1T バイト未満の UFS ファイルシステムを作成するときに、newfs -T オプションを使用すると、最終的に 64 ビットのカーネルでこのシステムを起動するときに、growfs コマンドを使ってこのファイルシステムを拡張できます。詳細は、newfs(1M) のマニュアルページを参照してください。

fstyp -v コマンドを使用し、magic 列で次の値を確認することにより、UFS ファイルシステムがマルチテラバイトをサポートしているかどうかを識別できます。

```
# /usr/sbin/fstyp -v /dev/md/rdsk/d3 | head -5
ufs
magic   decade  format  dynamic time    Mon Jul 12 11:12:36 2010
```

マルチテラバイトをサポートしてない UFS ファイルシステムでは、次の fstyp の出力が表示されます。

```
# /usr/sbin/fstyp -v /dev/md/rdsk/d0 | head -5
ufs
magic   11954   format  dynamic time    Mon Jul 12 12:41:29 2010
```

growfs コマンドを使用すると、サービスやデータを失わずに、UFS ファイルシステムをスライスまたはボリュームのサイズまで拡張できます。詳細については、growfs(1M) のマニュアルページを参照してください。

これに関連して、EFI ディスクラベルによるマルチテラバイトボリュームのサポートと、Solaris ボリュームマネージャーによるマルチテラバイトボリュームのサポートという新しい 2 つの機能が追加されました。詳細は、198 ページの「EFI ディスクラベル」および『Solaris ボリュームマネージャの管理』を参照してください。

マルチテラバイトの UFS ファイルシステムの機能

マルチテラバイトの UFS ファイルシステムには、次の機能があります。

- 最大 16T バイトの UFS ファイルシステムを作成できます。
- 16T バイト未満のファイルシステムを作成し、あとで 16T バイトまでサイズを増やすことができます。
- マルチテラバイトのファイルシステムを物理ディスク、Solaris ボリュームマネージャーの論理ボリューム、および Veritas の VxVM 論理ボリューム上に作成できます。
- マルチテラバイトのファイルシステムでは、UFS ロギングが有効になっているとパフォーマンスが向上するという利点があります。また、マルチテラバイトのファイルシステムは、ロギングが有効なときは fsck コマンドを実行しなくてもよい場合があるというロギングの利点を利用することもできます。
- マルチテラバイトの UFS ファイルシステム用のパーティションを作成すると、そのディスクには自動的に EFI ディスクラベルが付きます。EFI ディスクラベルの詳細は、198 ページの「EFI ディスクラベル」を参照してください。

- マルチテラバイトファイルシステムのスナップショットを作成できます (ファイルシステムのサイズが 512G バイトを超える場合、複数のバッキングストアファイルが作成される)。

マルチテラバイトの UFS ファイルシステムの制限事項

マルチテラバイトの UFS ファイルシステムには、次の制限事項があります。

- 32 ビットシステムではサポートされません。
- 32 ビットの Solaris カーネルを実行しているシステムでは、1T バイトを超えるファイルシステムをマウントすることはできません。
- 1T バイトを超える個々のファイルはサポートされていません。
- UFS ファイルシステムの 1T バイトあたりの最大ファイル数は 100 万です。たとえば、4T バイトのファイルシステムには、400 万個のファイルを含めることができます。

  100 万に制限されているのは、`fsck` コマンドでファイルシステムを検査するのに要する時間を減らすためです。
- マルチテラバイトの UFS ファイルシステムに設定できる最大割り当て数は、2T バイトの 1024 バイトブロックです。

マルチテラバイトの UFS 作業についての参照先

マルチテラバイトの UFS ファイルシステムの操作手順については、次を参照してください。

マルチテラバイトの UFS 作業	参照先
マルチテラバイトの UFS ファイルシステムを作成します	359 ページの「マルチテラバイトの UFS ファイルシステムを作成する方法」 360 ページの「マルチテラバイトの UFS ファイルシステムを拡張する方法」 361 ページの「UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法」
マルチテラバイトの UFS スナップショットを作成します	例 25–2
マルチテラバイトの UFS に関する問題の障害追跡を行います	362 ページの「マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡」

UFS ロギング

UFS ロギングは、1 つの完全な UFS 操作を構成する複数のメタデータ変更を、1 つのトランザクションにまとめます。ディスク上の単一のログファイルに、複数のトランザクションが記録されます。これらのトランザクションは、その後、実際の UFS ファイルシステムのメタデータに適用されます。

システムはリブート時に、不完全なトランザクションを廃棄しますが、完結している操作のトランザクションは適用します。完結しているトランザクションだけが適用されるために、ファイルシステムの整合性が保たれます。この整合性は、システムがクラッシュした場合も保持されます。システムのクラッシュは、システムコールを中断し、UFS ファイルシステムの不整合の原因となる可能性があります。

UFS ロギングには 2 つの長所があります。

- トランザクションログによりファイルシステムの整合性がすでに保持されている場合、システムがクラッシュしたとき、またはクリーンでないシステム停止が発生したときに、`fsck` コマンドを実行する必要がないことがあります。クリーンでないシステム停止については、431 ページの「`fsck` コマンドで検査して修復される内容」を参照してください。
- UFS ロギングのパフォーマンスは、非ロギングファイルシステムのパフォーマンスレベルを上回るようになりました。ロギングが有効なファイルシステムでは、同じデータに対する複数の更新が 1 回の更新に変換されるため、これがパフォーマンスの向上につながっています。したがって、オーバーヘッドディスクの操作回数を減らす必要があります。

ロギングは、次の場合を除き、すべての UFS ファイルシステムでデフォルトで有効になります。

- ロギングが明示的に無効とされた場合。
- ログ用のファイルシステム容量が不足している場合。

以前の Solaris リリースでは、UFS ロギングは手動で有効にする必要がありました。

UFS ロギングを使用する際には、次の点に注意してください。

- ユーザー、アプリケーション、UFS ロギングなどの一般的なシステム要求に対応できるだけのディスク容量が確保されているかどうかを確認してください。
- ディスク容量が不足していてデータをロギングできない場合は、次のようなメッセージが表示されます。

```
# mount /dev/dsk/c0t4d0s0 /mnt
/mnt: No space left on device
Could not enable logging for /mnt on /dev/dsk/c0t4d0s0.
#
```

  ただし、その場合でも、ファイルシステムはマウントされます。次に例を示します。

```
# df -h /mnt
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t4d0s0      142M   142M     0K   100%    /mnt
#
```

- ロギングが有効にされた UFS ファイルシステムでは、ほとんど空の状態であっても、いくらかのディスク容量がログ用として消費されます。
- 以前の Solaris リリースからこの Solaris リリースにアップグレードする場合、`/etc/vfstab` ファイル内で `logging` オプションが指定されていなかった場合でも、UFS ファイルシステムのロギングは有効になります。ロギングを無効にするには、`/etc/vfstab` ファイル内の UFS ファイルシステムのエントリに、`nologging` オプションを追加します。

UFS トランザクションログには、次の特徴があります。

- ファイルシステム上の空きブロックから割り当てられる。
- 1G バイトのファイルシステム領域ごとに約 1M バイトのサイズ (合計で 256M バイトまで) になる。ファイルシステムに大量のシリンダグループが存在する場合、ログのサイズはこれより大きくなることがあります (最大で 512M バイトまで)。
- これ以上書き込めない状態になるとフラッシュされる。
- ファイルシステムがマウント解除されたときや、`lockfs` コマンドを実行したときにもフラッシュされる。

UFS ロギングを有効にする必要がある場合は、`-/etc/vfstab` ファイル内で `mount` コマンドに `o logging` オプションを指定するか、ファイルシステムを手動でマウントするときに mount コマンドに o logging オプションを指定します。ログは、ルート (/) ファイルシステムを含む、任意の UFS ファイルシステムで有効にできます。また、`fsdb` コマンドには、UFS ロギングをサポートするための新しいデバッグコマンドが用意されています。

一部のオペレーティングシステムでは、ロギングが有効になっているファイルシステムを「ジャーナル」ファイルシステムと呼びます。

UFS スナップショット

`fssnap` コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成できます。「スナップショット」は、バックアップ操作のためのファイルシステムの一時イメージです。

詳細は、第 25 章「UFS スナップショットの使用 (手順)」を参照してください。

UFS 直接入出力

直接入出力の目的は、大容量入出力処理のスピードを速くすることです。大容量入出力処理では、大規模ファイル (256K バイトを超える) を転送するために、大容量のバッファーサイズを使用します。

UFS の直接入出力を使用すると、データベースエンジンなど、独自の内部バッファリングを行うアプリケーションにメリットがあります。UFS 直接入出力を使用すると、raw デバイスがアクセスされたときに起こる同様の入出力の並行処理が可能になります。現在では、パフォーマンスがわずかに低下するだけで、ファイルシステムのネーミングや柔軟性がもたらすメリットを受けることができます。データベースの製造元を調べて、その製品構成オプションで UFS の直接入出力を有効にできるかどうかを確認してください。

`mount` コマンドに `forcedirectio` オプションを使用しても、直接入出力をファイルシステムで有効にできます。直接入出力を有効にしてパフォーマンスが向上するのは、ファイルシステムが大量の連続するデータを転送するときだけです。

forcedirectio オプションでファイルシステムをマウントするとき、データはユーザーのアドレス空間とディスクの間で直接伝送されます。直接入出力がファイルシステムで無効な場合、ユーザーのアドレス空間とディスクの間で転送されるデータは、まず、カーネルアドレス空間にバッファーされます。

デフォルトでは、UFS ファイルシステムでは直接入出力は行われません。詳細については、`mount_ufs(1M)` のマニュアルページを参照してください。

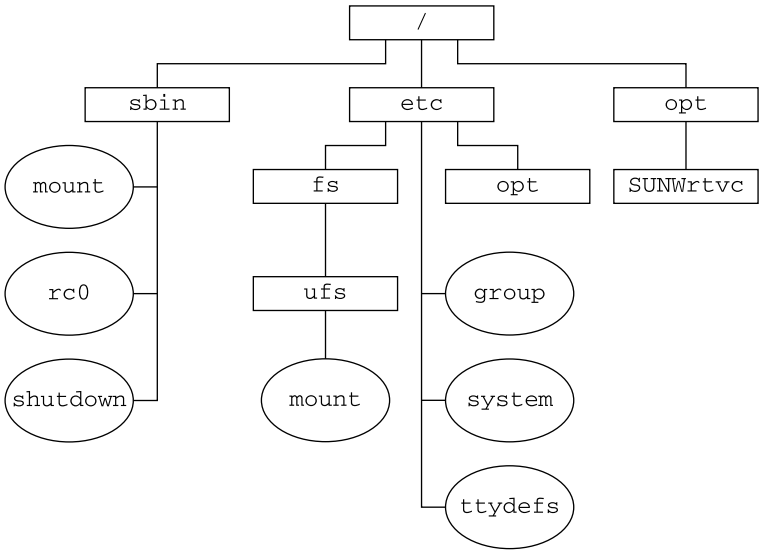
ファイルシステムのマウントとマウント解除

ファイルシステム上のファイルにアクセスするには、ファイルシステムをマウントする必要があります。ファイルシステムのマウントとは、ファイルシステムを特定のディレクトリ (マウントポイント) に接続し、システムで使用可能にすることです。ルート (/) ファイルシステムは、常にマウントされています。ほかのファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりできます。

ファイルシステムをマウントすると、そのファイルシステムがマウントされている間は、マウントポイントのディレクトリ内に実際に存在しているファイルやディレクトリは使用できなくなります。これらのファイルは、永続的にマウントプロセスの影響を受け続けるわけではありません。ファイルシステムのマウントが解除されると、再び使用可能な状態になります。ただし、通常は存在するがアクセスできないファイルは混乱の原因となるので、通常マウントディレクトリは空になっています。

たとえば、次の図は、ルート (/) ファイルシステムと、その下の `sbin`、`etc`、`opt` の各サブディレクトリから始まるローカルファイルシステムを示しています。

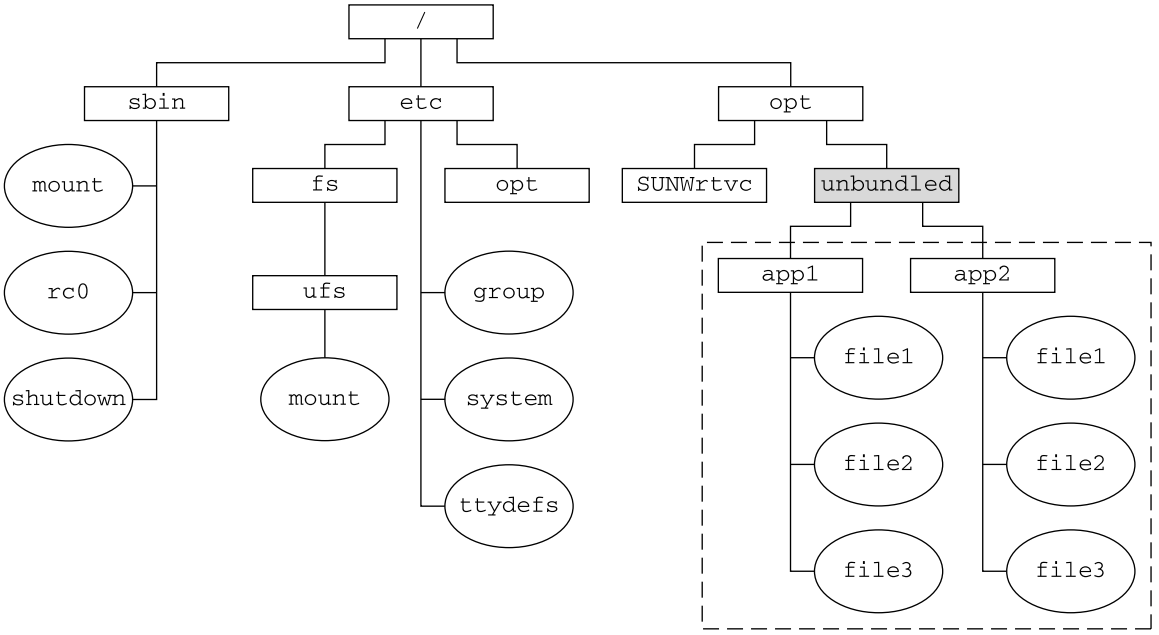
図 16–1　UFS ルート (/) ファイルシステムの例



別パッケージの製品が含まれている /opt ファイルシステムからローカルファイルシステムにアクセスするには、次の作業を行います。

- まず、ファイルシステムをマウントするためのマウントポイントとして使用するディレクトリを作成します (例: /opt/unbundled)。
- マウントポイントを作成したら、mount コマンドを使ってファイルシステムをマウントします。このコマンドを実行すると、/opt/unbundled 内のすべてのファイルおよびディレクトリが使用可能になります (次の図を参照)。

図 16–2 UFS ファイルシステムのマウント



マウントポイント

ファイルシステム

ファイルシステムのマウント手順については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

マウントされたファイルシステムテーブル

ファイルシステムをマウントまたはマウント解除すると、現在マウントされているファイルシステムのリストを使用して、/etc/mnttab (マウントテーブル) ファイルが変更されます。このファイルの内容を表示するには、cat または more コマンドを使用します。ただし、このファイルを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
rpool/ROOT/zfs509BE             /       zfs     dev=4010002     0
/devices        /devices        devfs   dev=5000000     1235087509
ctfs    /system/contract        ctfs    dev=5040001     1235087509
proc    /proc   proc    dev=5080000     1235087509
mnttab  /etc/mnttab     mntfs   dev=50c0001     1235087509
swap    /etc/svc/volatile       tmpfs   xattr,dev=5100001       1235087510
objfs   /system/object  objfs   dev=5140001     1235087510
```

```
sharefs /etc/dfs/sharetab       sharefs dev=5180001     1235087510
fd      /dev/fd fd      rw,dev=52c0001  1235087527
swap    /tmp    tmpfs   xattr,dev=5100002       1235087543
swap    /var/run        tmpfs   xattr,dev=5100003       1235087543
rpool/export    /export zfs     rw,devices,setuid,nonbmand,exec,xattr,...
rpool/export/home       /export/home    zfs     rw,devices,setuid,nonbmand,exec,...
rpool   /rpool  zfs     rw,devices,setuid,nonbmand,exec,xattr,atime,dev=4010005 1235087656
```

仮想ファイルシステムテーブル

必要に応じて毎回ファイルシステムを手動でマウントしていては、非常に効率が悪く、エラーも発生しやすくなります。これらの問題を解決するために、virtual file system table (`/etc/vfstab` ファイル) にファイルシステムのリストとそのマウント方法を指定できます。

`/etc/vfstab` ファイルは、2 つの重要な機能を持っています。

- 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ます。ZFS ファイルシステムは、ブート時に SMF サービスにより自動的にマウントされます。`vfstab` ファイル内のエントリは使用されません。
- マウントポイント名だけを使って、ファイルシステムをマウントできます。`/etc/vfstab` ファイルには、マウントポイントと実際のデバイススライス名のマッピング情報が含まれています。

システムをインストールすると、インストール時の選択内容に基づいて、デフォルトの `/etc/vfstab` ファイルが作成されます。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には、次のような情報を指定する必要があります。

- ファイルシステムが配置されるデバイス
- ファイルシステムのマウントポイント
- ファイルシステムのタイプ
- システムのブート時に、`mountall` コマンドを使ってファイルシステムを自動的にマウントするかどうか
- マウントオプション

次に、UFS ルートファイルシステムが動作するシステムの `/etc/vfstab` ファイルの例を示します。コメント行は `#` で始まります。この例は、2 つのディスク (`c0t0d0` と `c0t3d0`) を持つシステムの `/etc/vfstab` ファイルを示しています。

```
$ more /etc/vfstab
#device           device              mount           FS      fsck    mount   mount
#to mount         to fsck             point           type    pass    at boot options
#
fd                 -                  /dev/fd         fd      -       no      -
/proc              -                  /proc           proc    -       no      -
```

```
/dev/dsk/c0t0d0s1  -                   -                   swap    -       no      -
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0 /                    ufs     1       no      -
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6 /usr                 ufs     1       no      -
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7 /export/home         ufs     2       yes     -
/dev/dsk/c0t0d0s5  /dev/rdsk/c0t0d0s5 /opt                 ufs     2       yes     -
/devices           -                   /devices            devfs   -       no      -
sharefs            -                   /etc/dfs/sharetabsharefs -       no      -
ctfs               -                   /system/contract ctfs    -       no      -
objfs              -                   /system/object   objfs   -       no      -
swap               -                   /tmp                tmpfs   -       yes     -
```

この例では、ルート (/) や /usr の場合、mount at boot フィールドの値は no になります。これらのファイルシステムは、ブートシーケンスの一環として、mountall コマンドの実行前にカーネルによってマウントされます。

次に、ZFS ルートファイルシステムが動作するシステムの vfstab の例を示します。

```
# cat /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd              -               /dev/fd         fd      -       no      -
/proc           -               /proc           proc    -       no      -
/dev/zvol/dsk/rpool/swap -      -               swap    -       no      -
/devices        -              /devices         devfs   -       no      -
sharefs         -              /etc/dfs/sharetabsharefs -       no      -
ctfs            -              /system/contract ctfs    -       no      -
objfs           -              /system/object   objfs   -       no      -
swap            -              /tmp             tmpfs   -       yes     -
```

ZFS ファイルシステムは、ブート時に SMF サービスにより自動的にマウントされます。レガシーのマウント機能を使用し、vfstab に従って ZFS ファイルシステムをマウントできます。詳細については、『Oracle Solaris ZFS 管理ガイド』を参照してください。

/etc/vfstab の各フィールドの説明、およびこのファイルの編集方法と使用方法については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

NFS 環境

NFS は、1 つのシステム (通常はサーバー) のリソース (ファイルやディレクトリ) をネットワーク上のほかのシステムと共有するための分散型ファイルシステムサービスです。たとえば、他社製のアプリケーションやソースファイルをほかのシステム上のユーザーと共有できます。

NFS は、リソースの実際の物理的な位置をユーザーが意識しなくてすむようにします。よく使用されるファイルのコピーをシステムごとに配置しなくても、あるシステムのディスク上にコピーを 1 つ配置することによって NFS は、ほかのすべてのシステムがそのコピーにネットワークからアクセスできるようにします。NFS の環境では、リモートファイルはローカルファイルと区別がつきません。

詳細については、『Solaris のシステム管理 (ネットワークサービス)』の第 4 章「ネットワークファイルシステムの管理 (概要)」を参照してください。

システムは、ネットワーク上で共有するリソースがあるときに、NFS サーバーになります。サーバーは、現在共有されているリソースとそのアクセス制限 (読み取り / 書き込み、読み取り専用アクセスなど) のリストを管理します。

リソースを共有する場合は、リモートシステムにマウントできるように、そのリソースを使用可能な状態にします。

リソースを共有するには、次の方法があります。

- `share` コマンドまたは `shareall` コマンドを使用する
- `/etc/dfs/dfstab` (分散ファイルシステムテーブル) ファイルにエントリを追加し、システムをリブートする

リソースを共有する手順については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。NFS の詳細については、『Solaris のシステム管理 (ネットワークサービス)』の第 4 章「ネットワークファイルシステムの管理 (概要)」を参照してください。

NFS Version 4

Solaris リリースには、Sun の実装の NFS version 4 分散ファイルアクセスプロトコルが含まれています。

NFS version 4 では、ファイルアクセス、ファイルロック、およびマウントプロトコルが 1 つのプロトコルに統合されるので、ファイアウォールの通過が容易になり、セキュリティーが向上します。Solaris の NFS version 4 実装は、SEAM としても知られている Kerberos V5 と完全に統合されていますので、認証、整合性、およびプライバシの機能を備えています。NFS version 4 を使用して、クライアントとサーバーとの間で使用するセキュリティーの種類の交渉を行うこともできます。NFS version 4 を実装しているサーバーは、さまざまなセキュリティー種類とファイルシステムに対応できます。

NFS Version 4 の機能については、『Solaris のシステム管理 (ネットワークサービス)』の「NFS サービスの新機能」を参照してください。

自動マウント (autofs)

NFS ファイルシステムリソースをマウントするには、「自動マウント」(*autofs*) というクライアント側のサービスを使用します。autofs サービスにより、システムから NFS リソースにアクセスするたびに、これらのリソースを自動的にマウントしたりマウント解除したりできます。ユーザーがこのディレクトリ内で、このディレクトリに格納されているファイルを使用している間、ファイルシステムリソースはマウントされたままになります。リソースが一定の時間アクセスされなかった場合、リソースは自動的にマウント解除されます。

autofs サービスには、次の機能があります。

- システムブート時に NFS リソースをマウントする必要がないために、ブート時間が短くなります。
- NFS リソースをマウントまたはマウント解除するために、スーパーユーザーのパスワードを知っている必要はありません。
- NFS リソースは使用されるときにだけマウントされるために、ネットワークトラフィックが軽減されます。

autofs サービスは `automount` ユーティリティーによって初期化されます。このコマンドは、システムのブート時に自動的に実行されます。`automountd` デーモンは永続的に動作し、必要に応じて NFS ファイルシステムをマウントまたはマウント解除します。デフォルトでは、`/home` ファイルシステムは `automount` デーモンによってマウントされます。

autofs では、同じファイルシステムを提供するサーバーを複数指定できます。このような方法では、1 つのサーバーがダウンしても、autofs がその他のマシンからファイルシステムのマウントを試みることができます。

autofs の設定および管理方法については、『Solaris のシステム管理 (IP サービス)』を参照してください。

ファイルシステムのタイプを調べる

ファイルシステムのタイプは、次のいずれかによって判定できます。

- 仮想ファイルシステムテーブル (`/etc/vfstab` ファイル) 内の `FS type` フィールド
- ローカルファイルシステムの `/etc/default/fs` ファイル
- NFS ファイルシステムの `/etc/dfs/fstypes` ファイル

ファイルシステムのタイプを調べる方法

この手順は、ファイルシステムがマウントされているかどうかにかかわらず、使用できます。

`grep` コマンドを使用して、ファイルシステムのタイプを調べます。

```
$ grep mount-point fs-table
```

mount-point ファイルシステムタイプを調べるファイルシステムのマウントポイント名を指定します。たとえば、`/var` ディレクトリ。

fs-table ファイルシステムのタイプを調べるファイルシステムテーブルへの絶対パスを示します。ファイルシステムがマウントされている場合、*fs-table* は `/etc/mnttab` です。ファイルシステムがマウントされて

いない場合、*fs-table* は /etc/vfstab です。

マウントポイントの情報が表示されます。

注 – ディスクスライスの raw デバイス名がわかる場合、fstyp コマンドで、ファイルシステムのタイプを調べることができます (そのディスクスライスにファイルシステムが含まれている場合)。詳細は、fstyp(1M) のマニュアルページを参照してください。

例 16–1 ファイルシステムのタイプを調べる

次の例では、/etc/vfstab ファイルを使用して、/export ファイルシステムのタイプを調べます。

```
$ grep /export /etc/vfstab
/dev/dsk/c0t3d0s6   /dev/rdsk/c0t3d0s6  /export ufs   2       yes     -
$
```

次の例では、/etc/mnttab ファイルを使用して、現在マウントしている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
$
```

次の例は、fstyp コマンドを使ってファイルシステムのタイプを確認しています。

```
# fstyp /dev/rdsk/c0t0d0s0
zfs
```

◆ ◆ ◆ 第 17 章

ZFS、UFS、TMPFS、LOFS ファイルシステムの作成 (手順)

この章では、ZFS ファイルシステム、UFS ファイルシステム、一時ファイルシステム (TMPFS)、およびループバックファイルシステム (LOFS) の作成方法について説明します。TMPFS と LOFS は仮想ファイルシステムであるため、これらのファイルシステムを「実際に使用する」には、ファイルシステムをマウントします。

この章で説明する手順は次のとおりです。

- 355 ページの「Oracle Solaris ZFS ファイルシステムの作成」
- 357 ページの「UFS ファイルシステムを作成する方法」
- 359 ページの「マルチテラバイトの UFS ファイルシステムを作成する方法」
- 360 ページの「マルチテラバイトの UFS ファイルシステムを拡張する方法」
- 361 ページの「UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法」
- 363 ページの「TMPFS ファイルシステムを作成およびマウントする方法」
- 365 ページの「LOFS ファイルシステムを作成およびマウントする方法」

注 – UFS と DOS のファイルシステムをリムーバブルメディア上に作成する手順については、第 1 章「リムーバブルメディアの管理 (概要)」を参照してください。

Oracle Solaris ZFS ファイルシステムの作成

ZFS ファイルシステムは、特定のディスクパーティションに直接関連付けられていません。ZFS ファイルシステムで使用される ZFS ストレージプールには、多数のデバイスを含めることができます。ディスク全体またはディスクスライスを ZFS ストレージプールに追加できます。プールの中に、さらに別のファイルシステムを作成することをお勧めします。ファイルシステムに基づいてプールを管理すれば、プールに含まれるさまざまなデータを管理しやすくなります。

ZFS ストレージプールおよびファイルシステムの作成方法の詳細は、『Oracle Solaris ZFS 管理ガイド』を参照してください。

▼ Oracle Solaris ZFS ファイルシステムの作成方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. **ZFS** ストレージプールを作成します。

   次の例は、tank という名前の単純なミラー化ストレージプールと tank という名前の ZFS ファイルシステムを、1 つのコマンドで作成する方法を示しています。/dev/dsk/c1t0d0 ディスク全体と /dev/dsk/c2t0d0 ディスク全体を使用することを前提としています。

   ```
   # zpool create tank mirror c1t0d0 c2t0d0
   ```

3. **ZFS** ファイルシステムを作成します。

   ```
   # zfs create tank/fs
   ```

   この新規 ZFS ファイルシステム tank/fs では、ディスク領域を必要なだけ使用でき、/tank/fs に自動的にマウントされます。

4. ファイルシステムが作成されたことを確認します。

   ```
   # zfs list -r tank
   NAME      USED  AVAIL  REFER  MOUNTPOINT
   tank      117K   268G    21K  /tank
   tank/fs    21K   268G    21K  /tank/fs
   ```

UFS ファイルシステムの作成

UFS ファイルシステムをディスク上に作成する前に、そのディスクをフォーマットし、スライスに分割しなければなりません。「ディスクスライス」とは物理的なディスクのサブセットで、連続するブロックからなる 1 つの範囲のことです。スライスはスワップ空間などの raw デバイスとして使用することも、ディスクベースのファイルシステムとして使用することもできます。ディスクのフォーマットとスライスへの分割の方法については、第 10 章「ディスクの管理 (概要)」を参照してください。

Solaris ボリュームマネージャーなどのディスクおよびストレージ管理製品では、より洗練された「ボリューム」を作成できます。ボリュームは、1 つのスライスまたは 1 つのディスクの境界を超えて、拡張可能です。ボリュームの使用方法については、『Solaris ボリュームマネージャの管理』を参照してください。

注 – Solaris のデバイス名は、用語「スライス (デバイス名内の文字は s)」を使用して、スライス番号を参照します。スライスは「パーティション」とも呼ばれます。

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 オペレーティングシステム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
- ディスクの既存のパーティション構造を変更する場合
- ファイルシステムを完全に復元する場合

newfs コマンドを使用するのが、UFS ファイルシステムを作成する標準的な方法です。newfs コマンドは mkfs コマンドの使いやすいフロントエンドであり、実際に新しいファイルシステムを作成します。newfs コマンドは、パラメータのデフォルト値 (1 シリンダ当たりのトラック数や 1 トラック当たりのセクター数など) を、新しいファイルシステムが作成されるディスクのラベルから読み取ります。選択したオプションは、mkfs コマンドに渡され、ファイルシステムが作成されます。

newfs コマンドが使用するデフォルトのパラメータについては、newfs(1M) のマニュアルページを参照してください。

▼ UFS ファイルシステムを作成する方法

始める前に　次の前提条件を満たしているかどうかを確認します。

- ディスクがフォーマットされ、スライスに分割されている。
- 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。
- ファイルシステムを格納するスライスのデバイス名がわかっている。

ディスクおよびディスクスライス番号の検索方法については、第 11 章「ディスクの管理 (手順)」を参照してください。

ディスクのフォーマットとスライスへの分割方法については、第 10 章「ディスクの管理 (概要)」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name
```

-N	newfs コマンドが mkfs コマンドに渡すパラメータを表示します。ファイルシステムは実際に作成されません。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 1 ブロックあたり 4096 または 8192 バイトで指定します。デフォルトは 8192 バイトです。
-i *bytes*	i ノード 1 個当たりのバイト数を指定します。デフォルトはディスクのサイズによって異なります。詳細は、newfs(1M) のマニュアルページを参照してください。

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します。

システムから、確認を促すプロンプトが表示されます。

注意 – この手順を実行する前に、スライスのデバイス名が正しく指定されていることを確認してください。間違ったスライスを指定すると、その内容は新しいファイルシステムの作成時に消去されます。そして、システムがパニックを起こす原因となる可能性があります。

3　UFS ファイルシステムが作成されていることを確認するには、新しいファイルシステムを検査します。

```
# fsck /dev/rdsk/device-name
```

device-name 引数は、新しいファイルシステムを格納するディスクデバイスの名前を指定します。

`fsck` コマンドは、新しいファイルシステムの整合性を検査して、問題があれば通知し、問題を修復する前にプロンプトを表示します。`fsck` コマンドの詳細は、第 21 章「UFS ファイルシステムの整合性検査 (手順)」または `fsck(1m)` のマニュアルページを参照してください。

例 17–1　UFS ファイルシステムの作成

次の例は、`/dev/rdsk/c0t1d0s7` 上に UFS ファイルシステムを作成する方法を示しています。

```
# newfs /dev/rdsk/c0t1d0s7
/dev/rdsk/c0t1d0s7:  725760 sectors in 720 cylinders of 14 tracks, 72 sectors
        354.4MB in 45 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
 178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
 322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
 468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
 613376, 629584, 645792, 662000, 678208, 694416, 710624,
 fsck /dev/rdsk/c0t1d0s7
#
```

参考　UFS ファイルシステムを作成したあとの作業

UFS ファイルシステムをマウントし、使用可能にする場合は、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」に進みます。

▼ マルチテラバイトの UFS ファイルシステムを作成する方法

マルチテラバイトの UFS ファイルシステムは、Solaris ボリュームマネージャーまたは VxVM ボリューム、あるいは 1T バイトを超える物理ディスクとして提供されたマルチテラバイトの LUN が利用できることを前提にサポートされています。

マルチテラバイトの UFS ファイルシステムを作成する前に、次のいずれかを行う必要があります。

- `format` ユーティリティーまたは Solaris インストールユーティリティーを使ってマルチテラバイトのディスクパーティションを作成しておきます
- Solaris ボリュームマネージャーを使ってマルチテラバイトのボリュームを設定しておきます

マルチテラバイト UFS ファイルシステムの詳細は、341 ページの「64 ビット: マルチテラバイトの UFS ファイルシステムのサポート」を参照してください。

1. スーパーユーザーになります。

2. 論理ボリューム上にマルチテラバイトの **UFS** ファイルシステムを作成します。

   たとえば、次のコマンドを実行すると、1.8T バイトボリュームの UFS ファイルシステムが作成されます。

```
# newfs /dev/md/rdsk/d99
newfs: construct a new file system /dev/md/rdsk/d99: (y/n)? y
/dev/md/rdsk/d99:    3859402752 sectors in 628158 cylinders of 48 tracks,
128 sectors
        1884474.0MB in 4393 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....................................................................
super-block backups for last 10 cylinder groups at:
 3850872736, 3851751456, 3852630176, 3853508896, 3854387616, 3855266336,
 3856145056, 3857023776, 3857902496, 3858781216,
#
```

3. 新しく作成したファイルシステムの整合性を検査します。

   次に例を示します。

```
# fsck /dev/md/rdsk/d99
** /dev/md/rdsk/d99
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 2 used, 241173122 free (0 frags, 241173122 blocks, 0.0%
```

```
fragmentation)
#
```

4 新しく作成したファイルシステムをマウントして検査します。

次に例を示します。

```
# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem             size   used  avail capacity  Mounted on
/dev/md/dsk/d99        1.8T    64M   1.8T     1%    /bigdir
```

▼ マルチテラバイトの UFS ファイルシステムを拡張する方法

マルチテラバイトの UFS ファイルシステムを作成したあとは、growfs コマンドを使ってファイルシステムを拡張できます。たとえば、前の手順でボリュームに作成したファイルシステムを使用すると、別のディスクをこのボリュームに追加できます。その後で、ファイルシステムを拡張します。

1 スーパーユーザーになります。

2 別のディスクをボリュームに追加します。

次に例を示します。

```
# metattach d99 c4t5d0s4
d99: component is attached
# metastat
d99: Concat/Stripe
    Size: 5145882624 blocks (2.4 TB)
    Stripe 0:
        Device     Start Block  Dbase   Reloc
        c0t1d0s4      36864     Yes     Yes
    Stripe 1:
        Device     Start Block  Dbase   Reloc
        c3t7d0s4          0     No      Yes
    Stripe 2:
        Device     Start Block  Dbase   Reloc
        c1t1d0s4          0     No      Yes
    Stripe 3:
        Device     Start Block  Dbase   Reloc
        c4t5d0s4          0     No      Yes
```

3 ファイルシステムを拡張します。

次に例を示します。

```
# growfs -v /dev/md/rdsk/d99
/usr/lib/fs/ufs/mkfs -G /dev/md/rdsk/d99 5145882624
/dev/md/rdsk/d99:    5145882624 sectors in 837546 cylinders of 48 tracks,
128 sectors
        2512638.0MB in 5857 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
```

```
super-block backups (for fsck -F ufs -o b=#) at:
 32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
Initializing cylinder groups:
.....................................................................
super-block backups for last 10 cylinder groups at:
 5137130400, 5138009120, 5138887840, 5139766560, 5140645280, 5141524000,
 5142402720, 5143281440, 5144160160, 5145038880,
#
```

4 拡張したファイルシステムをマウントして検査します。

次に例を示します。

```
# mount /dev/md/dsk/d99 /bigdir
# df -h /bigdir
Filesystem             size   used  avail capacity  Mounted on
/dev/md/dsk/d99        2.4T    64M   2.4T     1%    /bigdir
```

▼ UFS ファイルシステムをマルチテラバイトの UFS ファイルシステムに拡張する方法

UFS ファイルシステムを 1T バイトを超えるサイズに拡張するには、次の手順を実行します。この手順は、newfs -T オプションを使って UFS ファイルシステムを作成したことを前提としています。

1 スーパーユーザーになります。

2 現在のディスクまたはボリュームのサイズを特定します。

たとえば、次のボリュームは 800G バイトです。

```
# metastat d98
d98: Concat/Stripe
    Size: 1677754368 blocks (800 GB)
    Stripe 0:
        Device     Start Block  Dbase   Reloc
        c0t1d0s4          0     No      Yes
    Stripe 1:
        Device     Start Block  Dbase   Reloc
        c3t7d0s4          0     No      Yes
```

3 このボリュームを **1T** バイトよりも大きくします。

次に例を示します。

```
# metattach d98 c1t1d0s4
d98: component is attached
# metastat d98
d98: Concat/Stripe
    Size: 2516631552 blocks (1.2 TB)
    Stripe 0:
        Device     Start Block  Dbase   Reloc
        c0t1d0s4          0     No      Yes
    Stripe 1:
```

```
        Device     Start Block  Dbase   Reloc
        c3t7d0s4          0     No      Yes
    Stripe 2:
        Device     Start Block  Dbase   Reloc
        c1t1d0s4          0     No      Yes
```

4 ディスクまたはボリュームの **UFS** ファイルシステムを **1T** バイトよりも大きくします。

次に例を示します。

```
growfs -v /dev/md/rdsk/d98
/usr/lib/fs/ufs/mkfs -G /dev/md/rdsk/d98 2516631552
/dev/md/rdsk/d98:    2516631552 sectors in 68268 cylinders of 144 tracks,
256 sectors
        1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 921888, 1843744, 2765600, 3687456, 4609312, 5531168, 6453024, 737...
 8296736,
Initializing cylinder groups:
.................................................
super-block backups for last 10 cylinder groups at:
 2507714848, 2508636704, 2509558560, 2510480416, 2511402272, 2512324128,
 2513245984, 2514167840, 2515089696, 2516011552,
```

5 拡張したファイルシステムをマウントして検査します。

次に例を示します。

```
# mount /dev/md/dsk/d98 /datadir
# df -h /datadir
Filesystem             size   used  avail capacity  Mounted on
/dev/md/dsk/d98        1.2T    64M   1.2T     1%    /datadir
```

マルチテラバイトの **UFS** ファイルシステムに関する問題の障害追跡

マルチテラバイトの UFS ファイルシステムに関する問題の障害追跡には、次のエラーメッセージと解決法を参考にしてください。

エラーメッセージ (次に例を示す)

```
mount: /dev/rdsk/c0t0d0s0 is not this fstype.
```

エラーの発生原因

1T バイトを超える UFS ファイルシステムを、この機能をサポートしていない Solaris リリースが動作するシステムにマウントしようとしました。

解決方法

1T バイトを超える UFS ファイルシステムは、現在の Solaris リリースが動作するシステムにマウントしてください。

エラーメッセージ

```
"File system was not set up with the multi-terabyte format."  "Its size
cannot be increased to a terabyte or more."
```

エラーの発生原因

`newfs -T` コマンドを使わずに作成したファイルシステムを拡張しようとしました。

解決方法

1. 1T バイトより大きくするファイルシステムのデータをバックアップします。
2. `newfs` コマンドを使用して、ファイルシステムをマルチテラバイトファイルシステムに再作成します。
3. 新しく作成したファイルシステムにバックアップされていたデータを復元します。

一時ファイルシステム (TMPFS) の作成

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。一般に、一時ファイルシステムは、UFS ファイルシステムに比べて読み書きの速度が高速です。TMPFS ファイルシステムを使用すると、ローカルディスク上で、あるいはネットワーク経由で一時ファイルの読み書きを行う際のオーバーヘッドが軽減されるのでシステムのパフォーマンスを向上できます。TMPFS ファイルシステム内のファイルは、リブートまたはマウント解除すると削除されます。

複数の TMPFS ファイルシステムを作成した場合は、すべてのファイルシステムが同じシステムリソースを使用するということに注意してください。`mount` コマンドの `-o size` オプションを使用して TMPFS のサイズを制限しないと、ある TMPFS ファイルシステムで作成されたファイルが、ほかの TMPFS ファイルシステムのための領域を使い切ってしまう可能性があります。

詳細は、`tmpfs(7FS)` のマニュアルページを参照してください。

▼ TMPFS ファイルシステムを作成およびマウントする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 必要に応じて、**TMPFS** ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir /mount-point
```

/mount-point は、TMPFS ファイルシステムがマウントされるディレクトリです。

3 **TMPFS ファイルシステムをマウントします。**

```
# mount -F tmpfs [-o size=number]  swap mount-point
```

`-o size=`*number* TMPFS ファイルシステムのサイズ制限を M バイト単位で示します。

mount-point TMPFS ファイルシステムがマウントされるディレクトリを示します。

ブート時に TMPFS ファイルシステムが自動的にマウントされるようにシステムを設定する方法については、例 17–3 を参照してください。

4 **TMPFS ファイルシステムが作成されていることを確認します。**

```
# mount -v
```

例 17–2 TMPFS ファイルシステムを作成およびマウントする

次の例は、TMPFS ファイルシステム `/export/reports` を作成およびマウントし、そのサイズを 50M バイトに制限する方法を示しています。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50m swap /export/reports
# mount -v
```

例 17–3 ブート時に TMPFS ファイルシステムをマウントする

ブート時にシステムが自動的に TMPFS ファイルシステムをマウントするように設定するには、`/etc/vfstab` のエントリを追加します。次の例は、ブート時に `/export/test` を TMPFS ファイルシステムとしてマウントする `/etc/vfstab` ファイルのエントリを示しています。`size=`*number* オプションを指定していないため、`/export/test` の TMPFS ファイルシステムのサイズは利用できるシステムリソースによってのみ制限されます。

```
swap - /export/test  tmpfs   -  yes  -
```

`/etc/vfstab` ファイルの詳細は、371 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) の作成およびマウント

LOFS ファイルシステムは、既存のファイルシステムに代替パスを提供する仮想ファイルシステムです。ほかのファイルシステムを LOFS 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 – LOFS ファイルシステムは慎重に作成してください。LOFS ファイルシステムは仮想ファイルシステムなので、ユーザーやアプリケーションを混乱させる可能性があります。

▼ LOFS ファイルシステムを作成およびマウントする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、**LOFS** ファイルシステムとしてマウントするディレクトリを作成します。

   # mkdir *loopback-directory*
3. 新しく作成したディレクトリに対して、適切なアクセス権と所有権を設定します。
4. 必要に応じて、**LOFS** ファイルシステムをマウントするマウントポイントを作成します。

   # mkdir */mount-point*
5. **LOFS** ファイルシステムをマウントします。

   # mount -F lofs *loopback-directory /mount-point*

   loopback-directory — ループバックマウントポイントにマウントするファイルシステムを指定します。

   /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します。
6. **LOFS** ファイルシステムがマウントされていることを確認します。

   # mount -v

例 17–4 LOFS ファイルシステムを作成およびマウントする

次の例は、新しいソフトウェアを、実際にはインストールしないで、ループバックファイルシステムとして /new/dist ディレクトリに作成、マウント、およびテストする方法を示しています。

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

例 17–5 ブート時に LOFS ファイルシステムをマウントする

ブート時にシステムが自動的に LOFS ファイルシステムをマウントするように設定するには、/etc/vfstab ファイルの最後にエントリを追加します。次の例は、ルート (/) ファイルシステムの LOFS ファイルシステムを /tmp/newroot にマウントする /etc/vfstab ファイルのエントリを示しています。

```
/ - /tmp/newroot  lofs   -  yes  -
```

ループバックファイルシステムのエントリは、/etc/vfstab ファイル内の最後のエントリでなければなりません。ループバックファイルシステムの /etc/vfstab エントリが、そこに組み込まれるファイルシステムよりも前にあると、ループバックファイルシステムをマウントできません。

参照 /etc/vfstab ファイルの詳細は、371 ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。

◆ ◆ ◆ 第 18 章

18 ファイルシステムのマウントとマウント解除 (手順)

この章では、Oracle Solaris OS でのファイルシステムのマウントとマウント解除の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。

- 373 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」
- 374 ページの「/etc/vfstab ファイルにエントリを追加する方法」
- 375 ページの「1 つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)」
- 377 ページの「UFS ファイルシステムのマウント方法 (mount コマンド)」
- 378 ページの「大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)」
- 379 ページの「NFS ファイルシステムのマウント方法 (mount コマンド)」
- 380 ページの「x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)」
- 382 ページの「ファイルシステムのマウント解除を確認する方法」
- 382 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」
- 383 ページの「1 つのファイルシステムをマウント解除する方法」

ファイルシステムのマウントの概要

ファイルシステムを作成したら、そのファイルシステムをシステムで使用できるようにする必要があります。ファイルシステムを使用できるようにするには、マウントします。マウントしたファイルシステムは、システムのディレクトリツリー内の指定したマウントポイントに接続されます。ルート (/) ファイルシステムは、常にマウントされています。

次の表に、ファイルシステムをその用途に応じてマウントするためのガイドラインを示します。

必要なマウントの種類	推奨されるマウント方法
ローカルまたはリモートのファイルシステムをときどきマウントする	コマンド行から手動で mount コマンドを入力します。
ローカルの UFS ファイルシステムを頻繁にマウントする。ローカルの ZFS ファイルシステムは、SMF サービスにより自動的にマウントされます。	/etc/vfstab 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します。
ホームディレクトリなどのリモートの UFS ファイルシステムを頻繁にマウントする	■ /etc/vfstab ファイルを使用して、システムがマルチユーザーモードでブートされたときに、自動的にファイルシステムをマウントします。 ■ autofs を使用して、ユーザーがアクセスするときに自動的にマウントします。その後、ファイルシステムから別のディレクトリに移動するときに自動的にマウント解除します。

ファイルシステムを含むリムーバブルメディアは、必要に応じてメディアをドライブに挿入し、volcheck コマンドを実行することによってマウントできます。リムーバブルメディアのマウント方法については、第 1 章「リムーバブルメディアの管理 (概要)」を参照してください。

UFS ファイルシステムのマウントとマウント解除に使用するコマンド

次の表に、UFS 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 18–1 UFS 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	マニュアルページ
mount	ファイルシステムとリモートリソースをマウントします。	mount(1M)
mountall	/etc/vfstab ファイルに指定されているすべてのファイルシステムをマウントします。システムがマルチユーザーモードになるときに、mountall コマンドが自動的に実行されます。	mountall(1M)
umount	ファイルシステムとリモートリソースをマウント解除します。	mount(1M)

表 18–1 UFS ファイルシステムのマウントとマウント解除に使用するコマンド (続き)

コマンド	説明	マニュアルページ
umountall	/etc/vfstab ファイルに指定されているすべてのファイルシステムをマウント解除します。	mountall(1M)

ZFS ファイルシステムをマウントおよびマウント解除する方法の詳細は、『Oracle Solaris ZFS 管理ガイド』を参照してください。

mount コマンドと mountall コマンドの使用時には、次の点に注意してください。

- mount や mountall コマンドでは、不整合が認められた読み取り / 書き込み用のファイルシステムをマウントできません。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検査する必要があります。ファイルシステムを検査する手順については、第 21 章「UFS ファイルシステムの整合性検査 (手順)」を参照してください。
- umount や umountall コマンドは、使用中のファイルシステムをマウント解除しません。ファイルシステムは、次のいずれかの場合に使用中であるとみなされます。
  - ユーザーがファイルシステム内のファイルまたはディレクトリにアクセスしている場合。
  - プログラムがそのファイルシステム上にあるファイルをオープンしている場合。
  - ファイルシステムが共有されている場合。
- 読み取り専用から読み取り / 書き込み専用アクセスへ再マウントする場合は、remount オプションを使用できます。読み取り / 書き込みアクセスから読み取り専用アクセスへの再マウントはできません。

UFS の汎用マウントオプション

次の表に、mount -o オプションで指定できる汎用オプションを示します。複数のオプションを指定する場合は、コンマ (空白を入れない) で区切ります。たとえば、-o ro,nosuid のようになります。

各ファイルシステムタイプで指定可能なマウントオプションのリストについては、各マウントコマンドのマニュアルページ (mount_ufs(1M) など) を参照してください。ZFS ファイルシステムのマウントについては、zfs(1M) を参照してください。

表 18–2 -o で指定する汎用マウントオプション

mount オプション	ファイルシステム	説明
bg | fg	NFS	最初のマウントに失敗すると、バックグラウンド (bg) またはフォアグラウンド (fg) で再試行します。このオプションは重要でない vfstab エントリには安全です。デフォルトは fg です。
hard | soft	NFS	サーバーが応答しない場合の手続きを指定します。soft オプションは、エラーが返されることを示します。hard オプションは、サーバーが応答するまで再試行要求が継続されることを示します。デフォルトは hard です。
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します。デフォルトは intr (割り込み可能) です。
largefiles | nolargefiles	UFS	2G バイトを超えるファイルを作成できるようにします。largefiles オプションを使ってマウントされたファイルシステムには、2G バイトを超えるファイルを格納できます。nolargefiles オプションを指定した場合、UFS ファイルシステムは以前の Solaris バージョンが動作しているシステムにはマウントできません。デフォルトは largefiles です。
logging | nologging	UFS	ファイルシステムのロギングを有効または無効にします。UFS ロギングとは、トランザクション (完全な UFS 処理を構成する変更) をログに保存してから、そのトランザクションを UFS ファイルシステムに適用するプロセスです。ロギングを使用すると、UFS ファイルシステムの整合性を保つことができます。つまり、fsck を省略できることを意味します。fsck を省略すると、システムがクラッシュしたとき、あるいはシステムをクリーンな状態でシャットダウンできなかったとき、システムをリブートする時間を省けます。 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領域ごとに約 1M バイトのサイズ (合計で 256M バイトまで) が割り当てられます。ファイルシステムに大量のシリンダグループが存在する場合、ログのサイズはこれより大きくなることがあります (最大で 512M バイトまで)。

表 18–2 -o で指定する汎用マウントオプション (続き)

mount オプション	ファイルシステム	説明
atime | noatime	UFS	ファイルのアクセス時間更新を抑制します。ただし、最後にファイルの状態が変わった時間または最後にファイルが変更された時間に対する更新が同時に行われる場合を除きます。詳細は、stat(2) のマニュアルページを参照してください。このオプションによって、アクセス時間が重要でないファイルシステム (Usenet ニューススプールなど) でのディスクに対する動作が減ります。デフォルトでは、通常アクセス時間 (atime) が記録されます。
remount	すべて	すでにマウントされているファイルシステムに関連付けられているマウントオプションを変更します。このオプションは、通常、ro 以外のオプションと併用できます。ただし、このオプションを使って変更できる内容は、ファイルシステムの種類によって異なります。
retry=*n*	NFS	マウント処理に失敗した場合に再試行します。*n* は再試行回数です。
ro | rw	CacheFS、NFS、PCFS、UFS、HSFS	読み取り / 書き込み (rw) または読み取り専用 (ro) を指定します。このオプションを指定しない場合のデフォルトは rw です。HSFS のデフォルトのオプションは ro です。
suid | nosuid	CacheFS、HSFS、NFS、UFS	setuid 実行を許可または禁止します。デフォルトは、setuid 実行を許可します。

/etc/vfstab ファイルのフィールドの説明

/etc/vfstab ファイル内のエントリには、次の表に示すように 7 つのフィールドがあります。

表 18–3 /etc/vfstab ファイルのフィールドの説明

フィールド名	説明
device to mount	このフィールドは、次のいずれかを指定します。 ■ ローカル UFS ファイルシステム用のブロックデバイス名 (/dev/dsk/c0t0d0s0 など)。 ■ リモートファイルシステム用のリソース名 (myserver:/export/home など)。NFS の詳細は、『Solaris のシステム管理 (IP サービス)』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)。 ■ 仮想ファイルシステム用のディレクトリ。
device to fsck	「device to mount」フィールドで指定した UFS ファイルシステムに対応する raw (キャラクタ型) デバイス名 (/dev/rdsk/c0t0d0s0 など)。このフィールドによって、fsck コマンドが使用する raw インタフェースが決まります。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 (-) を使用します。
マウントポイント	ファイルシステムのマウントポイントディレクトリ (/usr など) を指定します。
FS type	ファイルシステムのタイプを指定します。
fsck pass	fsck コマンドが UFS ファイルシステムを検査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検査されません。現在、/etc/vfstab ファイルの fsck pass 値はブートプロセスで無視されます。 このフィールドでゼロを指定すると、UFS ファイルシステムは検査されません。このフィールドに 0 より大きい値が指定されている場合、UFS ファイルシステムは常に検査されます。 このフィールドに 1 が指定されている場合、すべての UFS ファイルシステムは vfstab ファイル内の順番どおりに 1 つずつ検査されます。このフィールドに 1 より大きな値が指定され、さらに preen (修復) オプション (-o p) が指定されている UFS ファイルシステムが複数ある場合、効率を最大限に高めるために、fsck コマンドは複数のディスク上のファイルシステムを自動的に並行して検査します。それ以外の場合、このフィールドの値は意味を持ちません。
mount at boot	システムのブート時にファイルシステムが mountall コマンドによって自動的にマウントされるかどうかを yes または no で設定します。このフィールドは autofs とは連動していません。ルート (/)、/usr、/var のファイルシステムは最初は vfstab ファイルからマウントされません。これらのファイルシステムおよび /proc や /dev/fd などのような仮想ファイルシステムの場合、このフィールドは常に no に設定するべきです。

表 18–3 /etc/vfstab ファイルのフィールドの説明 (続き)

フィールド名	説明
mount options	ファイルシステムのマウントに使用されるオプションを (空白を入れずに) コンマで区切ったリスト。オプションなしを示すにはダッシュ (-) を使用します。よく使用されるマウントオプションの一覧については、表 18–2 を参照してください。

注 – /etc/vfstab ファイル内の各フィールドには必ずエントリが必要です。フィールドに値を指定しない場合は、必ずダッシュ (-) を入力してください。ダッシュを入力しないと、システムが正常にブートしない可能性があります。同様に、フィールドの値に空白文字を使用しないでください。

ファイルシステムのマウント

次の節では、/etc/vfstab ファイルにエントリを追加するか、コマンド行から mount コマンドを使用して UFS ファイルシステムをマウントする方法について説明します。

どのファイルシステムがマウントされているかを調べる方法

どのファイルシステムがすでにマウント済みであるかを調べるには、mount コマンドを使用します。

```
$ mount [ -v ]
```

-v は、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します。

例 18–1 どのファイルシステムがマウントされているかを調べる

この例は、mount コマンドを使用して、現在マウントされているファイルシステムに関する情報を表示する方法を示しています。

```
$ mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep  ...
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Sep  2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep  2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep  2 ...
```

例 18–1 どのファイルシステムがマウントされているかを調べる (続き)

```
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep  2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep  2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
$
```

この例は、`zfs mount` コマンドを使用して、現在マウントされている ZFS ファイルシステムに関する情報を表示する方法を示しています。

```
$ zfs mount
rpool/ROOT/zfs509BE           /
rpool/export                   /export
rpool/export/home              /export/home
rpool                          /rpool
```

▼ /etc/vfstab ファイルにエントリを追加する方法

レガシーのマウント動作が必要な ZFS ファイルシステムが存在しない場合は、ここに示す手順に従ってブート時に非 ZFS ファイルシステムをマウントします。ZFS ファイルシステムをマウントする方法の詳細は、『Oracle Solaris ZFS 管理ガイド』を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3 `/etc/vfstab` ファイルを編集し、エントリを追加します。次のことを確認してください。

a. 各フィールドを空白 (空白文字またはタブ) で区切ります。

b. フィールドで値を指定しない場合はダッシュ (-) を入力します。

c. 変更を保存します。

`/etc/vfstab` フィールドのエントリの詳細は、表 18–3 を参照してください。

注 – ルート (/) ファイルシステムは、ブートプロセスの過程でカーネルによって読み取り専用としてマウントされます。そのため、remount オプション (および、remount と一緒に使用できるオプション) だけが /etc/vfstab ファイルのルート (/) エントリでは有効です。

例 18–2 /etc/vfstab ファイルにエントリを追加する

次の例は、ディスクスライス /dev/dsk/c0t3d0s7 を UFS ファイルシステムとして、マウントポイント /files1 にマウントする方法を示しています。「device to fsck」として raw キャラクタ型デバイス /dev/rdsk/c0t3d0s7 を指定します。「fsck pass」の値が 2 なので、ファイルシステムは順不同で検査されます。

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
#
/dev/dsk/c0t3d0s7 /dev/rdsk/c0t3d0s7 /files1  ufs      2      yes       -
```

次の例は、システム pluto 上のディレクトリ /export/man を、NFS ファイルシステムとしてマウントポイント /usr/man にマウントする方法を示しています。ファイルシステムが NFS であるため、「device to fsck」や「fsck pass」は指定されません。この例では、「mount options」は ro (読み取り専用) と soft になっています。

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
pluto:/export/man   -                /usr/man nfs      -      yes       ro,soft
```

次の例は、ルート (/) ファイルシステムをループバックマウントポイント /tmp/newroot にマウントする方法を示しています。LOFS ファイルシステムをマウントするときは、LOFS ファイルシステム内に入るファイルシステムを先にマウントし、その後で LOFS をマウントします。

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
#
/                   -                /tmp/newroot lofs -      yes       -
```

▼ 1 つのファイルシステムをマウントする方法 (/etc/vfstab ファイル)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. /etc/vfstab ファイル内に列挙されているファイルシステムをマウントします。

   # **mount** */mount-point*

/mount-point は、/etc/vfstab ファイル内の「mount point」または「device to mount」フィールドのエントリを指定します。通常は、マウントポイントを指定するほうが簡単です。

例 18–3 1 つのファイルシステムをマウントする (/etc/vfstab ファイル)

次の例は、/etc/vfstab ファイル内に列挙されているファイルシステム /usr/dist をマウントする方法を示しています。

```
# mount /usr/dist
```

例 18–4 すべてのファイルシステムをマウントする (/etc/vfstab ファイル)

次に、ファイルシステムがすでにマウントされている状態で mountall コマンドを使用したとき表示されるメッセージの例を示します。

```
# mountall
/dev/rdsk/c0t0d0s7 already mounted
mount: /tmp already mounted
mount: /dev/dsk/c0t0d0s7 is already mounted, /export/home is busy,
        or the allowable number of mount points has been exceeded
```

mountall コマンドを実行すると、device to fsck エントリを持つすべてのファイルシステムが、マウント前に検査され、必要に応じて修正されます。

次の例は、/etc/vfstab ファイル内に列挙されているすべてのローカルシステムをマウントする方法を示しています。

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep  ...
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Sep  2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep  2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep  2 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep  2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep  2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
```

次の例は、使用可能なすべての ZFS ファイルシステムをマウントする方法を示します。

```
# zfs mount -a
```

次の例は、/etc/vfstab ファイル内に列挙されているすべてのリモートファイルシステムをマウントする方法を示しています。

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep  ...
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Sep  2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep  2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep  2 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep  2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep  2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
```

▼ UFS ファイルシステムのマウント方法 (mount コマンド)

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

   ```
   # mkdir /mount-point
   ```

   ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. UFS ファイルシステムをマウントします。

   ```
   # mount [-o mount-options] /dev/dsk/device-name /mount-point
   ```

`-o` *mount-options*	UFS ファイルシステムのマウントに使用できるマウントオプションを指定します。オプションの一覧は、表 18–2 または `mount_ufs(1M)` のマニュアルページを参照してください。
`/dev/dsk/`*device-name*	ファイルシステムが含まれているディスクスライス用のディスクデバイス名 (`/dev/dsk/c0t3d0s7` など) を指定します。ディスクのスライス情報を表示する方法については、222 ページの「ディスクスライス情報を表示する方法」を参照してください。
/mount-point	ファイルシステムをマウントするディレクトリを指定します。

例 18–5 UFS ファイルシステムをマウントする (mount コマンド)

次の例は、/dev/dsk/c0t3d0s7 を /files1 ディレクトリにマウントする方法を示しています。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

▼ 大規模ファイルを持たない UFS ファイルシステムをマウントする方法 (mount コマンド)

ファイルシステムのマウント時には、largefiles オプションがデフォルトで選択されます。このオプションにより、2G バイトを超えるファイルを作成できます。ファイルシステム内に大規模なファイルが存在する場合は、大規模ファイルをすべて削除し、fsck コマンドを実行して状態を「nolargefiles」にリセットしない限り、ファイルシステムを nolargefiles オプションを使って再マウントしたり、以前の Solaris バージョンが動作するシステムにマウントしたりすることはできません。

次の手順では、ファイルシステム用のエントリが /etc/vfstab ファイルにあるものとします。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

   ```
   # mkdir /mount-point
   ```

   ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。
3. ファイルシステム上に大規模ファイルが存在していないことを確認してください。

   ```
   # cd /mount-point
   # find . -xdev -size +20000000 -exec ls -l {} \;
   ```

   /mount-point には、大規模ファイルがあるかどうかを検査するファイルシステムのマウントポイントを指定します。
4. 大規模ファイルが当該ファイルシステム内に存在する場合は、必要に応じてそのファイルを削除するか、ほかのファイルシステムに移動します。
5. ファイルシステムをマウント解除します。

   ```
   # umount /mount-point
   ```

6 ファイルシステムの状態をリセットします。

```
# fsck /mount-point
```

7 nolargefiles オプションを指定してファイルシステムを再マウントします。

```
# mount -o nolargefiles /mount-point
```

例 18–6 大規模ファイルを持たないファイルシステムをマウントする (mount コマンド)

次の例は、/datab ファイルシステムを検査し、nolargefiles オプションを指定して再マウントする方法を示しています。

```
# cd /datab
# find . -xdev -size +20000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

▼ NFS ファイルシステムのマウント方法 (mount コマンド)

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3 リソース (ファイルまたはディレクトリ) がサーバーから使用可能かどうかを確認します。

NFS ファイルシステムをマウントするには、share コマンドを使用し、サーバー上のリソースを使用可能にしておかなければなりません。リソースを共有する方法については、『Solaris のシステム管理 (ネットワークサービス)』の「NFS サービスについて」を参照してください。

4 NFS ファイルシステムをマウントします。

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

-o *mount-options* NFS ファイルシステムのマウントに使用できるマウントオプションを指定します。よく使用される mount オプションの一覧に

ついては表 18–2、すべてのオプションの一覧については mount_nfs(1M) のマニュアルページを参照してください。

server:/directory 共有するリソースを持つサーバーのホスト名と、マウントするファイルまたはディレクトリへのパスを指定します。

/mount-point ファイルシステムをマウントするディレクトリを指定します。

例 18–7 NFS ファイルシステムをマウントする (mount コマンド)

次の例は、サーバー pluto の /export/packages ディレクトリを /mnt にマウントする方法を示しています。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)

次の手順で、PCFS (DOS) ファイルシステムをハードディスクからマウントします。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

ファイルシステムをマウントするには、ローカルシステム上にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3 PCFS ファイルシステムをマウントします。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
```

`-o rw | ro` PCFS ファイルシステムを読み取り / 書き込み (rw) または読み取り専用 (ro) にマウントできることを指定します。このオプションを指定しない場合のデフォルトは rw です。

`/dev/dsk/`*device-name* ディスク全体のデバイス名を指定します (/dev/dsk/c0t0d0p0 など)。

logical-drive DOS の論理ドライブ名 (c から z) またはドライブ番号 (1 から 24) を指定します。ドライブ c はドライブ 1 に相当し、ディスク上の基本 DOS スライスを表します。ほか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 論理ドライブを表します。

/mount-point ファイルシステムをマウントするディレクトリを指定します。

「device-name」と「logical-drive」は、コロンで区切る必要があります。

例 18–8 x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする (mount コマンド)

次の例は、基本 DOS スライス内の論理ドライブを /pcfs/c ディレクトリにマウントする方法を示しています。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例では、/mnt ディレクトリの拡張 DOS スライスに含まれる最初の論理ドライブを読み取り専用でマウントする方法を示します。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

UFS ファイルシステムをマウント解除する

UFS ファイルシステムをマウント解除すると、ファイルシステムがそのマウントポイントから削除され、そのエントリが /etc/mnttab ファイルから削除されます。マウントされているファイルシステム上では、一部のファイルシステム管理作業を実行できません。次の場合には、UFS ファイルシステムをマウント解除する必要があります。

- ファイルシステムが不要になった場合、または新しいソフトウェアが入ったファイルシステムに交換された場合。
- fsck コマンドを使用してファイルシステムを検査し、修復する必要がある場合。fsck コマンドの詳細は、第 21 章「UFS ファイルシステムの整合性検査 (手順)」を参照してください。

  完全バックアップの実行前に、ファイルシステムをマウント解除する必要があります。バックアップの実行の詳細は、第 24 章「UFS ファイルとファイルシステムのバックアップ (手順)」を参照してください。

注 – 各ファイルシステムは、ファイルシステムのシャットダウン手順の一部として自動的にマウント解除されます。

緊急時は、umount -f コマンドで、使用中のファイルシステムを強制的にマウント解除できます。ファイルを開いた状態でファイルシステムをマウント解除する

と、データが失われる可能性があるため、非常時以外はこの操作を行わないようにしてください。このオプションは、UFS と NFS のファイルシステムでのみ使用できます。

ファイルシステムをマウント解除する場合の前提条件

ファイルシステムをマウント解除する場合の前提条件は次のとおりです。

- スーパーユーザーになるか、同等の役割を引き受けます。
- ファイルシステムは、マウント解除が可能な状態でなければなりません。使用中のファイルシステムはマウント解除できません。ユーザーがそのファイルシステム内のディレクトリにアクセスしているとき、プログラムがそのファイルシステム上のファイルを開いているとき、またはファイルシステムが共有されているときには、ファイルシステムは使用中とみなされます。次の方法でファイルシステムをマウント解除が可能な状態にできます。
  - 別のファイルシステム内のディレクトリにカレントディレクトリを変更する。
  - システムからログアウトする。
  - `fuser` コマンドを使用して、そのファイルシステムを使用中のすべてのプロセスを表示し、必要に応じて終了させる。詳細は、382 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」を参照してください。

    ほか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ます。
  - ファイルシステムの共有を解除する。ファイルシステムの共有を解除する方法については、`unshare(1M)` のマニュアルページを参照してください。

ファイルシステムのマウント解除を確認する方法

ファイルシステムをマウント解除したことを確認するには、`mount` コマンドからの出力を調べます。

```
$ mount | grep unmounted-file-system
```

▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

2 どのプロセスを終了させるかがわかるように、ファイルシステムを使用中のすべてのプロセスを表示します。

```
# fuser -c [ -u ] /mount-point
```

`-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すべて表示されます。
`-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るファイルシステムの名前を指定します。

3 ファイルシステムを使用しているすべてのプロセスを終了させます。

```
# fuser -c -k /mount-point
```

ファイルシステムを使用している各プロセスに `SIGKILL` が送信されます。

注 – ユーザーのプロセスを終了させるときには、必ず事前に警告してください。

---

4 ファイルシステムを使用しているプロセスがないことを確認します。

```
# fuser -c /mount-point
```

例 18–9 あるファイルシステムを使用中のすべてのプロセスを終了させる

次の例は、`/export/home` ファイルシステムを使用中のプロセス `4006c` を終了させる方法を示しています。

```
# fuser -c /export/home
/export/home:     4006c
# fuser -c -k /export/home
/export/home:     4006c
# fuser -c /export/home
/export/home:
```

▼ 1 つのファイルシステムをマウント解除する方法

次の手順に従って、ファイルシステム (ルート (`/`)、`/usr`、`/var` を除く) をマウント解除します。

注 – ルート (`/`)、`/usr`、および `/var` ファイルシステムは、シャットダウン中にのみマウント解除可能です。これらのファイルシステムは、システムの機能上必須です。

---

1 382 ページの「ファイルシステムをマウント解除する場合の前提条件」の前提条件を満たしているかどうかを確認します。

2 ファイルシステムをマウント解除します。

```
# umount /mount-point
```

/mount-point は、マウント解除するファイルシステムの名前を示します。次のいずれかを指定できます。

- ファイルシステムがマウントされているディレクトリ名
- ファイルシステムのデバイス名パス
- NFS ファイルシステムのリソース
- LOFS ファイルシステムのループバックディレクトリ

例 18–10 ファイルシステムをアンマウントする

次の例は、ローカルのホームディレクトリから UFS ファイルシステムをマウント解除する方法を示しています。

```
# umount /export/home
```

次の例は、ローカルディスクの 7 番目のスライス上の UFS ファイルシステムをマウント解除する方法を示しています。

```
# umount /dev/dsk/c0t0d0s7
```

次の例は、UFS の `/export` ファイルシステムを強制的にマウント解除する方法を示しています。

```
# umount -f /export
#
```

次の例は、`/etc/vfstab` ファイル内のすべての UFS ファイルシステム (ルート (/)、`/proc`、`/var`、`/usr` を除く) をマウント解除する方法を示しています。

```
# umountall
```

次の例は、すべての ZFS ファイルシステムをマウント解除する方法を示しています。

```
# zfs umount -a
```

使用中のファイルシステムを除く、すべてのファイルシステムがマウント解除されます。

◆ ◆ ◆ 第 19 章

CacheFS ファイルシステムの使用 (手順)

この章では、CacheFS ファイルシステムの設定および管理の方法について説明します。

この章で説明する作業マップは、次のとおりです。

- 385 ページの「CacheFS ファイルシステムの概観 (作業マップ)」
- 388 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」
- 394 ページの「CacheFS ファイルシステムの保守 (作業マップ)」
- 399 ページの「CacheFS ファイルシステムのパッキング (作業マップ)」
- 409 ページの「CacheFS の統計情報の収集 (作業マップ)」

CacheFS に関するエラーの障害追跡については、405 ページの「cachefspack エラーの障害追跡」を参照してください。

注 – NFS version 4 および CacheFS ソフトウェアに関する重要な情報については、331 ページの「NFS Version 4 と CacheFS の互換性の問題」を参照してください。

CacheFS ファイルシステムの概観 (作業マップ)

次の作業マップは、CacheFS ファイルシステムを使用するためのすべての作業を確認するために使用してください。各作業は、CacheFS ファイルシステムの作成とマウント、キャッシュのパッキングと保守などの一連の付加された作業を指しています。

作業	説明	参照先
1. CacheFS ファイルシステムを作成してマウントします。	キャッシュを作成し、ファイルシステムをそのキャッシュにマウントします。	388 ページの「CacheFS ファイルシステムの作成とマウント (作業マップ)」

作業	説明	参照先
2. CacheFS ファイルシステムを保守します。	キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムを表示および変更します。	394 ページの「CacheFS ファイルシステムの保守 (作業マップ)」
3. (省略可能) CacheFS ファイルシステムをパックおよびパック解除します。	キャッシュをパックしてパッキングリストを使用するかどうかを決めます。キャッシュをパックすると、キャッシュ内の特定のファイルおよびディレクトリが常に更新されるようになります。	399 ページの「CacheFS ファイルシステムのパッキング (作業マップ)」
4. CacheFS の統計情報を収集します。	キャッシュの性能や適切なキャッシュサイズを決めます。	409 ページの「CacheFS の統計情報の収集 (作業マップ)」

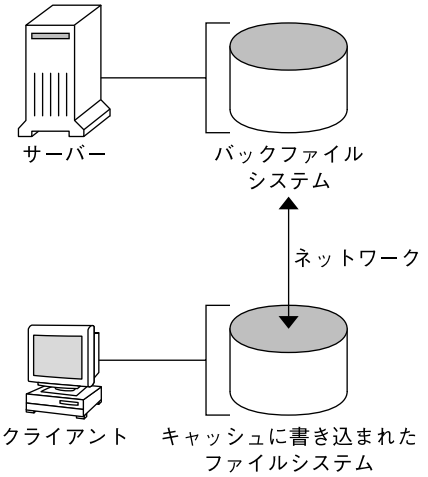
CacheFS ファイルシステムの概要

CacheFS ファイルシステムは、サーバーとネットワークの負荷を軽減して NFS サーバーのパフォーマンスとスケーラビリティーを改善する汎用ファイルシステムキャッシュメカニズムです。CacheFS ファイルシステムは、階層化ファイルシステムとして設計されており、あるファイルシステムを別のファイルシステムのキャッシュに書き込む機能を持っています。NFS 環境では、CacheFS ファイルシステムはサーバーあたりのクライアント比率を高め、サーバーとネットワークの負荷を軽減し、ポイントツーポイントプロトコル (PPP) などの低速リンク上のクライアントのパフォーマンスを向上させます。

CacheFS ファイルシステムの機能

CacheFS ファイルシステムをクライアントシステム上に作成すると、キャッシュに書き込んだファイルシステムをクライアントがネットワークを経由しなくても、ローカルにアクセスできます。次の図は、CacheFS ファイルシステムの使用に関連する構成要素の関係を示しています。

図 19–1　CacheFS ファイルシステムの機能



キャッシュ内にマウントするように指定するファイルシステムを「バック」ファイルシステムと呼びます。バックファイルシステムは、NFS と HSFS (High Sierra File System) のどちらかになります。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と、そのファイルはキャッシュに書き込まれます。「フロント」ファイルシステムとは、キャッシュにマウントされ、ローカルのマウントポイントからアクセスされるファイルシステムです。フロントファイルシステムのタイプは、UFS でなければなりません。

CacheFS ファイルシステム内のファイルにはじめてアクセスするときは、要求の処理速度が遅く感じられる可能性があります。しかし、2 回目以降は、もっと短い時間で処理できます。

CacheFS ファイルシステムの構造と動作

各キャッシュには、キャッシュの構造とその動作を決定する 1 組のパラメータが付いています。各パラメータは、次の表に示すデフォルト値に設定されています。デフォルト値は、フロントファイルシステム全体をキャッシュに使用するように指定しますが、これはファイルシステムをキャッシュに書き込む場合の推奨方法です。

表 19–1　CacheFS ファイルシステムのパラメータとそのデフォルト値

CacheFS ファイルシステムのパラメータ	デフォルト値	定義
`maxblocks`	90 %	CacheFS ファイルシステムがフロントファイルシステム内で要求できる最大ブロック数を設定します。

表 19–1 CacheFS ファイルシステムのパラメータとそのデフォルト値 (続き)

CacheFS ファイルシステムのパラメータ	デフォルト値	定義
minblocks	0 %	CacheFS ファイルシステムがフロントファイルシステム内で要求できる最小ブロック数を設定します。
threshblocks	85 %	CacheFS ファイルシステムが minblocks で指定したより多数のブロックを要求する前に、フロントファイルシステム内で使用可能でなければならないブロック数を設定します。
maxfiles	90 %	CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最大数 (ファイル数) を設定します。
minfiles	0 %	CacheFS ファイルシステムがフロントファイルシステム内で要求できる使用可能な i ノードの最小数を設定します。
threshfiles	85 %	CacheFS ファイルシステムが minfiles で指定したより多くのファイルを要求する前に、フロントファイルシステム内で使用可能でなければならない i ノード数を設定します。

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、それをほかのファイルシステムに使用する場合は、maxblocks と maxfiles の値を変更することをお勧めします。この変更を行うには、cfsadmin コマンドを使用します。次に例を示します。

```
$ cfsadmin -o maxblocks=60
```

CacheFS ファイルシステムの作成とマウント (作業マップ)

次の作業マップの手順を参照して、CacheFS ファイルシステムを作成およびマウントしてください。

作業	説明	参照先
1. キャッシュに書き込むファイルシステムを共有します。	キャッシュに書き込むファイルシステムが共有されていることを確認します。	share(1M)
2. キャッシュを作成します。	cfsadmin コマンドを使用してキャッシュを作成します。	389 ページの「キャッシュを作成する方法」

作業	説明	参照先
3. ファイルシステムをキャッシュにマウントします。	次のいずれかの方法を使用して、ファイルシステムをキャッシュにマウントします。	
	mount コマンドで、CacheFS ファイルシステムをマウントします。	390 ページの「CacheFS ファイルシステムのマウント方法 (mount)」
	/etc/vfstab ファイルを編集して、CacheFS ファイルシステムをマウントします。	392 ページの「CacheFS ファイルシステムのマウント方法 (/etc/vfstab)」
	autofs サービスを使用して、キャッシュされたファイルシステムをマウントします。	393 ページの「CacheFS ファイルシステムのマウント方法 (autofs)」

▼ キャッシュを作成する方法

1 クライアントシステムでスーパーユーザーになります。

2 キャッシュを作成します。

```
# cfsadmin -c /cache-directory
```

cache-directory は、キャッシュがあるディレクトリの名前を示します。

詳細は、cfsadmin(1M) のマニュアルページを参照してください。

注 – キャッシュを作成し終わったら、キャッシュディレクトリ内で処理を実行しないでください。何か実行すると、CacheFS ソフトウェア内で矛盾が生じる可能性があります。

例 19–1 キャッシュを作成する

次の例は、デフォルトのキャッシュパラメータ値を使用して、/local/mycache ディレクトリ内にキャッシュを作成する方法を示しています。

```
# mkdir /local
# cfsadmin -c /local/mycache
```

ファイルシステムをキャッシュにマウントする

キャッシュにマウントされるファイルシステムを指定して、ユーザーがそのファイルシステム内のファイルにローカルにアクセスできるようにします。実際には、各ファイルは、ユーザーがアクセスするまではキャッシュに書き込まれません。

次の表に、CacheFS ファイルシステムをマウントする 3 つの方法を示します。

CacheFS ファイルシステムのマウント方法	CacheFS マウント方法の使用頻度
mount コマンドを使用する方法	同じファイルにアクセスするには、システムをリブートするたびに mount コマンドを使用します。
/etc/vfstab ファイルを編集する方法	1 度だけですみます。/etc/vfstab ファイルの内容は、システムをリブートしたあとも変更されずに残ります。
autofs を使用する方法	1 度だけですみます。autofs のマップは、システムをリブートしたあとも変更されずに残ります。

いずれかの方法を選択してファイルシステムをマウントしてください。

共有可能なファイルシステムしかマウントできません。ファイルシステムの共有については、share(1M) のマニュアルページを参照してください。

注 – CacheFS ファイルシステムでは、ルート (/) と /usr のファイルシステムをキャッシュに書き込むことはできません。

▼ CacheFS ファイルシステムのマウント方法 (mount)

1 クライアントシステムでスーパーユーザーになります。

2 必要に応じて、マウントポイントを作成します。

```
# mkdir /mount-point
```

マウントポイントはどこからでも作成できますが、UFS ファイルシステムでなければなりません。次の手順のように、mount コマンドで CacheFS オプションを使用すると、作成するマウントポイントが指定したキャッシュディレクトリ内のキャッシュに書き込まれるように指定できます。

3 ファイルシステムをキャッシュにマウントします。

```
# mount -F cachefs -o backfstype=fstype,cachedir=/cache-directory[,options]
/back-filesystem /mount-point
```

fstype バックファイルシステムのファイルシステムタイプ (NFS または HSFS)。

/cache-directory キャッシュがある UFS ディレクトリの名前。389 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に指定した名前と同じです。

options 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ほかのマウントオプション。CacheFS の mount オプションの詳細は、`mount_cachefs(1M)` のマニュアルページを参照してください。

/back-filesystem キャッシュにマウントするバックファイルシステムのマウントポイント。バックファイルシステムが NFS ファイルシステムである場合は、ファイルシステムのマウント元となるサーバーのホスト名と、キャッシュにマウントするファイルシステム名 (コロンで区切る) を指定する必要があります。たとえば、*merlin:/data/abc*。

/mount-point ファイルシステムのマウント先となるディレクトリ。

4 作成したキャッシュが実際にマウントされたかどうかを確認します。

```
# cachefsstat /mount-point
```

/mount-point は作成した CacheFS ファイルシステムです。

次に例を示します。

```
# cachefsstat /docs
/docs
            cache hit rate:   100% (0 hits, 0 misses)
        consistency checks:      1 (1 pass, 0 fail)
                  modifies:      0
        garbage collection:      0
```

ファイルシステムがキャッシュにマウントされなかった場合は、次のようなエラーメッセージが表示されます。

```
# cachefsstat /mount-point
cachefsstat: mount-point: not a cachefs mountpoint
```

`cachefsstat` コマンドの詳細は、409 ページの「CacheFS の統計情報の収集」を参照してください。

例 19–2 CacheFS ファイルシステムをマウントする (mount)

次の例は、NFS ファイルシステム `merlin:/docs` を `/docs` という CacheFS ファイルシステムとして、`/local/mycache` というキャッシュにマウントする方法を示しています。

```
# mkdir /docs
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

次の例は、Solaris 10 DVD を `/cfssrc` という CacheFS ファイルシステムとして使用できるようにする方法を示しています。DVD には書き込めないので、引数 `ro` を指定して

CacheFS ファイルシステムを読み取り専用にします。この例では、リムーバブルメディアサービスを実行していないものとします。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /sol10
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/sol10 /dev/dsk/c0t6d0s0 /cfssrc
# ls /cfssrc
Copyright  Solaris_10
```

次の例は、vold を実行しながら、Solaris 10 DVD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/sol_10_sparc/s0 /vol/dev/dsk/c0t2d0/sol_10_sparc/s0 /cfssrc
```

次の例は、vold を実行しながら、DVD を CacheFS ファイルシステムとしてマウントする方法を示しています。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/epson /vol/dev/dsk/c0t2d0/epson /drvrs
```

次の例では、demandconst オプションを使用して、NFS CacheFS ファイルシステム /docs の整合性検査を指定します。/docs のバックファイルシステムは merlin:/docs です。詳細は、396 ページの「CacheFS ファイルシステムの整合性検査」を参照してください。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ CacheFS ファイルシステムのマウント方法 (/etc/vfstab)

1. クライアントシステムでスーパーユーザーになります。
2. エディタを使用して、マウントするファイルシステムを /etc/vfstab ファイル内で指定します。

   下の例を参照してください。

   /etc/vfstab ファイルの詳細は、371 ページの「/etc/vfstab ファイルのフィールドの説明」を参照してください。
3. CacheFS ファイルシステムをマウントします。

   # mount */mount-point*

   あるいはシステムをリブートします。

例 19–3 CacheFS ファイルシステムをマウントする (/etc/vfstab)

次の例は、キャッシュされたディレクトリ /opt/cache にマウントされるリモートシステム starbug からの /data/abc ディレクトリ用の /etc/vfstab エントリを示しています。

```
#device           device             mount      FS     fsck  mount   mount
#to mount         to fsck            point      type   pass  at boot options
#
starbug:/data/abc /local/abc         /opt/cache cachefs 7     yes     local-access,bg,
nosuid,demandconst,backfstype=nfs,cachedir=/opt/cache
```

▼ CacheFS ファイルシステムのマウント方法 (autofs)

自動マウントマップ内で -fstype=cachefs マウントオプションを指定して、autofs によってファイルシステムをキャッシュにマウントします。CacheFS のマウントオプション (backfstype や cachedir など) も、自動マウントマップ内で指定します。

自動マウントマップの詳細は、『Solaris のシステム管理 (ネットワークサービス)』の「autofs 管理作業の概要」または automount(1M) のマニュアルページを参照してください。

1. クライアントシステムでスーパーユーザーになります。
2. エディタを使用して、次の 1 行を auto_direct マップに追加します。

   /mount-point -fstype=cachefs,cachedir=/*directory*,backfstype=nfs *server:/file-system*
3. エディタを使用して、次の 1 行を auto_master マップに追加します。

   /-

   /- エントリによって、auto_direct マップを検査するように指示します。
4. システムを再起動します。
5. エントリが正しく作成されたか確認するには、次のようにキャッシュにマウントしたファイルシステムにカレントディレクトリを変更して内容を表示します。

   ```
   # cd /filesystem
   # ls
   ```

例 19–4 CacheFS ファイルシステムをマウントする (autofs)

次の auto_direct エントリは、CacheFS ファイルシステムを /docs ディレクトリに自動的にマウントします。

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

CacheFS ファイルシステムの保守 (作業マップ)

CacheFS ファイルシステムを設定し終わったら、保守作業を多少行う必要があります。CacheFS ファイルシステムの保守作業を行う必要がある場合は、次の作業マップの手順 (オプション) に従ってください。

作業	説明	参照先
CacheFS ファイルシステムを変更します。	キャッシュをマウント解除、削除、または作成し直して、CacheFS ファイルシステムの動作を変更します。	395 ページの「CacheFS ファイルシステムの変更」
CacheFS ファイルシステム情報を表示します。	`cfsadmin` コマンドを使用して、CacheFS ファイルシステムに関する情報を表示します。	395 ページの「CacheFS ファイルシステムに関する情報を表示する方法」
整合性検査を実行します。	`cfsadmin` コマンドを使用して必要に応じて整合性検査を実行します。	396 ページの「必要に応じてキャッシュの整合性検査を指定する方法」
CacheFS ファイルシステムを削除します。	`umount` コマンドと `cfsadmin` コマンドを使用して、CacheFS ファイルシステムを削除します。	397 ページの「CacheFS ファイルシステムを削除する方法」
CacheFS ファイルシステムの整合性を検査します。	`fsck_cachefs` コマンドを使用して、CacheFS ファイルシステムの整合性を検査します。	398 ページの「CacheFS ファイルシステムの整合性を検査する方法」

CacheFS ファイルシステムの保守

この節では、CacheFS ファイルシステムの保守方法について説明します。

`/etc/vfstab` ファイルを使用してファイルシステムをマウントしている場合は、このファイル内でファイルシステムオプションを編集してキャッシュを変更します。autofs を使用している場合は、autofs マップ内でファイルシステムオプションを編集してキャッシュを変更します。

CacheFS ファイルシステムの変更

キャッシュ内でファイルシステムを変更する場合は、キャッシュを削除してから作成し直す必要があります。また、ファイルシステムの共有方法とアクセス方法によっては、マシンをシングルユーザーモードでリブートしなければならない場合があります。

次の例では、キャッシュが削除されてから再び作成され、ファイルシステム /docs に指定された demandconst オプションを使用して再びマウントされます。

```
# shutdown -g30 -y
.
.
.
Root password for system maintenance (control-d to bypass):
single-user privilege assigned to /dev/console.

.
.
.
Here is where you might be prompted 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
#
```

▼ CacheFS ファイルシステムに関する情報を表示する方法

1. クライアントシステムでスーパーユーザーになります。
2. 指定したキャッシュにマウントされたすべてのファイルシステムに関する情報を表示します。

   # cfsadmin -l */cache-directory*

   /cache-directory は、キャッシュがあるディレクトリの名前です。

例 19–5 CacheFS ファイルシステムに関する情報を表示する

次の例は、/local/mycache キャッシュディレクトリに関する情報を示しています。この例では、/docs ファイルシステムが /local/mycache にキャッシュされます。最終行には、CacheFS ファイルシステムの名前が表示されます。

```
# cfsadmin -l /local/mycache
cfsadmin: list cache FS information
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
   maxfilesize    3MB
merlin:_docs:_docs
#
```

CacheFS ファイルシステムの整合性検査

CacheFS ソフトウェアでは、キャッシュされたディレクトリとファイルが最新の状態に保たれることを保証するために、キャッシュに格納されているファイルの整合性を定期的に検査します。整合性を検査するために、CacheFS ソフトウェアでは現在の変更時刻を前回の変更時刻と比較します。変更時刻が異なる場合は、そのディレクトリまたはファイルに関するすべてのデータと属性がキャッシュから消去されます。そして、バックファイルシステムから新しいデータと属性が取り出されます。

必要に応じて行う整合性検査

整合性検査は、-o demandconst オプションを使用してマウントされたファイルシステムに対して明示的に要求した場合にのみ実行できます。このオプションでファイルシステムをキャッシュにマウントした場合は、-s オプションを指定した cfsadmin コマンドを実行して整合性検査を要求します。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検査されます。ファイルがアクセスされなければ、検査は実行されません。-o demandconst オプションを使用すると、整合性検査によってネットワークがいっぱいになるという事態を回避できます。

詳細は、mount_cachefs(1M) のマニュアルページを参照してください。

▼ 必要に応じてキャッシュの整合性検査を指定する方法

1 クライアントシステムでスーパーユーザーになります。

2 ファイルシステムをキャッシュにマウントし、キャッシュの整合性検査を指定します。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/directory,demandconst
server:/file-system  /mount-point
```

3 特定の CacheFS ファイルシステムに対する整合性検査を開始します。

```
# cfsadmin -s /mount-point
```

▼ CacheFS ファイルシステムを削除する方法

1 クライアントシステムでスーパーユーザーになります。

2 CacheFS ファイルシステムをマウント解除します。

```
# umount /mount-point
```

/mount-point は、削除する CacheFS ファイルシステムを示します。

3 CacheFS ファイルシステムの名前 (キャッシュ ID) を判別します。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
   maxfilesize    3MB
cache-ID
#
```

4 指定したキャッシュから CacheFS ファイルシステムを削除します。

```
# cfsadmin -d cache-ID /cache-directory
```

cache-ID CacheFS ファイルシステム名を表し、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ます。詳細は、395 ページの「CacheFS ファイルシステムに関する情報を表示する方法」を参照してください。*cache-ID* に `all` を指定すると、特定のキャッシュに書き込まれた CacheFS ファイルシステムをすべて削除できます。

/cache-directory キャッシュがあるディレクトリを指定します。

5 CacheFS ファイルシステムが削除されたことを確認します。

直前に削除したファイルシステムのキャッシュ ID が、`cfsadmin -l` の出力に表示されなくなります。

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
```

```
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
   maxfilesize    3MB
#
```

このコマンド出力に指定されるフィールドについては、cfsadmin(1M) のマニュアルページを参照してください。

6 キャッシュのリソース数を更新します。

```
# fsck -F cachefs /cache-directory
```

詳細は、398 ページの「CacheFS ファイルシステムの整合性を検査する方法」を参照してください。

例 19–6 CacheFS ファイルシステムを削除する

次の例は、キャッシュからファイルシステムを削除する方法を示しています。

```
# umount /cfssrc
# cfsadmin -l /cfssrc
# cfsadmin -d _dev_dsk_c0t6d0s0:_cfssrc
# cfsadmin -l
# fsck -F cachefs /cache-directory
```

▼ CacheFS ファイルシステムの整合性を検査する方法

fsck コマンドを使用して、CacheFS ファイルシステムの整合性を検査します。何も操作しなくても、fsck コマンドの CacheFS バージョンによって問題が自動的に解決されます。fsck コマンドはブート時またはファイルシステムのマウント時に自動的に実行されるため、CacheFS ファイルシステムに対して fsck コマンドを手作業で実行する必要はありません。整合性を手作業で検査する場合は、次の手順を使用できます。

詳細は、fsck_cachefs(1M) のマニュアルページを参照してください。

1 クライアントシステムでスーパーユーザーになります。

2 指定されたキャッシュ内でファイルシステムを検査します。

```
# fsck -F cachefs [-m -o noclean] /cache-directory
```

-m fsck コマンドに CacheFS ファイルシステムを検査させますが、修復は行いません。

`-o noclean`	CacheFS ファイルシステムに対する検査のみを実行させます。修復は行いません。
/cache-directory	キャッシュがあるディレクトリの名前を指定します。

例 19–7 CacheFS ファイルシステムの整合性を検査する

次の例は、`/local/mycache` キャッシュに書き込まれているファイルシステムを検査する方法を示しています。

```
# fsck -F cachefs /local/mycache
#
```

CacheFS ファイルシステムのパッキング (作業マップ)

次の作業マップでは、CacheFS ファイルシステムのパッキングに関連する手順について説明します。これらの手順はすべてオプションです。

作業	説明	参照先
ファイルをキャッシュにパックします。	キャッシュにロードされるファイルとディレクトリを識別し、それらをパックします。パッキングは、これらのファイルのコピーがキャッシュで利用できることを保証します。	400 ページの「キャッシュにファイルをパックする方法」
パッキングリストを作成します。	キャッシュにパックするファイルを個々に指定しない場合は、パッキングリストを作成します。	403 ページの「パッキングリストを作成する方法」
パッキングリストを使ってファイルをキャッシュにパックします。	キャッシュにパックするファイルが記載されているパッキングリストの名前を指定します。	403 ページの「パッキングリストを使ってファイルをキャッシュにパックする方法」
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します。	不要になったファイルをキャッシュから削除します。	404 ページの「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」404 ページの「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」

作業	説明	参照先
パックされたファイルの情報を表示します。	パックしたファイルに関する情報 (パッキング状況など) を表示します。	401 ページの「パックされたファイルの情報を表示する方法」

CacheFS ファイルシステムのパッキング

一般的な使い方では、設定が終わると、CacheFS ソフトウェアは、ユーザーが要求しなくても自動的に適切な動作をします。最近使用されたファイルがキャッシュされます。「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。

`cachefspack` コマンドを使用すると、キャッシュにロードするファイルとディレクトリを指定できます。このコマンドは、これらのファイルのコピーがキャッシュで利用できることを保証します。

パッキングリストには、特定のファイル名やディレクトリ名が入っています。ほかのパッキングリストを入れることも可能です。この機能によって、たくさんの項目をキャッシュにパックする場合に、個々のファイルやディレクトリを指定する手間が省けます。

次のように、`-h` オプションを使用すると、`cachefspack` のすべてのオプションの簡単なヘルプ情報を出力できます。

```
$ cachefspack -h
Must select 1 and only 1 of the following 5 options
-d Display selected filenames
-i Display selected filenames packing status
-p Pack selected filenames
-u Unpack selected filenames
-U Unpack all files in directory 'dir'
-f Specify input file containing rules
-h Print usage information
-r Interpret strings in LIST rules as regular expressions
-s Strip './' from the beginning of a pattern name
-v Verbose option
files - a list of filenames to be packed/unpacked
```

▼ キャッシュにファイルをパックする方法

- ファイルをキャッシュにパックします。

  ```
  $ cachefspack -p filename
  ```

  `-p` ファイルまたは複数のファイルをパックすることを示します。このオプションはデフォルトです。

filename キャッシュにパックするファイルまたはディレクトリの名前を指定します。ディレクトリを指定すると、そのサブディレクトリもすべてパックされます。詳細は、cachefspack(1M) のマニュアルページを参照してください。

例 19–8 例 — ファイルをキャッシュにパックする

次の例は、projects ファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects
```

次の例は、3 つのファイルがキャッシュにパックされることを示しています。

```
$ cachefspack -p projects updates master_plan
```

次の例は、ディレクトリがキャッシュにパックされることを示しています。

```
$ cachefspack -p /data/abc/bin
```

▼ パックされたファイルの情報を表示する方法

- パックされたファイルの情報を表示します。

  $ **cachefspack -i[v]** *cached-filename-or-directory*

-i	パックされたファイルの情報を表示することを示します。
-v	詳細表示オプションです。
cached-filename-or-directory	情報を表示するファイル名またはディレクトリ名を指定します。

例 19–9 パックされたファイルの情報を表示する

次の例は、doc_file ファイルが正常にパックされていることを示しています。

```
$ cachefspack -i doc_file
cachefspack: file doc_file marked packed YES, packed YES
```

次の例では、/data/abc ディレクトリに bin サブディレクトリがあります。bin サブディレクトリには、 big、medium、および small という 3 つのファイルがあります。big と small ファイルはパックするように指定されていますが、パックされていません。medium ファイルは正常にパックされています。

```
$ cd /data/abc
$ cachefspack -i bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO
cachefspack: file /bin/medium marked packed YES,
packed YES
cachefspack: file /bin/small marked packed YES,
packed NO
.
.
.
```

-iv オプションを指定した場合は、指定したファイルまたはディレクトリがキャッシュからフラッシュされたかどうかに関する追加情報が表示されます。次に例を示します。

```
$ cd /data/bin
FSCACHEPACK-4$ cachefspack -iv bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/medium marked packed YES,
packed YES, nocache NO
cachefspack: file /bin/small marked packed YES,
packed NO
nocache NO
.
.
.
```

上記例の最後の行は、ディレクトリの内容がキャッシュからフラッシュされていないことを示しています。

パッキングリストの使用

cachefspack コマンドの機能の 1 つに、パッキングリストを作成するという機能があります。

パッキングリストには、キャッシュにパックするファイルやディレクトリが入っています。ディレクトリがパッキングリストに入っている場合、そのサブディレクトリとファイルもすべてパックされます。

この機能によって、キャッシュにパックする各ファイルをそれぞれ指定するという手間が省けます。

▼ パッキングリストを作成する方法

- vi を使ってパッキングリストファイルを作成する

  パッキングリストファイルの書式は、filesync コマンドで使用する書式と同じです。詳細は、filesync(1) のマニュアルページを参照してください。

  パッキングリストには、次の 2 つの機能があります。

  - パッキングリスト内のファイルを文字どおりのファイル名ではなく、正規表現として識別することができるため、ファイル名を個別に指定する必要がありません。
  - 所有するファイルだけがパックされるようにすることで、共有ディレクトリのファイルをパックできます。

  これらの機能の使い方については、cachefspack(1M) のマニュアルページを参照してください。

例 19–10 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、当該ディレクトリ中のパックされるファイルを指定しています。
- IGNORE 文は、パックしないファイルタイプ (ここでは .ps) を指定しています。

▼ パッキングリストを使ってファイルをキャッシュにパックする方法

- パッキングリストにファイルをパックする

  $ cachefspack -f *packing-list*

  `-f` パッキングリストを使用することを指定します。

  packing-list パッキングリスト名を指定します。

例 19–11 パッキングリストを使ってファイルをキャッシュにパックする

この例では、list.pkg ファイルを cachefspack コマンドのパッキングリストとして使用しています。

```
$ cachefspack -f list.pkg
```

キャッシュからファイルまたはパッキングリストのパッキングを解除する

キャッシュからファイルを削除、つまりそのパッキングを「解除」する場合もあります。ほかよりも優先度が高いファイルまたはディレクトリがいくつか存在する場合があるため、重要でないファイルのパッキングを解除する必要があります。たとえば、あるプロジェクトを終了して、そのプロジェクトに関連するファイルをアーカイブしたと仮定します。次は、新しいプロジェクト、つまり新しいファイルのセットで作業することになります。

▼ キャッシュからファイルまたはパッキングリストのパッキングを解除する方法

- キャッシュからファイルまたはパッキングリストのパッキングを解除します。

  ```
  $ cachefspack -u filename | -U cache-directory
  ```

  -u　ファイルまたは複数のファイルのパッキングを解除することを示します。このオプションを使用する場合は、ファイル名を指定しなければなりません。

  filename　キャッシュからパッキングを解除するファイルまたはパッキングリストの名前を示します。

  -U　キャッシュ中のすべてのファイルのパッキングを解除することを示します。

  詳細は、cachefspack(1M) のマニュアルページを参照してください。

例 19–12 キャッシュからファイルまたはパッキングリストのパッキングを解除する

次の例は、キャッシュから /data/abc/bin/big ファイルのパッキングが解除されることを示しています。

```
$ cachefspack -u /data/abc/bin/big
```

次の例は、キャッシュから 3 つのファイルのパッキングが解除されることを示しています。

```
$ cd /data/abc/bin/big
$ cachefspack -u big small medium
```

次の例は、パッキングリストのパッキングを解除する方法を示しています。パッキングリストは、ファイルのディレクトリへのパスを含むファイルです。

```
$ cachefspack -uf list.pkg
```

次の例は、キャッシュディレクトリ中のすべてのファイルのパッキングが解除されるように -U オプションを使用する方法を示しています。

```
$ cachefspack -U /local/mycache
```

ファイルシステムを 1 つもマウントしていないキャッシュのパッキングは解除できません。-U オプションで、マウント済みファイルシステムを持っていないキャッシュを指定した場合、次のような出力が表示されます。

```
$ cachefspack -U /local/mycache
cachefspack: Could not unpack cache /local/mycache, no mounted
filesystems in the cache.
```

cachefspack エラーの障害追跡

cachefspack コマンドを使用すると、次のようなエラーメッセージが表示されることがあります。

```
cachefspack: pathname - can't open directory: permission denied
```

エラーの発生原因
　ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

対処方法
　適切なアクセス権を取得してください。

```
cachefspack: pathname - can't open directory: no such file or
directory
```

エラーの発生原因
　ファイルまたはディレクトリの指定が間違っている可能性があります。

対処方法
　入力ミスがないか確認してください。

```
cachefspack: pathname - can't open directory: stale NFS file handle
```

エラーの発生原因
　アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

対処方法
　サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

```
cachefspack: pathname - can’t open directory: interrupted system
call
```

エラーの発生原因
　コマンドの実行中に間違って Control + C を押した可能性があります。

対処方法
　このコマンドを再度実行してください。

```
cachefspack: pathname - can’t open directory: I/O error
```

エラーの発生原因
　ハードウェアの障害の可能性があります。

対処方法
　ハードウェアの接続を確認してください。

```
cachefspack: error opening dir
```

エラーの発生原因
　ファイルまたはディレクトリの指定が間違っている可能性があります。ファイル書式中の BASE コマンドの後に指定したパスが、ディレクトリではなくファイルになっている可能性があります。指定するパスはディレクトリでなければなりません。

対処方法
　入力ミスがないか確認してください。ファイル書式中の BASE コマンドの後に指定したパスを確認してください。ファイルではなくディレクトリが指定されていることを確認してください。

```
cachefspack: unable to get shared objects
```

エラーの発生原因
　実行可能ファイルが壊れているか、その形式を認識できません。

対処方法
　実行可能ファイルを交換してください。

```
cachefspack: filename - can’t pack file: permission denied
```

エラーの発生原因
　ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

対処方法
　適切なアクセス権を取得してください。

`cachefspack: `*filename*` - can’t pack file: no such file or directory`

エラーの発生原因
　ファイルまたはディレクトリの指定が間違っている可能性があります。

対処方法
　入力ミスがないか確認してください。

`cachefspack: `*filename*`- can’t pack file: stale NFS file handle`

エラーの発生原因
　アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

対処方法
　サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

`cachefspack: `*filename*`- can’t pack file: interrupted system call`

エラーの発生原因
　コマンドの実行中に間違って Control + C を押した可能性があります。

対処方法
　このコマンドを再度実行してください。

`cachefspack: `*filename*`- can’t pack file: I/O error`

エラーの発生原因
　ハードウェアの障害の可能性があります。

対処方法
　ハードウェアの接続を確認してください。

`cachefspack: `*filename*`- can’t pack file: no space left on device.`

エラーの発生原因
　キャッシュのディスク容量が不足しています。

対処方法
　ディスク容量を増やしてキャッシュのサイズを大きくする必要があります。

`cachefspack: `*filename*` - can’t unpack file: permission denied`

エラーの発生原因
　ファイルまたはディレクトリにアクセスするための正しいアクセス権を持っていません。

対処方法
　適切なアクセス権を取得してください。

`cachefspack: *filename* - can't unpack file: no such file or directory`

エラーの発生原因
　ファイルまたはディレクトリの指定が間違っている可能性があります。

対処方法
　入力ミスがないか確認してください。

`cachefspack: *filename*- can't unpack file: stale NFS file handle`

エラーの発生原因
　アクセスしようとしたときに、ファイルまたはディレクトリがサーバーから移動または削除されていた可能性があります。

対処方法
　サーバー上のファイルやディレクトリにまだアクセスできることを確認してください。

`cachefspack: *filename*- can't unpack file: interrupted system call`

エラーの発生原因
　コマンドの実行中に間違って Control + C を押した可能性があります。

対処方法
　このコマンドを再度実行してください。

`cachefspack: *filename*- can't unpack file I/O error`

エラーの発生原因
　ハードウェアの障害の可能性があります。

対処方法
　ハードウェアの接続を確認してください。

`cachefspack: only one 'd', 'i', 'p', or 'u' option allowed`

エラーの発生原因
　コマンドに対して上記オプションのうち複数のオプションが指定されています。

対処方法
　オプションは 1 つだけ選択してください。

`cachefspack: can't find environment variable.`

エラーの発生原因
　構成ファイル中で $ で指定されている環境変数を設定していません。

対処方法
　環境変数を適切な場所に定義してください。

```
cachefspack: skipping LIST command - no active base
```

エラーの発生原因
LIST コマンドが構成ファイル内にありますが、対応する BASE コマンドがありません。

対処方法
BASE コマンドを定義してください。

CacheFS の統計情報の収集 (作業マップ)

次の作業マップは、CacheFS の統計情報の収集に関連する手順を示しています。これらの手順はすべてオプションです。

作業	説明	参照先
ロギングを設定します。	cachefslog コマンドを使用して、CacheFS ファイルシステムのロギングを設定します。	410 ページの「CacheFS ロギングを設定する方法」
ログファイルを検索します。	cachefslog コマンドを使用してログファイルの位置を特定します。	411 ページの「CacheFS ログファイルの場所を調べる方法」
ロギングを停止します。	cachefslog コマンドを使用してロギングを停止します。	412 ページの「CacheFS ロギングを停止する方法」
キャッシュサイズを表示します。	cachefswssize コマンドを使用してキャッシュサイズを表示します。	412 ページの「作業セット (キャッシュ) のサイズを表示する方法」
キャッシュ統計情報を表示します。	cachefsstat コマンドを使用して統計情報を表示します。	413 ページの「CacheFS 統計情報を表示する方法」

CacheFS の統計情報の収集

CacheFS の統計情報を収集すると、次の作業を行うことができます。

- 適切なキャッシュサイズを判断します。
- キャッシュのパフォーマンスを監視します。

これらの統計情報を使用すると、キャッシュサイズと望ましいパフォーマンスを選択して調整できます。

次の表に、CacheFS 統計コマンドを示します。

コマンド	説明	マニュアルページ
cachefslog	ログファイルの位置を指定します。統計情報が現在どこに記録されているかを表示し、ロギングを中止することもできます。	cachefslog(1M)
cachefswssize	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。	cachefswssize(1M)
cachefsstat	特定の CacheFS ファイルシステム、またはすべての CacheFS ファイルシステムに関する統計情報を表示します。このコマンドの出力に含まれる情報は、キャッシュから直接取り出されます。	cachefsstat(1M)

注 – CacheFS 統計情報コマンドは、どのディレクトリから実行してもかまいません。ただし、cachefswssize コマンドを実行するには、スーパーユーザーにならなければなりません。

CacheFS 統計の累計は、ログファイルの作成時から始まります。作業時間が終わったら、cachefslog -h コマンドを使用してロギングを停止してください。手順については、412 ページの「CacheFS ロギングを停止する方法」を参照してください。

CacheFS 統計情報コマンドを使用する前に、次の操作を実行する必要があります。

- cfsadmin コマンドを使用してキャッシュを設定します。
- 作成するログファイルに統計情報を収集できるように、適切な時間を決定します。この時間は、一般的な作業間隔に等しくする必要があります。たとえば、1 日、1 週間、1 カ月などにします。
- ログファイルの位置またはパスを選択します。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します。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ります。

注 – 次の手順は推奨する順序を示しています。異なる順序で作業してもかまいません。

▼ CacheFS ロギングを設定する方法

1 ロギングを設定します。

```
$ cachefslog -f log-file-path /mount-point
```

`-f` ロギングを設定します。

log-file-path ログファイルの位置を指定します。ログファイルは、vi などのエディタで作成する標準ファイルです。

/mount-point 統計情報を収集するマウントポイント (CacheFS ファイルシステム) を指定します。

2 ログファイルを正しく設定したかどうかを確認します。

```
$ cachefslog /mount-point
```

例 19–13 CacheFS ロギングを設定する

次の例は、`/var/tmp/samlog` ログファイルを設定して、`/home/sam` ディレクトリに関する統計情報を収集する方法を示しています。

```
$ cachefslog -f /var/tmp/samlog /home/sam
  /var/tmp/samlog: /home/sam
```

▼ CacheFS ログファイルの場所を調べる方法

- **CacheFS 統計情報のログが記録される場所を表示する**

  ```
  $ cachefslog /mount-point
  ```

  /mount-point は、統計情報を表示する CacheFS ファイルシステムを示します。

  オプションを指定せずに `cachefslog` コマンドを使用して、特定のマウントポイントに対するログファイルの場所を調べることもできます。

例 19–14 CacheFS ログファイルを検索する

次の例は、ログファイルが設定されている場合の表示を示します。ログファイルは、`/var/tmp/stufflog` にあります。

```
$ cachefslog /home/stuff
     /var/tmp/stufflog: /home/stuff
```

次の例は、指定したファイルシステムのログファイルが設定されていないことを示しています。

```
$ cachefslog /home/zap
    not logged: /home/zap
```

CacheFS ロギングを停止する方法

ロギングを停止するには、cachefslog -h オプションを使用します。

```
$ cachefslog -h /mount-point
```

次の例は、/home/stuff に対するロギングを停止する方法を示しています。

```
$ cachefslog -h /home/stuff
not logged: /home/stuff
```

上記の例と異なるシステム応答が表示される場合は、ロギングが正常に停止されていません。正しいログファイル名とマウントポイントを指定したかどうかを確認してください。

▼ 作業セット (キャッシュ) のサイズを表示する方法

キャッシュサイズを増やすべきかどうかを確認できます。または、特定のマウントポイントに関して前回 cachefslog コマンドを使用したあとの作業に基づいて、理想的なキャッシュサイズを決定することもできます。

1. クライアントシステムでスーパーユーザーになります。
2. 現在のキャッシュサイズとロギングされた最大キャッシュサイズを表示します。

   ```
   # cachefswssize log-file-path
   ```

   詳細は、cachefswssize(1M) のマニュアルページを参照してください。

例 19–15 作業セット (キャッシュ) のサイズを表示する

次の例で、「end size」とは cachefswssize コマンドを実行した時点のキャッシュサイズです。「high water size」とは、ロギングが発生した時間枠内のキャッシュの最大サイズです。

```
# cachefswssize /var/tmp/samlog

    /home/sam
           end size:  10688k
    high water size:  10704k

    /
           end size:   1736k
    high water size:   1736k

    /opt
           end size:    128k
    high water size:    128k
```

```
/nfs/saturn.dist
       end size:   1472k
high water size:   1472k

/data/abc
       end size:   7168k
high water size:   7168k

/nfs/venus.svr4
       end size:   4688k
high water size:   5000k

/data
       end size:   4992k
high water size:   4992k

total for cache
   initial size: 110960k
       end size:  30872k
high water size:  30872k
```

CacheFS の統計情報の表示

次の表では、出力された CacheFS ファイルシステムの統計情報の用語について解説します。

表 19–2　CacheFS 統計情報の用語

用語	説明
キャッシュのヒット率	キャッシュのヒット率対ミスヒット率の比と、それに続く実際のヒット数とミスヒット数。キャッシュヒットは、ユーザーがファイル操作を実行するときに、そのファイルが実際にキャッシュ内にあると発生します。キャッシュのミスヒットは、ファイルがキャッシュにないときに発生します。サーバーにかかる負荷は、キャッシュのミスヒット数、整合性検査数、および変更数の合計です。
整合性検査	実行された整合性検査の回数、合格回数、不合格回数。
変更数 (`modifies`)	変更操作の回数。書き込みや作成など。

▼ CacheFS 統計情報を表示する方法

`cachefsstat` コマンドを使用して統計情報を表示します。この操作はいつでも実行できます。たとえば、ロギングを設定しなくても統計情報を表示できます。

- **CacheFS 統計情報を表示する**

  `$` **`cachefsstat`** */mount-point*

/mount-point は、統計情報を表示する CacheFS ファイルシステムを示します。

マウントポイントを指定しなければ、マウントされているすべての CacheFS ファイルシステムに関する統計情報が表示されます。

例 19–16 CacheFS の統計情報の表示

この例は、キャッシュされたファイルシステム /home/sam に関する統計情報の表示方法を示しています。

```
$ cachefsstat /home/sam
        cache hit rate: 73% (1234 hits, 450 misses)
    consistency checks: 700 (650 pass, 50 fail)
             modifies: 321
garbage collection:  0
```

◆ ◆ ◆ 第 20 章

20 追加スワップ空間の構成 (手順)

この章では、Oracle Solaris OS をインストールしたあとで追加のスワップ空間を構成するためのガイドラインと手順について説明します。

この章で説明する手順は次のとおりです。

- 425 ページの「UFS ルート環境でスワップファイルを作成して使用可能にする方法」
- 427 ページの「Oracle Solaris ZFS ルート環境でスワップ空間を追加する方法」
- 428 ページの「不要になったスワップ空間を削除する方法」

この章の内容は以下のとおりです。

- 415 ページの「スワップ空間について」
- 419 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」
- 420 ページの「スワップ空間の割り当て方法」
- 420 ページの「スワップ空間の計画」
- 422 ページの「スワップリソースの監視」
- 424 ページの「スワップ空間の追加」

スワップ空間について

ユーザーは、次の事柄を決定する上で SunOS のスワップ機構を理解しておく必要があります。

- スワップ空間の要件
- スワップ空間と TMPFS ファイルシステムとの関係
- スワップ空間に関連するエラーメッセージから回復する方法

スワップ空間と仮想メモリー

Solaris ソフトウェアは、いくつかのディスクスライスを、ファイルシステムとしてではなく一時記憶域として使用します。これらのスライスを「スワップ」スライスと呼びます。スワップスライスは、システムの物理メモリーが不足し現在のプロセスを処理することができないときに、仮想メモリー記憶域として使用されます。

仮想メモリーシステムは、ディスク上のファイルの物理コピーをメモリー内の仮想アドレスに対応付けます。これらのマッピングに関するデータが入った物理メモリーページは、ファイルシステム内の通常ファイルまたはスワップ空間から読み直されます。メモリーをバックアップしているディスク空間に割り当てられる ID はわからないため、スワップ空間から読み直されたメモリーは「匿名メモリー」と呼ばれます。

Oracle Solaris OS 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、匿名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ります。
- SWAPFS という疑似ファイルシステムが、匿名メモリーページのアドレスを提供します。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ます。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ます。

スワップ空間と **TMPFS** ファイルシステム

Solaris 環境では、TMPFS ファイルシステムは `/etc/vfstab` ファイル内のエントリによって自動的に稼働されます。TMPFS ファイルシステムは、ファイルとそれに関連付けられた情報をディスクではなくメモリー (`/tmp` ディレクトリ) に格納するので、これらのファイルへのアクセスが高速になります。この機能によって、コンパイラや DBMS 製品のように `/tmp` の使用量の大きいアプリケーションの場合は、パフォーマンスが大幅に改善されます。

TMPFS ファイルシステムは、システムのスワップリソースから `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` ディレクトリ内の領域を使い果たすと、スワップ空間も使い果たしたことになります。したがって、`/tmp` ディレクトリの使用量が大きいアプリケーションの場合は、スワップ空間の使用状況を監視しなければ、システムがスワップ空間を使い果たす可能性があります。

スワップリソースが限られているときに TMPFS を使用する場合は、次の方法を使用してください。

- サイズオプション (`-o` *size*) を指定して TMPFS ファイルシステムをマウントし、TMPFS が使用できるスワップリソースを制御します。
- より大きな別のディレクトリをポイントするように、コンパイラの `TMPDIR` 環境変数の設定を変更します。

  コンパイラの `TMPDIR` 変数を使用すると、コンパイラが `/tmp` ディレクトリを使用するかどうかだけが制御されます。この変数は、ほかのプログラムによる `/tmp` ディレクトリの使用には影響しません。

ダンプデバイスとしてのスワップ空間

通常、ダンプデバイスとは、システムクラッシュダンプ情報を格納するために予約されているディスク領域のことです。デフォルトでは、UFS ルート環境内のスワップスライスがシステムのダンプデバイスとして構成されます。可能であれば、スワップパーティションを使用する代わりに、代替パーティションを「専用ダンプデバイス」として設定してください。クラッシュダンプの信頼性を高めたり、システム障害が発生したあとのリブート時間を短縮したりできます。専用ダンプデバイスの構成は、`dumpadm` コマンドで行えます。詳細は、『Solaris のシステム管理 (上級編)』の第 17 章「システムクラッシュ情報の管理 (手順)」を参照してください。

ZFS ルート環境では、スワップとダンプは別々の ZFS ボリュームとして構成されます。このモデルの利点は次のとおりです。

- スワップ領域とダンプ領域を含めるためにディスクをパーティションに分割する必要がない。
- スワップおよびダンプデバイスが背後の ZFS 入出力パイプラインアーキテクチャーを利用できる。
- スワップおよびダンプデバイス上で圧縮などの特性を設定できる。
- スワップおよびダンプデバイスのサイズをリセットできる。次に例を示します。

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME        PROPERTY  VALUE       SOURCE
rpool/dump  volsize   2G          -
```

  大規模ダンプデバイスを割り当てる処理には時間がかかることに注意してください。

- アクティブなシステムに 2 番目のスワップボリュームを追加して、全体のスワップサイズを増やすことを考慮できる。

ZFS スワップおよびダンプデバイスの使用方法の詳細については、『Oracle Solaris ZFS 管理ガイド』の「スワップデバイスおよびダンプデバイスの ZFS サポート」を参照してください。

Solaris ボリュームマネージャーなどのボリュームマネージャーを使用して UFS 環境内のディスクを管理している場合は、専用ダンプデバイスを Solaris ボリュームマネージャーの管理下に置かないように設定してください。スワップ領域は、Solaris ボリュームマネージャーの管理下に保管することをお勧めします。ただし、使いやすさとパフォーマンスの理由から、Solaris ボリュームマネージャーの管理下で動作しないディスクを専用ダンプデバイスとして設定してください。

スワップ空間と動的再構成

動的再構成時に CPU やシステムボードで障害が発生した場合に対応できるよう、十分なスワップ空間を確保することをお勧めします。スワップ空間が不足していると、CPU やシステムボードで障害が発生した際に、ホストまたはドメインはより少ないメモリーでリブートしなければなりません。

この追加スワップ空間を使用できない場合、メモリー不足のためアプリケーションの起動に失敗する可能性があります。この問題が発生した場合は、ユーザーが介入して、スワップ空間を追加するか、起動に失敗したアプリケーションのメモリー使用の設定を変更する必要があります。

リブート時のメモリー破損に備えて十分な追加スワップ空間が確保されていた場合、メモリーを大量に消費するアプリケーションのすべてが、通常どおりに起動します。したがって、スワップの多発によりシステムの動作は多少遅くなるにしても、ユーザーは通常どおりにシステムを利用できます。

詳細は、使用するハードウェアの動的構成マニュアルを参照してください。

SAN 環境でのスワップ空間の構成

SAN 環境などで、ネットワークで接続されたディスクにスワップ空間を構成するかどうかを判断するには、次の点を確認してください。

- ローカルに接続されたディスクのスワップ空間の問題を診断する方が、ネットワークで接続されたディスクのスワップ空間の問題を診断するより簡単です。
- SAN 経由のスワップ空間のパフォーマンスは、ローカルに接続されたディスクに構成されているスワップ空間と同等になります。
- システムのパフォーマンスに問題がある場合は、パフォーマンスデータを分析したあとでシステムにメモリーを追加する方が、ローカルに接続されたディスクにスワップを移動するよりも、SAN 経由のスワップのパフォーマンス問題をより適切に解決できる可能性があります。

スワップ空間の追加が必要かどうかを調べる方法

スワップ空間の追加が必要かどうかを調べるには、`swap -l` コマンドを使用します。

たとえば、次の `swap -l` コマンドの出力は、このシステムのスワップ空間がほぼ完全に使い果たされたか、割り当て率が 100% に達していることを示しています。

```
% swap -l
swapfile             dev   swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608    88
```

システムのスワップ空間の割り当て率が 100% に達すると、アプリケーションのメモリーページが一時的にロックしてしまいます。アプリケーションエラーが発生しない場合でも、システムパフォーマンスが低下する可能性があります。

システムにスワップ空間を追加する方法については、425 ページの「UFS ルート環境でスワップファイルを作成して使用可能にする方法」を参照してください。

スワップ関連のエラーメッセージ

次のメッセージは、アプリケーションに匿名メモリーを追加する必要がある場合に表示されます。ただし、バックアップ用のスワップ空間はもう余っていません。

application `is out of memory`

```
malloc error 0
```

```
messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

TMPFS 関連のエラーメッセージ

次のメッセージは、ファイルに書き込むときにページを割り当てることができない場合に表示されます。この問題は、TMPFS が許容限度を超えて書き込もうとしたときや、現在実行されているプログラムが大量のメモリーを使用している場合に発生することがあります。

directory`: File system full, swap space limit exceeded`

次のメッセージは、TMPFS が新しいファイルやディレクトリの作成中に物理メモリーを使い果たしたことを意味します。

directory`: File system full, memory allocation failed`

TMPFS 関連のエラーメッセージから回復する方法については、`tmpfs(7FS)` のマニュアルページを参照してください。

スワップ空間の割り当て方法

最初に、スワップ空間はSolarisインストールプロセスの一部として割り当てられます。インストールプログラムによるディスクスライスの自動レイアウトを選択し、かつスワップスライスのサイズを手作業で変更しない場合、Solarisインストールプログラムは UFS ルート環境にデフォルトのスワップ領域 (512M バイト) を割り当てます。ZFS ルート環境では、デフォルトのスワップサイズは通常 512M バイトから 2G バイトの範囲です。

スワップ空間の割り当て方法については、420 ページの「スワップ空間の計画」を参照してください。

UFS ルート環境にスワップファイルを作成することで、システムに追加のスワップ空間を割り当てることができます。スワップファイルは ZFS ルート環境ではサポートされていません。スワップファイルの作成方法については、424 ページの「スワップ空間の追加」を参照してください。

スワップ空間と /etc/vfstab ファイル

システムのインストールが完了すると、`/etc/vfstab` ファイルにスワップスライスとスワップファイルが一覧表示されます。スワップスライスとスワップファイルは、システムのブート時に `/sbin/swapadd` スクリプトによって有効になります。

`/etc/vfstab` ファイル内のスワップデバイスエントリには、次の情報が入っています。

- UFS ルートファイルシステムを備えるシステム上のスワップスライスまたはスワップファイルのフルパス名、あるいは ZFS ルートファイルシステムを備えるシステム上のスワップボリュームのパス名
- スワップスライスまたはスワップファイルのファイルシステムタイプ

スワップファイルが入っているファイルシステムは、スワップファイルが有効になる前にマウントしておかなければなりません。このため、`/etc/vfstab` ファイル内で、ファイルシステムをマウントするエントリが、スワップファイルを有効にするエントリより前に入っていることを確認してください。

スワップ空間の計画

スワップ空間のサイズを決定する上でもっとも重要な要素は、システムのソフトウェアアプリケーションの要件です。たとえば、コンピュータ支援設計シミュレータ、データベース管理製品、トランザクションモニター、地質分析システムなどの大型アプリケーションは、200 - 1000M バイトのスワップ空間を消費することがあります。

アプリケーションのスワップ空間の要件については、アプリケーションベンダーに問い合わせてください。

アプリケーションベンダーに問い合わせてもスワップ空間の要件を判断できない場合は、次のシステムタイプ別のガイドラインに従ってスワップ空間を割り当ててください。

システムタイプ	スワップ空間のサイズ	専用ダンプデバイスのサイズ
約 4G バイトの物理メモリーを備えたシステム	1G バイト	1G バイト
約 8G バイトの物理メモリーを備えた中型のサーバー	2G バイト	2G バイト
約 16 - 128G バイトの物理メモリーを備えたハイエンドサーバー	4G バイト	4G バイト
128G バイトを超える物理メモリーを備えたハイエンドサーバー	物理メモリーサイズの 1/4	物理メモリーサイズの 1/4

注 – クラッシュダンプの内容は圧縮されるため、ダンプデバイスを物理メモリーと同じサイズにする必要はありません。デフォルトでは、ダンプ内容の値はカーネルページに設定されます。ただし、すべてのメモリーページをダンプするようにダンプ内容の値が設定されている場合は、ダンプサイズを物理メモリーの半分以上に増やすことを検討してください。

UFS ベースのシステムでのスワップ空間の割り当て

上記の一般的なガイドラインのほかに、次の場合に UFS ベースのシステムにスワップ空間やディスク領域を割り当てることを検討してください。

- 専用ダンプデバイス。
- `/var/crash` ディレクトリ。デフォルトのシステムクラッシュダンプの内容は、カーネルメモリーページのみであり、ダンプは圧縮後に書き込まれます。複数のシステムクラッシュダンプを保持するのでない限り、このディレクトリのサイズをダンプデバイスのサイズに設定することを検討してください。

- 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します。次に、TMPFS によって使用される追加のスワップ空間を割り当てます。TMPFS については、416 ページの「スワップ空間と TMPFS ファイルシステム」を参照してください。

ZFS ベースのシステムでのスワップ空間の割り当て

ZFS ルートファイルシステムの初期インストール、または UFS ファイルシステムから ZFS ルートファイルシステムへの Solaris Live Upgrade の実行時に、スワップ領域が ZFS ルートプール内の ZFS ボリュームに、通常は 512M バイトから 2G バイトの範囲で自動的に作成されます。Solaris Live Upgrade を使用して UFS から ZFS に移行する場合は、既存のスワップ領域のサイズに基づいてスワップボリュームが作成されます。

ZFS ルートプールでは、スワップデバイスが固定サイズのスライスに事前に割り当てられることはないため、あとでスワップサイズを変更するのは非常に簡単です。

アプリケーションのスワップ要件を評価してから、初期インストール時またはインストール後に必要に応じて、デフォルトのスワップサイズを使用することも、スワップボリュームのサイズを調整することもできます。

初期インストール時のデフォルトのダンプボリュームサイズは、dumpadm の情報と物理メモリーのサイズに基づいて、カーネルにより計算されます。Live Upgrade を実行して UFS ルートファイルシステムから ZFS ルートファイルシステムに移行する際、デフォルトのダンプボリュームサイズが ZFS BE の物理メモリーの半分のサイズ (512M バイトから 2G バイト) に設定されます。

ZFS 環境では、ファイルシステムはプールの領域を消費するため、/var/crash ディレクトリは、保存するクラッシュダンプの数に応じて必要な容量を消費します。

スワップリソースの監視

/usr/sbin/swap コマンドを使用してスワップ領域を管理します。2 つのオプション -l と -s は、スワップリソースに関する情報を表示します。

swap -l コマンドを使用すると、システムのスワップ領域を確認できます。有効になっているスワップデバイスやファイルは、swapfile カラムの下に表示されます。次に例を示します。

```
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
```

ZFS ルートファイルシステムを備えるシステムでは、swap -l コマンドにより、ZFS ボリュームパス名が出力されることを除いて、類似の出力が得られます。次に例を示します。

```
# swap -l
swapfile                  dev  swaplo blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 1058800 1058800
```

swap -s コマンドを使用すると、スワップリソースを監視できます。

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

used 値と available 値の合計は、システム上の合計スワップ空間に等しくなります。これには、物理メモリーの一部とスワップデバイス (またはファイル) が含まれます。

使用可能なスワップ空間と使用済みスワップ空間の容量 (swap -s の出力内) を使用して、時間経過に伴うスワップ空間の使用状況を監視できます。システムパフォーマンスが適正であれば、swap -s を使用するとどの程度のスワップ空間が使用可能であるかがわかります。システムパフォーマンスが低下したときは、使用可能なスワップ空間の容量を検査して減少していないかどうかを調べてください。これによって、システムに対するどのような変更が原因でスワップ空間の使用量が増大したかを識別できます。

このコマンドを使用するときには、カーネルとユーザープロセスが物理メモリーをロックして解除するたびに、スワップに使用できる物理メモリーの容量が動的に変化するので注意してください。

注 – swap -l コマンドでは、スワップ空間が 512 バイトのブロック数として表示されます。swap -s コマンドでは、スワップ空間が 1024 バイトのブロック数として表示されます。swap -l で表示されたブロック数を合計して K バイトに変換すると、結果は「used + available」(swap -s の出力) より小さくなります。これは、swap -l では、スワップ空間の計算に物理メモリーが算入されないからです。

次の表に、swap -s コマンドの出力とその説明を示します。

表 20–1　swap -s コマンドの出力

キーワード	説明
bytes allocated	現在バックアップ用の記憶域 (ディスク上にとられたバックアップ用のスワップ空間) として使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数。

表 20–1 swap -s コマンドの出力 (続き)

キーワード	説明
reserved	現在は割り当てられていないが、あとから使用できるようにメモリーによって回収されるスワップ空間の合計容量を表す 1024 バイトのブロック数。
used	割り当て済みまたは予約済みのスワップ空間の合計容量を表す 1024 バイトのブロック数。
available	あとから予約や割り当てに使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数。

スワップ空間の追加

システム構成を変更して新しいソフトウェアパッケージをインストールした後に、スワップ空間を追加しなければならないことがあります。その場合に望ましいのは、mkfile コマンドと swap コマンドを使用して、既存の UFS または NFS ファイルシステムの一部を補助スワップ空間として指定する方法です。次の節で説明するように、これらのコマンドを使用すると、ディスクをパーティションに分割し直さなくても、スワップ空間を追加できます。

スワップ空間を追加するもう 1 つの方法は、既存のディスクをパーティションに分割し直すか、別のディスクを追加することです。ディスクのパーティション分割をやり直す方法については、第 10 章「ディスクの管理 (概要)」を参照してください。

UFS ルート環境でのスワップファイルの作成

UFS ルートファイルシステムで使用されるスワップファイルを作成できます。スワップファイルは ZFS ルート環境では現在サポートされていません。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します。
- swap コマンドを使用してスワップファイルを有効にします。
- 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 ファイルに追加します。

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、ファイルに 0 が埋め込まれます。スワップファイルのサイズは、バイト数 (デフォルト) として指定するか、接尾辞 k、b、m を使用して、それぞれ K バイト数、ブロック数、M バイト数として指定できます。

次の表に、mkfile コマンドのオプションを示します。

表 20–2 mkfile コマンドのオプション

オプション	説明
-n	空のファイルを作成します。サイズが表示されます。ただし、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ません。
-v	作成されたファイル名とサイズが表示されます。

注 – -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。

▼ UFS ルート環境でスワップファイルを作成して使用可能にする方法

1 スーパーユーザーになります。

root の権限がなくてもスワップファイルを作成できます。しかし、スワップファイルが誤って上書きされないように、root を所有者にしておくとよいでしょう。

2 必要に応じて、スワップファイル用のディレクトリを作成します。

3 スワップファイルを作成します。

```
# mkfile nnn[k|b|m] filename
```

指定したサイズ *nnn* (K バイト、バイト、または M バイト) とファイル名でスワップファイルが作成されます。

4 スワップファイルを有効にします。

```
# /usr/sbin/swap -a /path/filename
```

絶対パス名を使用してスワップファイルを指定しなければなりません。スワップファイルが追加され、ファイルシステムがマウント解除されるか、またはシステムがリブートされるまで使用可能になります。プロセスまたはプログラムがスワップファイルにスワップしているときは、ファイルシステムのマウントを解除できないことに注意してください。

5 ファイルのフルパス名を指定し、ファイルシステムのタイプとして swap を指定して、スワップファイルのエントリを /etc/vfstab ファイルに追加します。

```
/path/filename - - swap - no -
```

6 スワップファイルが追加されたことを確認するには、次のように入力します。

```
$ /usr/sbin/swap -l
```

注 – スワップファイルがアクティブにならない場合は、次のサービスが稼働していることを確認してください。

```
# svcs nfs/client
STATE          STIME    FMRI
enabled        14:14:34 svc:/network/nfs/client:default
```

例 20–1 UFS ルート環境でスワップファイルを作成して使用可能にする

次の例は、/files/swapfile という 100M バイトのスワップファイルを作成する方法を示しています。

```
# mkdir /files
# mkfile 100m /files/swapfile
# swap -a /files/swapfile
# vi /etc/vfstab
(An entry is added for the swap file):
/files/swapfile   -      -       swap     -     no     -
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
/files/swapfile        -       16 204784  204784
```

Oracle Solaris ZFS ルート環境でのスワップ空間の追加または変更

次の節では、ZFS ルート環境でスワップ空間を追加または変更する方法を説明します。システムまたはアプリケーションで追加のスワップ空間が必要かどうかを確認する方法については、前述の節を参照してください。

▼ Oracle Solaris ZFS ルート環境でスワップ空間を追加する方法

1 スーパーユーザーになります。

2 現在のスワップボリュームを確認します。

使用中のスワップボリュームは削除できません。現在のスワップボリュームが使用中かどうかは、blocks 列に示されるブロックと free 列に示されるブロックを比較して確認できます。2 つの列のブロックが等しい場合、スワップ領域は使用中ではありません。次に例を示します。

```
# swap -l
swapfile                 dev  swaplo   blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 1058800 1058800
```

3 次のいずれかを選択して、スワップボリュームのサイズを変更します。

a. 現在のスワップ領域が使用中でない場合、現在のスワップボリュームのサイズを変更できます。

次に例を示します。

```
# zfs get volsize rpool/swap
NAME        PROPERTY  VALUE    SOURCE
rpool/swap  volsize   517M     -
# zfs set volsize=2g rpool/swap
# zfs get volsize rpool/swap
NAME        PROPERTY  VALUE    SOURCE
rpool/swap  volsize   2G       -
```

b. 現在のスワップ領域が使用中の場合は、別のスワップボリュームを追加できます。

次に例を示します。

```
# zfs create -V 2G rpool/swap2
```

c. 2 番目のスワップボリュームを有効にします。

次に例を示します。

```
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile                  dev  swaplo   blocks   free
/dev/zvol/dsk/rpool/swap  256,1      16 1058800 1058800
/dev/zvol/dsk/rpool/swap2 256,3      16 4194288 4194288
```

スワップファイルの削除

不要になったスワップ空間は、削除できます。

▼ 不要になったスワップ空間を削除する方法

1. スーパーユーザーになります。
2. スワップ空間を削除します。

   # /usr/sbin/swap -d */path/filename*

   スワップファイル名が削除されるので、スワッピングに使用できなくなります。ファイルそのものは削除されません。
3. `/etc/vfstab` ファイルを編集して、スワップファイルのエントリを削除します。
4. ほかの目的に使用できるようにディスク領域を復元します。

   # rm */path/filename*

   スワップ空間がファイルの場合は削除します。また、スワップ空間が別のスライスに入っていて、不要なことがわかっている場合は、新しいファイルシステムを作成してマウントします。

   ファイルシステムのマウント方法については、第 18 章「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。
5. 削除したスワップファイルが使用できなくなっていることを確認します。

   # swap -l

例 20–2 不要になったスワップ空間を削除する

次の例は、`/files/swapfile` スワップファイルを削除する方法を示しています。

```
# swap -d /files/swapfile
# (Remove the swap entry from the /etc/vfstab file)
# rm /files/swapfile
# swap -l
swapfile             dev  swaplo  blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
```

◆ ◆ ◆ 第 21 章

UFS ファイルシステムの整合性検査 (手順)

この章では、UFS ファイルシステムの整合性検査に関する概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 439 ページの「代替ブートデバイスからルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムを検査する方法」
- 441 ページの「ルート (/)、/usr、または /var 以外のファイルシステムを検査する方法」
- 444 ページの「UFS ファイルシステムを修復する方法」
- 449 ページの「不正なスーパーブロックを復元する方法 (Solaris 8、9、および 10 リリース)」
- 446 ページの「不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース)」

この章の内容は以下のとおりです。

- 430 ページの「ファイルシステムの整合性」
- 430 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」
- 431 ページの「fsck コマンドで検査して修復される内容」
- 438 ページの「UFS ファイルシステムを対話式で検査して修復する」
- 445 ページの「不正なスーパーブロックの復元」
- 450 ページの「fsck コマンドの構文とオプション」

Solaris 10 6/06 リリースの fsck に関する最新情報については、321 ページの「UFS ファイルシステムユーティリティー (fsck、mkfs、および newfs) の拡張機能」を参照してください。

fsck のエラーメッセージについては、『Solaris のシステム管理 (上級編)』の第 20 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

この章で取り上げる UFS ファイルシステム構造のバックグラウンド情報については、第 22 章「UFS ファイルシステム (参照情報)」を参照してください。

ファイルシステムの整合性

UFS ファイルシステムは、一連の内部テーブルを基にして使用済み i ノード、使用可能ブロックを特定します。これらの内部テーブルがディスク上のデータと正しく同期していないと、整合性が失われ、ファイルシステムの修復が必要になります。

次のような原因でオペレーティングシステムが異常終了すると、ファイルシステムの整合性が失われることがあります。

- 電源障害
- 不注意によるシステム電源の切断
- 正しいシャットダウン手順以外の方法によるシステム電源の切断
- カーネル内のソフトウェアエラー

ファイルシステムの整合性が失われることは重大ですが、頻繁に起きるものではありません。システムをブートすると、ファイルシステムの整合性検査が (`fsck` コマンドを使用して) 自動的に実行されます。ほとんどの場合は、このファイルシステムの検査によって問題が修復されます。

`fsck` コマンドは、ファイルシステム上に配置されているが参照不可能なファイルとディレクトリを `lost+found` ディレクトリに入れます。参照不可能なファイルとディレクトリの名前として i ノード番号が割り当てられます。`lost+found` ディレクトリが存在しない場合は、`fsck` コマンドによって作成されます。`lost+found` ディレクトリ内の領域が足りない場合は、`fsck` コマンドによってそのサイズが拡張されます。

i ノードについては、454 ページの「i ノード」を参照してください。

ファイルシステムの状態はどのように記録されるか

`fsck` コマンドは、スーパーブロックに保存される状態タグを使用してファイルシステムの状況を記録します。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検査する必要があるか判断します。このフラグは、ブート時に `/sbin/rcS` スクリプトによって使用されるか、`fsck -m` コマンドによって使用されます。`fsck -m` コマンドの結果を無視する場合は、状態フラグの設定に関係なく、すべてのファイルシステムを検査できます。

スーパーブロックについては、454 ページの「スーパーブロック」を参照してください。

次の表に、状態フラグが取り得る値とその説明を示します。

表 21–1 ファイルシステムの状態フラグの値

状態フラグの値	説明
FSACTIVE	このファイルシステムがマウント済みであり、メモリー内のデータが変更されたことを示します。この状態フラグを持つマウント済みファイルシステムでは、システムの電源が切断されると、ユーザーデータまたはメタデータが失われます。
FSBAD	このファイルシステムに不整合なファイルシステムデータが含まれていることを示します。
FSCLEAN	このファイルシステムが正常にマウント解除されており、破損していないことを示します。
FSLOG	このファイルシステムのロギングが有効になっていることを示します。このフラグが設定されたファイルシステムは、マウントされている場合も、マウント解除されている場合もあります。ロギングが有効になっているファイルシステムに設定できるフラグは、FSLOG、FSBAD のいずれかです。ロギングが無効になっているファイルシステムに設定できるフラグは、FSACTIVE、FSSTABLE、FSCLEAN のいずれかです。
FSSTABLE	このファイルシステムがマウント済みであり、アイドル状態であることを示します。この状態フラグを持つマウント済みファイルシステムでは、システムの電源が切断されても、ユーザーデータやメタデータが失われることはありません。

fsck コマンドで検査して修復される内容

この節では、ファイルシステムの通常の処理中に発生する問題、原因、fsck コマンド (検査および修復ユーティリティー) 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。

UFS ファイルシステムの不整合が発生する理由

就業日には毎日多数のファイルが作成、変更、または削除されます。ファイルが変更されるたびに、オペレーティングシステムは一連のファイルシステムの更新処理を実行します。これらの更新処理がディスクに確実に書き込まれると、ファイルシステムの整合性が保たれます。

ユーザープログラムが書き込みなどの、ファイルシステムを変更する処理を実行すると、書き込まれるデータはまずカーネルのインコアバッファーにコピーされます。一般に、ディスクの更新は非同期に処理されます。このため、ユーザープロセスは、書き込みシステムコールが値を返したあとすぐに処理を続けることができますが、実際のデータの書き込みは、ずいぶんあとに実行されることもあります。したがって、ディスク上にあるファイルシステムは、インコア情報で表されるファイルシステムの状態から常に遅延することになります。

別の目的にバッファーが必要になったり、カーネルが fsflush デーモンを自動的に (30 秒間隔で) 実行すると、インコア情報を反映するようにディスク情報が更新されます。システムがインコア情報を書き込まずに停止すると、ディスク上のファイルシステムの整合性がなくなります。

ファイルシステムの整合性は、さまざまな原因で失われることがあります。もっとも一般的な原因は、オペレータのエラーとハードウェア障害です。

システムを正しくシャットダウンしなかったり、マウントされているファイルシステムが正しくオフラインにされないと、「クリーンでない停止」が原因で問題が発生することがあります。クリーンでないシャットダウンを防ぐには、システムをシャットダウンしたり、ディスクをドライブから物理的に取り出したり、ディスクをオフライン状態にしたりする前に、ファイルシステムの現在の状態をディスクに書き込まなければなりません。つまり、「同期」させなければなりません。

このほか、ハードウェアの欠陥や、ディスクまたはコントローラのファームウェアの問題が原因で整合性が失われることもあります。ディスクドライブ上ではいつでもブロックが損傷を受ける可能性があります。あるいは、ディスクコントローラが正常に機能しなくなることもあります。

整合性が検査される UFS 構成要素

この節では、UFS ファイルシステムの構成要素、つまりスーパーブロック、シリンダグループブロック、i ノード、間接ブロック、データブロックに fsck コマンドが適用する整合性検査の種類について説明します。

UFS ファイルシステム構造については、453 ページの「UFS ファイルシステムのシリンダグループの構造」を参照してください。

スーパーブロックの検査

スーパーブロックには集計情報が格納されており、UFS ファイルシステム内でもっとも破損しがちな構成要素です。ファイルシステムの i ノードやデータブロックが変更されるたびに、スーパーブロックも変更されます。CPU が停止した場合、直前のコマンドが sync コマンドでなければ、スーパーブロックはほぼ確実に破損しています。

スーパーブロックの不整合は、次の面から検査されます。

- ファイルシステムのサイズ
- i ノード数
- 空きブロック数
- 空き i ノード数

ファイルシステムのサイズと i ノードリストのサイズの検査

ファイルシステムのサイズは、スーパーブロックと i ノードリストに使用されるブロック数よりも大きくなければなりません。i ノード数は、ファイルシステムの最大許容数よりも小さくなければなりません。i ノードは、ファイルに関するすべての情報を表します。ファイルシステムのサイズとレイアウト情報は、fsck コマンドにとってもっとも重要な情報部分です。これらのサイズはファイルシステムの作成時に静的に決められるため、実際に検査する方法はありません。ただし、fsck コマンドを使用してサイズが妥当な範囲内にあるかどうかは検査できます。ファイルシステムのほかのすべての検査を行うには、これらのサイズが正確でなければなりません。fsck コマンドが一次スーパーブロックの静的パラメータ内に不正な情報を検出すると、オペレータに代替スーパーブロックの位置を指定するように促します。

UFS ファイルシステム構造の詳細は、453 ページの「UFS ファイルシステムのシリンダグループの構造」を参照してください。

空きブロック数の検査

空きブロック数は、シリンダグループのブロックマップに格納されます。fsck コマンドは、空きマーク付きのすべてのブロックがファイルによって使用されていないかどうかを検査します。すべてのブロックを検査し終わると、fsck コマンドは空きブロック数と i ノードによって使用されるブロック数の合計がファイルシステム内の合計ブロック数に等しくなるかどうかを検査します。ブロックマップ内に間違いがあると、fsck コマンドはブロックが割り当てられている状態のままで構築し直します。

スーパーブロック内の集計情報には、ファイルシステム内の空きブロックの合計数のカウントが入っています。fsck コマンドは、このブロック数をファイルシステム内で見つかった空きブロック数と比較します。数が一致しなければ、fsck コマンドはスーパーブロック内の空きブロック数を実際の空きブロック数で置き換えます。

空き i ノード数の検査

スーパーブロック内の集計情報には、ファイルシステム内の空き i ノード数が入っています。fsck コマンドは、この i ノード数をファイルシステム内で見つかった空き i ノード数と比較します。数が一致しなければ、fsck はスーパーブロック内の空き i ノード数を実際の空き i ノード数で置き換えます。

i ノード

i ノードリストは、i ノード 2 から順番に検査されます。(i ノード 0 と i ノード 1 は予約ノード)。各 i ノードの不整合は、次の面から検査されます。

- 形式とタイプ
- リンク数
- 重複ブロック

- 不正なブロック番号
- i ノードのサイズ

i ノードの形式とタイプ

各 i ノードには、そのタイプと状態を記述するモードのワードが入っています。i ノードには、次の 9 つのタイプがあります。

- 通常
- ディレクトリ
- ブロック型特殊ファイル
- キャラクタ型特殊ファイル
- FIFO (名前付きパイプ)
- シンボリックリンク
- シャドウ (ACL で使用される)
- 属性ディレクトリ
- ソケット

i ノードの状態は、次の 3 つのうちのいずれかです。

- 割り当て済み
- 未割り当て
- 部分的に割り当て済み

ファイルシステムの作成時、決まった数の i ノードは無効になっています。これらは、必要なときが来るまで割り当てられません。「割り当て済みの i ノード」とは、ファイルを指す i ノードです。「未割り当ての i ノード」は、ファイルを指さないので空のはずです。「部分的に割り当て済み」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 ノードリストに不正なデータが書き込まれると、i ノードは「部分的に割り当て済み」の状態になることがあります。`fsck` コマンドが実行できる唯一の修正動作は、その i ノードを消去することです。

リンク数の検査

各 i ノードには、そこにリンクされているディレクトリエントリ数が入っています。`fsck` コマンドは、ルート (*/*) ディレクトリから順番にディレクトリ構造全体を検査し、i ノードごとに実際のリンク数を計算して、各 i ノードのリンク数を検証します。

i ノードに格納されているリンク数が `fsck` コマンドによって判断された実際のリンク数と一致しない場合は、次の 3 つの状況が考えられます。

- 格納されたリンク数が 0 でなく、実際のリンク数が 0 の場合。

  この状況は、i ノードにリンクされているディレクトリエントリが存在しない場合に発生することがあります。この場合、`fsck` コマンドはリンクされていないファイルを `lost+found` ディレクトリに入れます。

- 格納されたリンク数が 0 でなく、実際のリンク数も 0 でない。しかし、それらの数が等しくない場合。

  この状況は、ディレクトリエントリが追加または削除されたが、i ノードが更新されていない場合に発生することがあります。この場合、fsck コマンドは格納されたリンク数を実際のリンク数で置き換えます。
- 格納されたリンク数が 0 で実際のリンク数が 0 でない場合。

  この場合、fsck コマンドは i ノードのリンク数を実際のリンク数に変更します。

重複ブロックの検査

各 i ノードには、それが使用するすべてのブロックのリスト、またはリストを指すポインタ (間接ブロック) が入っています。間接ブロックは i ノードによって所有されるので、間接ブロックの整合性が失われると、それを所有する i ノードが直接影響を受けます。

fsck コマンドは、i ノードから使用される各ブロック番号を、割り当て済みブロックのリストと比較します。別の i ノードからすでにブロック番号が使用されていると、そのブロック番号は重複ブロックのリストに入れられます。それ以外の場合は、割り当て済みブロックのリストが更新され、ブロック番号が追加されます。

重複ブロックが見つかると、fsck コマンドは再び i ノードリストを調べて、各重複ブロックを使用するほかの i ノードを検索します。fsck コマンドでは、どの i ノードでエラーが発生しているか、正確に判断することはできません。このため、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が表示されます。i ノード内に多数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書き込まれていない間接ブロックの影響を受けている可能性があります。

不正なブロック番号の検査

fsck コマンドは、i ノードから使用される各ブロック番号を検査して、その値がファイルシステム内の最初のデータブロック番号の値よりも大きく、最後のデータブロック番号の値より小さいかどうかを調べます。ブロック番号がこの範囲に含まれない場合は、不正なブロック番号と見なされます。

間接ブロックがファイルシステムに正しく書き込まれていないことが原因で、i ノード内に不正なブロック番号が発見されることがあります。fsck コマンドはその i ノードの消去を促すプロンプトを表示します。

i ノードサイズの検査

各 i ノードには、参照するデータブロック数が入っています。実際のデータブロック数は、割り当て済みのデータブロック数と間接ブロック数の合計です。fsck コマンドはデータブロック数を計算し、そのブロック数を i ノードから使用されるブロック数と比較します。i ノードに不正なブロック数が入っていると、fsck コマンドはその修正を促すプロンプトを表示します。

各 i ノードには、64 ビットのサイズフィールドがあります。このフィールドは、i ノードに関連付けられたファイル内の文字数 (データバイト数) を示します。i ノードのサイズフィールドに整合性があるかどうかは、サイズフィールド内の文字数を使用して、i ノードに関連付けるべきブロック数を計算し、その結果を i ノードから使用される実際のブロック数と比較して概算で検査されます。

間接ブロック

間接ブロックは i ノードによって所有されます。したがって、間接ブロック内の整合性が失われると、それを所有する i ノードが影響を受けます。不整合は、次の面から検査されます。

- すでに別の i ノードから使用されているブロック
- ファイルシステムの範囲に含まれないブロック番号

直接ブロックに対しても整合性検査が実行されます。

データブロック

i ノードは、3 種類のデータブロックを直接または間接に参照できます。参照されるブロックは、すべて同じ種類でなければなりません。次の 3 種類のデータブロックがあります。

- プレーンデータブロック
- シンボリックリンクデータブロック
- ディレクトリデータブロック

プレーンデータブロックには、ファイルに格納される情報が入っています。シンボリックリンクデータブロックには、シンボリックリンクに格納されるパス名が入っています。ディレクトリデータブロックには、ディレクトリエントリが入っています。`fsck` コマンドはディレクトリデータブロックの妥当性しか検査できません。

ディレクトリは、i ノードの `mode` フィールド内のエントリによって通常ファイルと区別されます。ディレクトリに関連付けられたデータブロックには、ディレクトリエントリが入っています。ディレクトリデータブロックの不整合は、次の面から検査されます。

- 未割り当ての i ノードを指すディレクトリ内の i ノード番号
- ファイルシステム内の i ノード番号より大きいディレクトリ内の i ノード番号
- 「.」と「..」ディレクトリには許されないディレクトリ内の i ノード番号
- ファイルシステムから切り離されたディレクトリ

未割り当てディレクトリの検査

ディレクトリデータブロック内の i ノード番号が未割り当て i ノードを指す場合、`fsck` コマンドはそのディレクトリエントリを削除します。この状況は、新しい

ディレクトリエントリが入っているデータブロックが変更されて書き出されたが、i ノードが書き込まれていない場合に発生します。また、警告なしに CPU が停止された場合にも発生します。

不正な i ノード番号の検査

ディレクトリエントリの i ノード番号が i ノードリストの最後を超える位置を指す場合、fsck コマンドはそのディレクトリエントリを削除します。この状況は、不正なデータがディレクトリのデータブロックに書き込まれると発生します。

不正な「.」と「..」エントリのチェック

「.」ディレクトリの i ノード番号のエントリは、ディレクトリデータブロックの最初のエントリでなければなりません。また、それ自体を参照しなければなりません。つまり、その値はディレクトリデータブロックの i ノード番号に等しくなければなりません。

「..」ディレクトリの i ノード番号は、ディレクトリデータブロックの第 2 のエントリでなければなりません。その値は、親ディレクトリの i ノード番号 (または、ディレクトリがルート (/) ディレクトリの場合は、それ自体の i ノード番号) に等しくなければなりません。

「.」と「..」ディレクトリの i ノード番号が不正であれば、fsck コマンドはそれらを正しい値で置き換えます。ディレクトリへのハードリンクが複数個ある場合は、最初に見つかったハードリンクが「..」が指す実際の親であると見なされます。この場合、fsck コマンドはほかの名前を削除するように促すプロンプトを表示します。

切り離されたディレクトリ

fsck コマンドは、ファイルシステム全体で参照関係を検査します。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ると、fsck コマンドはそのディレクトリをファイルシステムの lost+found ディレクトリにリンクします。この条件は、ファイルシステムに i ノードは書き込まれたが、対応するディレクトリデータブロックは書き込まれていない場合に発生する可能性があります。

通常データブロック

通常ファイルに関連付けられたデータブロックには、ファイルの内容が入っています。fsck コマンドは、通常ファイルのデータブロックの内容が有効かどうかは検査しません。

fsck 要約メッセージ

fsck コマンドを対話式で実行して正常に終了すると、次のようなメッセージが表示されます。

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)
#
```

fsck 出力の最後の行は、ファイルシステムについて次のような情報を記述します。

# files	使用中の i ノード数
# used	使用中のフラグメント数
# free	未使用のフラグメント数
# frags	未使用の非ブロックフラグメント数
# blocks	未使用の完全ブロック数
% fragmentation	断片化の比率。ファイルシステム内の空きフラグメント × 100 / 全フラグメント

フラグメントについては、458 ページの「フラグメントサイズ」を参照してください。

UFS ファイルシステムを対話式で検査して修復する

次の場合には、ファイルシステムを対話式で検査する必要があります。

- マウントできない場合
- 使用中に不整合が発生する場合

使用中のファイルシステムの整合性が失われると、コンソールウィンドウやシステムメッセージファイルにエラーメッセージが出力されたり、システムがクラッシュしたりすることがあります。たとえば、システムメッセージファイル /var/adm/messages に次のようなメッセージが出力されることがあります。

```
Sep  5 13:42:40 hostname ufs: [ID 879645 kern.notice] NOTICE: /: unexpected
free inode 630916, run fsck(1M)
```

hostname は、エラーを報告したシステムです。

fsck コマンドを実行する前に、これらの参照情報を利用して、fsck エラーメッセージの内容を解決する場合があります。

- 450 ページの「fsck コマンドの構文とオプション」

- 『Solaris のシステム管理 (上級編)』の第 20 章「UFS ファイルシステムの不整合解決 (手順)」

fsck コマンドを実行して UFS ファイルシステムを検査するときには、次の点を考慮してください。

- fsck を使用して検査するファイルシステムは、非アクティブでなければなりません。ディスクにまだフラッシュされていないファイルシステムの変更がある場合、または fsck による検査の進行中にファイルシステムを変更した場合、ファイルシステムの破損とみなされることがあります。この場合、問題の報告内容に信頼性があるとは言えません。
- fsck を使用して修復するファイルシステムは、非アクティブでなければなりません。ディスクにまだフラッシュされていないファイルシステムの変更がある場合、または fsck による修復の進行中にファイルシステムを変更した場合、ファイルシステムが破損したり、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、そのファイルシステムのマウントを解除します。これにより、ファイルシステムのデータ構造の整合性をできるだけ確保できます。ただし、ルート (/) ファイルシステム、/usr ファイルシステム、および /var ファイルシステムだけは、アクティブにします。これらのファイルシステムは、fsck を実行するためにマウントする必要があります。
- ルート (/) ファイルシステム、/usr ファイルシステム、および /var ファイルシステムを修復する必要がある場合には、可能であれば代替デバイスからシステムをブートすると、マウント解除されて非アクティブになります。

  ルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムに対して fsck を実行する手順については、439 ページの「代替ブートデバイスからルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムを検査する方法」を参照してください。

▼ 代替ブートデバイスからルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムを検査する方法

Solaris 10 6/06 リリースの fsck に関する最新情報については、321 ページの「UFS ファイルシステムユーティリティー (fsck、mkfs、および newfs) の拡張機能」を参照してください。次のメッセージが表示される場合は、fsck を再実行する必要はありません。

```
***** FILE SYSTEM WAS MODIFIED *****
```

ただし、このメッセージのあとに fsck を再実行してもファイルシステムは損傷しません。このメッセージは、fsck の修正処置に関する情報を示すだけのものです。

この手順では、ローカルの Solaris DVD またはネットワークブートサーバーを利用でき、代替デバイスからシステムをブートできることを前提とします。

不正なスーパーブロックの復元については、446 ページの「不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース)」または 449 ページの「不正なスーパーブロックを復元する方法 (Solaris 8、9、および 10 リリース)」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 ルート (/) ファイルシステムがミラー化されているシステムの場合のみ: ルート (/) ミラーを切り離したあと、代替デバイスからブートします。先にブートしてしまうと、ファイルシステムが破損するおそれがあります。

ルート (/) ミラーの切り離しについては、『Solaris ボリュームマネージャの管理』の「サブミラーに関する作業」を参照してください。

3 検査の必要なルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムのデバイス (/dev/dsk/c0t0d0s0 など) を特定します。

このデバイス名は、代替デバイスからブートしたあとで指定する必要があります。代替デバイスからブートしたあとでは、このデバイスの特定がより難しくなります。

4 ローカルの **Solaris DVD** やネットワークなどの代替デバイスから検査する必要があるルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムのあるシステムを、シングルユーザーモードでブートします。

この場合、ファイルシステム上に動作は存在しません。

次に例を示します。

```
# init 0
ok boot net -s
.
.
.
#
```

5 手順 **3** で特定したルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムが含まれるデバイスを検査します。

検査または修復するファイルシステムのハードウェアが変更されている場合には、そのデバイス名も変更されている可能性があります。fsck -n メッセージの「Last Mounted on ...」を利用して、そのファイルシステムのデバイスが予期したデバイスであることを確認してください。

たとえば、検査するルート (/) ファイルシステムが /dev/dsk/c0t0d0s0 の場合は、次のように出力されます。

```
# fsck -n /dev/rdsk/c0t0d0s0
** /dev/rdsk/c0t0d0s0 (NO WRITE)
```

```
** Last Mounted on /
.
.
.
fsck /dev/rdsk/c0t0d0s0
** /dev/rdsk/c0t0d0s0
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
.
.
.
```

6 報告された fsck エラーをすべて修正します。

1 つまたは複数の UFS ファイルシステムを対話式で検査しながら、エラーメッセージのプロンプトに応答する方法については、『Solaris のシステム管理 (上級編)』の第 20 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

7 fsck コマンドを実行してもすべての問題を修復できない場合は、444 ページの「fsck コマンドで修復できない UFS ファイルシステムの修正」を参照してください。

8 修復したファイルシステムをマウントして、lost+found ディレクトリにファイルが存在するかどうかを確認します。

fsck コマンドによって lost+found ディレクトリに入れられた各ファイルの名前は、その i ノード番号を使用して変更されます。可能であれば、ファイル名を変更し、ファイルが含まれるべきディレクトリに移動してください。grep コマンドを使用して各ファイル内の語句を探したり、file コマンドを使用してファイルタイプを識別してください。

どうしても特定できないファイルまたはディレクトリについては、 lost+found ディレクトリからそれらを削除して、不要な領域を解放してください。

9 システムをマルチユーザーモードに戻します。

```
# init 6
```

10 ルート (/) ファイルシステムがミラー化されているシステムの場合のみ: ルート (/) ミラーを再接続します。

▼ ルート (/)、/usr、または /var 以外のファイルシステムを検査する方法

Solaris 10 6/06 リリースの fsck に関する最新情報については、321 ページの「UFS ファイルシステムユーティリティー (fsck、mkfs、および newfs) の拡張機能」を参照してください。次のメッセージが表示される場合は、fsck を再実行する必要はありません。

```
***** FILE SYSTEM WAS MODIFIED *****
```

ただし、このメッセージのあとに fsck を再実行してもファイルシステムは損傷しません。このメッセージは、fsck の修正処置に関する情報を示すだけのものです。

この手順では、検査するファイルシステムがマウント解除されていることを前提としています。

不正なスーパーブロックの復元については、446 ページの「不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース)」または 449 ページの「不正なスーパーブロックを復元する方法 (Solaris 8、9、および 10 リリース)」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. ローカルファイルシステムをマウント解除し、ファイルシステム上でほかの動作が存在しないようにします。

   fsck コマンドの引数として、マウントポイントディレクトリや /dev/dsk/ *device-name* を指定します。整合性が失われている場合には、そのことを示すメッセージが表示されます。

   次に例を示します。

   ```
   # umount /export/home
   # fsck /dev/rdsk/c0t0d0s7
   ** /dev/dsk/c0t0d0s7
   ** Last Mounted on /export/home
   .
   .
   .
   ```

3. 報告された fsck エラーをすべて修正します。

   1 つまたは複数の UFS ファイルシステムを対話式で検査しながら、エラーメッセージのプロンプトに応答する方法については、『Solaris のシステム管理 (上級編)』の第 20 章「UFS ファイルシステムの不整合解決 (手順)」を参照してください。

4. fsck コマンドを実行してもすべての問題を修復できない場合は、**444** ページの「fsck コマンドで修復できない **UFS** ファイルシステムの修正」を参照してください。

5. 修復したファイルシステムをマウントして、 lost+found ディレクトリにファイルが存在するかどうかを確認します。

   fsck コマンドによって lost+found ディレクトリに入れられた各ファイルの名前は、その i ノード番号を使用して変更されます。

6 lost+found ディレクトリに保存されているファイルの名前を変更して移動します。

可能であれば、ファイル名を変更し、ファイルが含まれるべきディレクトリに移動してください。grep コマンドを使用して各ファイル内の語句を探したり、file コマンドを使用してファイルタイプを識別してください。

どうしても特定できないファイルまたはディレクトリについては、 lost+found ディレクトリからそれらを削除して、不要な領域を解放してください。

例 21–1 ルート (/) 以外、または /usr 以外のファイルシステムの対話型検査

次の例は、/dev/rdsk/c0t0d0s6 ファイルシステムを検査し、不正なブロック数を訂正する方法を示しています。この例では、ファイルシステムがマウント解除されていることを前提としています。

```
# fsck /dev/rdsk/c0t0d0s6
** Phase 1 - Check Block and Sizes
INCORRECT BLOCK COUNT I=2529 (6 should be 2)
CORRECT? y

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Cylinder Groups
929 files, 8928 used, 2851 free (75 frags, 347 blocks, 0.6%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
#
```

UFS ファイルシステムの修復

fsck -o p コマンド (p は preen (修復) を表す) は、UFS ファイルシステムを検査し、通常は予期しないシステムのシャットダウンによって発生する問題を自動的に修正します。オペレータの介入が必要な問題が発見されると、このコマンドは即座に終了します。このコマンドによって、ファイルシステムを並列に検査することも可能です。

システムがクリーンな状態でシャットダウンしなかった後のファイルシステムの修復にも、fsck -o p コマンドを実行できます。このモードでは、fsck コマンドはクリーンフラグを調べずに完全検査を実行します。これらの処理は、fsck コマンドを対話式で実行した場合の処理のサブセットです。

▼ UFS ファイルシステムを修復する方法

この手順では、ファイルシステムがマウント解除されているか、非アクティブであることを前提としています。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. UFS ファイルシステムをマウント解除します。

   ```
   # umount /mount-point
   ```

3. 修復オプションを指定して UFS ファイルシステムを検査します。

   ```
   # fsck -o p /dev/rdsk/device-name
   ```

   fsck コマンドの引数として */mount-point* または /dev/rdsk/*device-name* を使用すると、個々のファイルシステムを修復できます。

例 21–2 UFS ファイルシステムを修復する

次の例は、/export/home ファイルシステムの修復方法を示します。

```
# fsck -o p /export/home
```

fsck コマンドで修復できない UFS ファイルシステムの修正

fsck コマンドを何度か実行しているとき、問題の修正を進めていく過程で、以前の実行で検出されていた問題が再度発生することがあります。場合によっては、問題が報告されなくなるまで、繰り返し fsck を実行する必要があります。これにより、すべてのエラーが検出され、修復されます。

fsck コマンドで表示される情報に注目してください。問題を解決する上で参考になります。たとえば、メッセージは損傷したディレクトリを指している場合があります。そのディレクトリを削除すると、fsck コマンドが問題なく実行されるようになる場合もあります。

それでも fsck コマンドでファイルシステムを修復できない場合は、ff、clri、および ncheck コマンドを使用し、間違いを特定して修正します。これらのコマンドの使用方法については、次の情報を参照してください。

- fsdb(1M)
- ff(1M)
- clri(1M)
- ncheck(1M)

最終的には、ファイルシステムを作成し直し、その内容をバックアップメディアから復元せざるをえない場合があります。

完全なファイルシステムを復元する方法については、第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。

ファイルシステムを完全に修復できないが、読み取り専用としてマウントできる場合は、`cp`、`tar`、または `cpio` コマンドを使用して、データのすべてまたは一部をファイルシステムから取り出してください。

問題の原因がハードウェア上のディスクエラーであれば、ファイルシステムを作成し直して復元する前に、ディスクをフォーマットし直して再びパーティション分割を行わなければならない場合があります。ディスクデバイスを交換する前に、デバイスのケーブルおよびコネクタが正常に機能するかどうかを検査してください。一般に、ハードウェアエラーが発生すると、さまざまなコマンドで同じエラーが繰り返し表示されます。`format` コマンドはディスク上の不良ブロックを使用しないようにします。ただし、ディスクの損傷が致命的な場合、フォーマットし直したあとも問題が解決されないことがあります。`format` コマンドの使用方法については、`format(1M)` のマニュアルページを参照してください。新しいディスクのインストール方法については、第 12 章「SPARC: ディスクの追加 (手順)」または第 13 章「x86: ディスクの追加 (手順)」を参照してください。

不正なスーパーブロックの復元

ファイルシステムのスーパーブロック内のデータが破壊された場合は、復元しなければなりません。スーパーブロックが不正なときには、`fsck` コマンドからメッセージが表示されます。さいわい、スーパーブロックのコピーがファイルシステム内に格納されています。

`fsck -o b` コマンドを使用すると、スーパーブロックをいずれかのコピーと置き換えることができます。または `fsck` のバックアップスーパーブロックの自動検索機能を使用します。この機能は Solaris 10 6/06 リリースの新機能です。この機能の詳細は、322 ページの「バックアップスーパーブロックの自動検索」を参照してください。

スーパーブロックの詳細は、454 ページの「スーパーブロック」を参照してください。

ルート (/) ファイルシステム内のスーパーブロックが損傷し、修復できない場合は、次のどちらかの操作を実行します。

- システムをインストールし直します。
- ネットワークまたはローカルの Solaris DVD からブートし、次の手順を実行します。それらの手順で失敗する場合は、`newfs` コマンドを使ってルート (/) ファイルシステムを作成し直し、バックアップコピーから復元します。

▼ 不正なスーパーブロックを復元する方法 (Solaris 10 6/06 リリース)

これは、Solaris 10 6/06 リリースの新しい手順です。ファイルシステムに不正なスーパーブロックがある場合は、次のメッセージに示すように fsck によって自動的に代替のスーパーブロックが計算されます。

```
BAD SUPERBLOCK AT ...

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS?
LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS?
```

注意 – 損傷したスーパーブロックが含まれるファイルシステムが、ntrack や nsect などの newfs または mkfs のカスタマイズされたパラメータを使用して作成されている場合、修復処理のために fsck で自動的に計算されたスーパーブロックがファイルシステムを損傷させる可能性があります。

カスタマイズされたパラメータを使用して作成されたファイルシステムで、ファイルシステムに不正なスーパーブロックが含まれている場合、fsck は、fsck セッションを取り消すための次のようなプロンプトを表示します。

```
CANCEL FILESYSTEM CHECK?
```

ファイルシステムがカスタマイズされたパラメータを使用して作成されている場合またはこのシステム上での fsck の実行に関する他の心配がある場合は、fsck セッションの取り消しが適切な応答です。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 不正なスーパーブロックがあると思われるファイルシステムを検査します。

   ```
   # fsck /dev/rdsk/c0t1d0s0

   ** /dev/rdsk/c0t1d0s0

   BAD SUPERBLOCK at ...
   ```

3. ファイルシステムの作成方法を特定し、次のいずれかを選択します。
   - ファイルシステムが **newfs** コマンドを使用して作成された。
     - fsck がすべてのスーパーブロックが破損していることを報告し、汎用のスーパーブロックを使用する必要があります。下の例で説明するように、fsck プロンプトに応答します。

注意 – カスタマイズされたパラメータを使用してファイルシステムが作成されている場合はこのオプションを使用しないでください。このオプションは最後の手段として使用してください。バックアップコピーからファイルシステムを復元する準備をします。

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? no


USE GENERIC SUPERBLOCK FROM NEWFS? yes

CALCULATED GENERIC SUPERBLOCK WITH NEWFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y


UPDATE STANDARD SUPERBLOCK? y

81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0% fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
```

- `fsck` が次のようなメッセージを表示し、代替のスーパーブロックが見つかったことを報告します。

  ```
  FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS
  ```

  この `fsck` のシナリオでは、322 ページの「バックアップスーパーブロックの自動検索」 に示すようにプロンプトに従います。

- ファイルシステムが **mkfs** コマンドを使用して作成された。

- fsck がすべてのスーパーブロックが破損していることを報告し、汎用のスーパーブロックを使用する必要があります。下の例で説明するように、fsck プロンプトに応答します。

注意 – カスタマイズされたパラメータを使用してファイルシステムが作成されている場合はこのオプションを使用しないでください。このオプションは最後の手段として使用してください。バックアップコピーからファイルシステムを復元する準備をします。

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? no

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? yes

CALCULATED GENERIC SUPERBLOCK WITH MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y


UPDATE STANDARD SUPERBLOCK? y

81 files, 3609 used, 243605 free (117 frags, 30436 blocks, 0.0% fragmentation)
```

- fsck が次のようなメッセージを表示し、代替のスーパーブロックが見つかったことを報告します。

```
FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS
```

この fsck のシナリオでは、322 ページの「バックアップスーパーブロックの自動検索」 に示すようにプロンプトに従います。

4 プロンプトに応答し、スーパーブロックを復元します。

次のメッセージが表示される場合は、fsck を再実行する必要はありません。

```
***** FILE SYSTEM WAS MODIFIED *****
```

ただし、このメッセージのあとに fsck を再実行してもファイルシステムは損傷しません。このメッセージは、fsck の修正処置に関する情報を示すだけのものです。

▼ 不正なスーパーブロックを復元する方法 (Solaris 8、9、および 10 リリース)

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 不正なスーパーブロックがルート (/)、/usr、または /var ファイルシステム内にあるかどうかを調べ、次のどちらかの操作を実行します。

- ルート (/) ファイルシステム、/usr ファイルシステム、または /var ファイルシステムに不正なスーパーブロックが存在する場合、ネットワークまたはローカルに接続された Solaris DVD からブートします。

  ローカル接続された DVD からブートする場合は、次のコマンドを使用します。

  ```
  ok boot cdrom -s
  ```

  ブートサーバーまたはインストールサーバーがすでに設定済みのネットワークからブートする場合は、次のコマンドを使用します。

  ```
  ok boot net -s
  ```

  システムを停止する必要がある場合は、『Solaris のシステム管理 (基本編)』の第 12 章「Oracle Solaris システムのブート (手順)」または『Solaris のシステム管理 (基本編)』の「GRUB を使用して x86 システムをブートする (作業マップ)」を参照してください。

- ルート (/)、/usr、または /var のどのファイルシステムにも不正なスーパーブロックが存在しない場合は、損傷を受けたファイルシステム以外のディレクトリに移動し、マウントを解除します。

  ```
  # umount /mount-point
  ```

注意 – 次の手順では、必ず newfs -N オプションを使用してください。-N オプションを指定しない場合は、そのファイルシステムのデータがすべて破壊され、空のファイルシステムに置き換わります。

3 newfs -N コマンドを使用して、スーパーブロックの値を表示します。

```
# newfs -N /dev/rdsk/device-name
```

このコマンドの出力には、newfs コマンドによってファイルシステムが作成されたときに、スーパーブロックのコピーとして使用されることになったブロック番号が表示されます。カスタマイズされたファイルシステムを作成する方法については、457 ページの「UFS ファイルシステムパラメータのカスタマイズ」を参照してください。

4 fsck コマンドを使用して、代替スーパーブロックを指定します。

```
# fsck -F ufs -o b=block-number /dev/rdsk/device-name
```

fsck コマンドは、指定された代替スーパーブロックを使用して、一次スーパーブロックを復元します。代替ブロックとしては、常に 32 を使用できます。または、newfs -N コマンドの実行結果として出力される任意の代替ブロックを使用します。

例 21–3 不正なスーパーブロックの復元 (Solaris 8、9、および 10 リリース)

次の例は、スーパーブロックのコピー 5264 を復元する方法を示しています。

```
# newfs -N /dev/rdsk/c0t3d0s7
/dev/rdsk/c0t3d0s7: 163944 sectors in 506 cylinders of 9 tracks, 36 sectors
 83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
super-block backups (for fsck -b #) at:
 32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,
 47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,
 93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
 140528, 145760, 150992, 156224, 161456,
# fsck -F ufs -o b=5264 /dev/rdsk/c0t3d0s7
Alternate superblock location: 5264.
** /dev/rdsk/c0t3d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
#
```

fsck コマンドの構文とオプション

fsck コマンドは、ファイルシステム内の不整合を検査して修復します。オプションを指定しないで fsck コマンドを実行した場合は、修復が行われる前に確認を求めるプロンプトが表示されます。このコマンドには、次の 4 つのオプションがあります。

コマンドとオプション	説明
`fsck -m`	ファイルシステムがマウント可能かどうかの検査を行います
`fsck -y`	すべての修復に対して yes の応答が指定されたものとして処理を行います
`fsck -n`	すべての修復に対して no の応答が指定されたものとして処理を行います
`fsck -o p`	確認を求めるプロンプトを表示することなく、ファイルシステムを修復し、想定される (軽微な) 不整合箇所をすべて修正しますが、重大な問題が見つかると終了します

◆ ◆ ◆ 第 22 章

22 UFS ファイルシステム (参照情報)

この章の内容は次のとおりです。

- 453 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」
- 457 ページの「UFS ファイルシステムパラメータのカスタマイズ」

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、ディスクスライスがシリンダグループに分割されます。「シリンダグループ」は、1 つ以上の連続したディスクシリンダで構成されます。シリンダグループはさらにアドレス指定可能なブロックに分割され、このブロックによって、シリンダグループ内のファイルの構造が編成され、制御されます。各種のブロックは、ファイルシステム内で特定の機能を持っています。UFS ファイルシステムには、次の 4 種類のブロックがあります。

ブロックの種類	格納されている情報の種類
ブートブロック	システムブート時に使用される情報
スーパーブロック	ファイルシステムに関する大部分の情報
i ノード	ファイルに関する名前以外のすべての情報
記憶域またはデータブロック	各ファイルのデータ

次の節では、これらのブロックの編成と機能について説明します。

ブートブロック

ブートブロックには、システムのブート時に使用されるオブジェクトが格納されます。ファイルシステムがブートに使用されなければ、ブートブロックは空白のままです。ブートブロックは最初のシリンダグループ (シリンダグループ 0) にのみ置かれ、スライス内の最初の 8K バイトです。

スーパーブロック

スーパーブロックには、次のようなファイルシステムに関する大部分の情報が格納されます。

- ファイルシステムのサイズと状態
- ラベル (ファイルシステム名とボリューム名を含む)
- ファイルシステムの論理ブロックのサイズ
- 最終更新日時
- シリンダグループのサイズ
- シリンダグループ内のデータブロック数
- 集計データブロック
- ファイルシステムの状態
- 最後のマウントポイントのパス名

スーパーブロックには重要なデータが入っているので、ファイルシステムの作成時には複数のスーパーブロックが作成されます。

集計情報ブロックは、スーパーブロック内に保管されます。集計情報ブロックは複製されませんが、通常はシリンダグループ 0 内の主スーパーブロックと一緒のグループに分類されます。集計ブロックには、ファイルシステムが使用されるときに発生する変更が記録されます。さらに、ファイルシステム内の i ノード数、ディレクトリ数、フラグメント数、および記憶ブロック数が表示されます。

i ノード

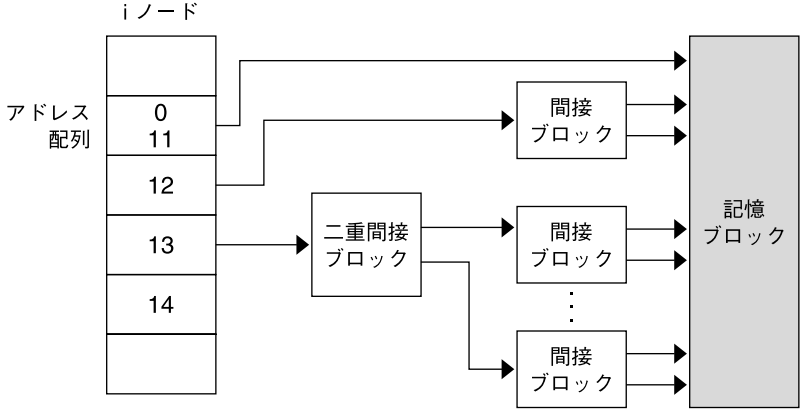
i ノードには、ファイルに関して名前以外のすべての情報が入っており、ディレクトリ内に保管されます。i ノードは 128 バイトです。i ノード情報はシリンダ情報ブロック内に保管され、次の情報が入っています。

- ファイルのタイプ
  - 通常ファイル
  - ディレクトリ
  - ブロック型特殊ファイル
  - キャラクタ型特殊ファイル
  - FIFO (名前付きパイプとも呼ぶ)

  - シンボリックリンク
  - ソケット
  - その他の i ノード – 属性ディレクトリとシャドウファイル (ACL 用)
- ファイルのモード (読み取り権 - 書き込み権 - 実行権のセット)
- ファイルへのハードリンク数
- ファイルの所有者のユーザー ID
- ファイルが属するグループ ID
- ファイル内のバイト数
- 15 個のディスクブロックアドレスの配列
- ファイルの最終アクセス日時
- ファイルの最終変更日時
- i ノードの変更日時

15 個のディスクブロックアドレスの配列 (0 - 14) は、ファイルの内容が格納されているデータブロックを指します。最初の 12 個は直接アドレスです。つまり、ファイルの内容のうち最初の 12 個の論理記憶ブロックを直接指します。ファイルが論理ブロック 12 個分より大きい場合は、13 番目のアドレスは間接ブロックを指します。間接ブロックには、ファイルの内容ではなく直接ブロックのアドレスが入っています。14 番目のアドレスは、二重間接ブロックを指します。二重間接ブロックには、間接ブロックのアドレスが入っています。15 番目のアドレスは三重間接アドレス用です。次の図に、i ノードから始まるこのアドレスブロックチェーンを示します。

図 22–1 UFS ファイルシステムのアドレスチェーン



データブロック

ファイルシステムに割り当てられた残りの領域には、データブロック (記憶ブロックともいう) が入っています。これらのデータブロックのサイズは、ファイルシステムの作成時に決定されます。デフォルトでは、データブロックは、2 つのサイズに割り当てられます。 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す。

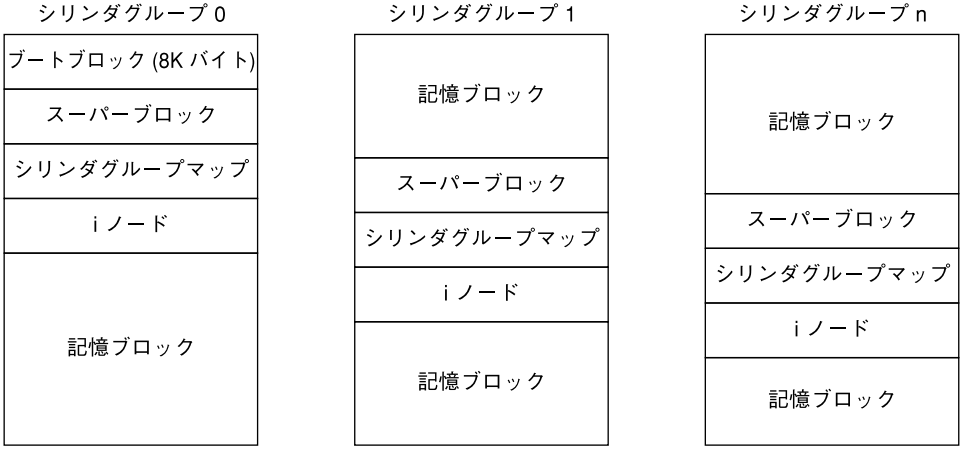
通常ファイルの場合、データブロックにはファイルの内容が入っています。ディレクトリの場合、データブロックにはディレクトリ内のファイルの i ノード番号とファイル名を示すエントリが入っています。

空きブロック

現在、i ノード、間接アドレスブロック、または記憶ブロックとして使用されていないブロックには、シリンダグループマップ内で空きを示すマークが付けられます。また、このマップはフラグメントを追跡し、断片化によるディスクパフォーマンスの低下を防止します。

UFS ファイルシステムの内容の構造を理解しやすいように、次の図に、一般的な UFS システム内の一連のシリンダグループを示します。

図 22–2 一般的な UFS ファイルシステム



UFS ファイルシステムパラメータのカスタマイズ

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。

- 457 ページの「論理ブロックサイズ」
- 458 ページの「フラグメントサイズ」
- 458 ページの「最小空き容量」
- 459 ページの「回転待ち」 (廃止)
- 459 ページの「最適化のタイプ」
- 459 ページの「i ノード数 (ファイルの数)」

これらのパラメータをカスタマイズするコマンドオプションについては、newfs(1M) と mkfs_ufs(1M) のマニュアルページを参照してください。

論理ブロックサイズ

論理ブロックサイズは、UNIX カーネルがファイルの読み書きに使用するブロックのサイズです。一般に、論理ブロックサイズは物理ブロックサイズとは異なります。物理ブロックサイズは、通常は 512K バイトで、ディスクコントローラが読み書きできる最小ブロックのサイズです。

論理ブロックサイズは、デフォルトでシステムのページサイズに設定されます。UFS ファイルシステムの場合、このデフォルト論理ブロックサイズは 8192 バイト (8K バイト) です。UFS ファイルシステムでは、ブロックサイズとして 4096 バイトまたは 8192 バイト (4K または 8K バイト) がサポートされます。論理ブロックの推奨サイズは 8K バイトです。

SPARC のみ – sun-4u プラットフォームでは、8192 バイトのブロックサイズしか指定できません。

システムに最善の論理ブロックサイズを選択するには、必要なパフォーマンスと使用可能容量を検討してください。ほとんどの UFS システムでは、8K バイトのファイルシステムが最高のパフォーマンスを発揮し、ディスクパフォーマンスと一次メモリーやディスク上の領域の使用量が適切なバランスに保たれます。

原則として、効率を高めるには、ほとんどのファイルがきわめて大きいファイルシステムには大きめの論理ブロックサイズを使用します。ほとんどのファイルがきわめて小さいファイルシステムには、小さめの論理ブロックサイズを使用します。ファイルシステム上で quot -c *filesystem* コマンドを使用すると、ファイルの分散に関する詳細なレポートをブロックサイズ別に表示できます。

ただし、通常は、ファイルシステムの作成時に設定されたページサイズが最適です。

フラグメントサイズ

ファイルが作成または拡張されると、論理ブロック全体または「フラグメント」と呼ばれる部分のディスク容量が割り当てられます。ファイルのためにディスク容量が必要になると、まずブロック全体が割り当てられ、次に残りの部分にブロックのうち 1 つまたは複数のフラグメントが割り当てられます。小型ファイルの場合、割り当てはフラグメントから始まります。

ブロック全体ではなく、そのフラグメントを割り当てることができるので、ブロック内の未使用のホールによって生じるディスク容量の「断片化」が減少し、容量の節約になります。

UFS ファイルシステムを作成するときに、「フラグメントサイズ」を定義します。デフォルトのフラグメントサイズは 1K バイトです。各ブロックは、1 個、2 個、4 個、または 8 個のフラグメントに分割できます。この場合、フラグメントサイズは 8192 バイトから 512 バイト (4K バイトのファイルシステムのみ) までです。実際には、下限はディスクのセクターサイズ、通常は 512 バイトに連動します。

マルチテラバイトのファイルシステムの場合、フラグメントサイズはファイルシステムのブロックサイズに等しくなります。

注 – フラグメントサイズの上限は論理ブロックサイズに等しくなります。この場合、フラグメントは存在しないことになります。容量よりも速度を重視する場合、きわめて大型のファイルがあるファイルシステムには、この構成が最適なことがあります。

フラグメントサイズを選択するときには、処理時間と容量の兼ね合いを考慮してください。 フラグメントサイズが小さければ容量の節約になりますが、割り当てには時間がかかります。原則として、格納効率を高めるには、ほとんどのファイルが大型のファイルシステムには、大きめのフラグメントサイズを使用します。ほとんどのファイルが小型のファイルシステムには、小さめのフラグメントサイズを使用します。

最小空き容量

「最小空き容量」とは、ファイルシステムの作成時に予約分として保持されるディスク容量の割合です。デフォルトの予約分は、((64M バイト/パーティションサイズ) * 100) で算出し、その値はもっとも近い整数に切り捨てられ、ディスク容量の 1% から 10% の範囲に制限されます。

ファイルシステム内の空き容量が少なくなるほど、アクセス速度が低下するので、空き容量は重要です。十分な空き容量があれば、UFS ファイルシステムは効率

よく動作します。ファイルシステムがいっぱいになって、使用可能なユーザー領域を使い果たすと、スーパーユーザー以外は予約済みの空き容量にアクセスできなくなります。

df などのコマンドは、最小空き容量として割り当て済みの分を差し引いて、ユーザーに使用可能な容量をパーセントで表示します。コマンドでファイルシステム内のディスク容量の 100 パーセント以上が使用中であると表示される場合は、予約分の一部がスーパーユーザーによって使用されています。

ユーザーに割り当てを適用する場合に、各ユーザーが使用可能な容量には予約分の空き容量は含まれません。tunefs コマンドを使用すると、既存のファイルシステムの最小空き容量の値を変更できます。

回転待ち

このパラメータはもう使用しません。指定した値に関わらず、値は常に 0 に設定されます。

最適化のタイプ

「最適化のタイプ」パラメータの設定には、「*space*」と「*time*」があります。

- **space** – 最適化のタイプに space を選択すると、断片化を最小限度に抑え、ディスクの使用状況が最適化されるようにディスクブロックが割り当てられます。
- **time** – 最適化のタイプに time を選択すると、配置はあまり重視されず、できるだけ高速になるようにディスクブロックが割り当てられます。空き容量が十分ある場合、それほど細かく断片化しなくても、比較的簡単にディスクブロックを割り当てることができます。デフォルトは *time* 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。

詳細は、tunefs(1M) のマニュアルページを参照してください。

i ノード数 (ファイルの数)

i ノード 1 個あたりのバイト数によって、ファイルシステム内における i ノードの密度が決まります。ファイルシステムの合計サイズをこの値で割ると、作成すべき i ノードの個数が得られます。i ノードが割り当てられたら、ファイルシステムを作成し直さないかぎり、その数は変更できません。

1G バイト未満ファイルシステムが 1G バイト以下の場合、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) です。ファイルシステムが 1G バイトを超える場合、次の公式が使用されます。

ファイルシステムのサイズ	i ノードごとのバイト数
1G バイト以下	2048
2G バイト未満	4096
3G バイト未満	6144
3G バイト以上、1T バイト以下	8192
1T バイト超または -T オプションを指定して作成した場合	1048576

多数のシンボリックリンクを持つファイルシステムでは、平均ファイルサイズが小さくなることがあります。ファイルシステムに多数の小型ファイルが格納される場合は、このパラメータに小さい値を与えてもかまいません。i ノード数が少ないために i ノードが不足するよりも、多すぎる方が好ましいことを留意してください。i ノード数が少なすぎると、実際には空のディスクスライス上でも最大ファイル数に達してしまうことがあります。

UFS ファイルとファイルシステムの最大サイズ

UFS ファイルシステムの最大サイズは、およそ 16T バイトで、使用できる領域は、そこから約 1 パーセントのオーバーヘッドを引いたサイズです。「空白」ファイルの論理サイズは 1T バイトです。ただし、ファイルに格納できる実際のデータ容量は、1T バイトから約 1 パーセントのファイルシステムオーバーヘッドを引いたサイズです。

UFS サブディレクトリの最大数

UFS ファイルシステム内のディレクトリあたりのサブディレクトリの最大数は、32,767 です。この制限はあらかじめ定義されたものなので、変更できません。

◆ ◆ ◆ 第 23 章

UFS ファイルシステムのバックアップと復元 (概要)

この章では、ufsdump コマンドと ufsrestore コマンドを使用して、UFS ファイルシステムのバックアップおよび復元を実行する際のガイドラインと計画の作成について説明します。

この章の内容は以下のとおりです。

- 461 ページの「UFS バックアップと復元についての参照先」
- 462 ページの「UFS ファイルシステムのバックアップと復元の概要」
- 463 ページの「ファイルシステムをバックアップする理由」
- 463 ページの「どの UFS ファイルシステムをバックアップするかの計画」
- 466 ページの「バックアップタイプの選択」
- 467 ページの「テープデバイスの選択」
- 467 ページの「UFS ファイルシステムのバックアップおよび復元の概要 (作業マップ)」
- 468 ページの「バックアップスケジュールに関するその他の注意事項」
- 472 ページの「バックアップスケジュールの例」

UFS バックアップと復元についての参照先

バックアップまたは復元処理	参照先
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います。	第 24 章「UFS ファイルとファイルシステムのバックアップ (手順)」
fssnap コマンドを使用して UFS スナップショットを作成します。	第 25 章「UFS スナップショットの使用 (手順)」
ufsrestore コマンドを使用してファイルシステムを復元します。	第 26 章「UFS ファイルとファイルシステムの復元 (手順)」

バックアップまたは復元処理	参照先
cpio、dd、pax、および cpio コマンドを使用してファイルおよびディレクトリをコピーします。	第 28 章「UFS ファイルとファイルシステムのコピー (手順)」

UFS ファイルシステムのバックアップと復元の概要

ファイルシステムの「バックアップ」とは、消失、損傷、または破損に備えて、ファイルシステムをリムーバブルメディア (テープなど) にコピーすることを意味します。ファイルシステムの「復元」とは、現在のバックアップファイルをリムーバブルメディアから作業ディレクトリに適宜コピーすることを意味します。

この章では、UFS ファイルシステムのバックアップおよび復元に使用する ufsdump および ufsrestore コマンドについて説明します。ほかのコマンドを使用してファイルやファイルシステムをコピーし、ファイルの共有や移動を行うこともできます。次の表に、個々のファイルやファイルシステムをほかのメディアにコピーするためのすべてのコマンドの参照先を示します。

表 23–1 ファイルとファイルシステムのバックアップおよび復元用コマンド

作業	コマンド	参照先
1 つまたは複数のファイルシステムをローカルまたはリモートのテープデバイスにバックアップします。	ufsdump	第 24 章「UFS ファイルとファイルシステムのバックアップ (手順)」または 第 27 章「UFS バックアップおよび復元コマンド (参照情報)」
ファイルシステムの読み取り専用コピーを作成します。	fssnap	第 25 章「UFS スナップショットの使用 (手順)」
ネットワーク上のシステムのファイルシステム全体をバックアップサーバーからバックアップします。	Solstice Backup ソフトウェア	『Solstice Backup 6.1 Administration Guide』
NIS+ マスターサーバーをバックアップ、復元します。	nisbackup および nisrestore コマンド	『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』
テープまたはフロッピーディスク上でファイルをコピー、表示、検索します。	tar、cpio、または pax コマンド	第 28 章「UFS ファイルとファイルシステムのコピー (手順)」
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	第 28 章「UFS ファイルとファイルシステムのコピー (手順)」

表 23–1 ファイルとファイルシステムのバックアップおよび復元用コマンド (続き)

作業	コマンド	参照先
ファイルシステム全体または個々のファイルを、リムーバブルメディアから作業ディレクトリに復元します。	ufsrestore コマンド	第 26 章「UFS ファイルとファイルシステムの復元 (手順)」

ファイルシステムをバックアップする理由

ファイルのバックアップは、もっとも重要なシステム管理作業の 1 つです。次のような問題によるデータの消失に備えて、定期的にバックアップを実行することをお勧めします。

- システムのクラッシュ
- 不注意によるファイルの削除
- ハードウェア障害
- 天災 (火災、台風、地震など)
- システムの再インストールやアップグレード時に発生する問題

どの **UFS** ファイルシステムをバックアップするかの計画

頻繁に更新されるファイルシステムなど、ユーザーにとって重要なファイルシステムはバックアップしておく必要があります。次の表に、スタンドアロンシステムおよびサーバーについてバックアップすべきファイルシステムに関する一般的なガイドラインを示します。

表 23–2 スタンドアロンシステムにおけるバックアップ対象のファイルシステム

バックアップするファイルシステム	説明	バックアップ間隔
ルート (/) – スライス 0	このファイルシステムにはカーネルが格納されていますが、/var ディレクトリが格納されていることもあります。/var ディレクトリには、一時ファイル、ロギングファイル、または状態ファイルのほか、頻繁に更新されるシステムアカウンティングファイルやメールファイルが含まれる場合もあります。	定期的 (毎週、毎日など)

表 23–2 スタンドアロンシステムにおけるバックアップ対象のファイルシステム (続き)

バックアップするファイルシステム	説明	バックアップ間隔
/usr – スライス 6、/opt	/usr および /opt ファイルシステムには、ソフトウェアと実行可能ファイルが含まれます。/opt ディレクトリは、ルート (/) の一部であるか、独自のファイルシステムです。	随時
/export/home – スライス 7	このファイルシステムには、スタンドアロンシステム上のすべてのユーザーのディレクトリおよびサブディレクトリを格納できます。	ルート (/) や /usr よりも頻繁に、サイトのニーズによっては毎日
/export、/var、またはその他のファイルシステム	/export ファイルシステムには、ディスクレスクライアント用のカーネルおよび実行可能プログラムが含まれます。/var ディレクトリには、一時ファイル、ロギングファイル、または状態ファイルが含まれます。	サイトの必要に応じて

表 23–3 UFS ファイルシステムをサーバー用にバックアップする

バックアップするファイルシステム	説明	バックアップ間隔
ルート (/) – スライス 0	このファイルシステムには、カーネルおよび実行可能プログラムが含まれます。	サイトの必要に応じて毎日 1 度ないし月に 1 度。 ネットワーク上でユーザーとシステムの追加および削除を頻繁に実行する場合、このファイルシステム内の構成ファイルを変更する必要があります。この場合、週に 1 度から月に 1 度の間隔で、ルート (/) ファイルシステムの完全バックアップを実行する必要があります。 サイトでユーザーのメールをメールサーバー上の /var/mail ディレクトリに保管しておき、後でクライアントシステムがマウントする場合は、ルート (/) ディレクトリを毎日バックアップすることをお勧めします。別のファイルシステムの場合は、/var を毎日バックアップすることをお勧めします。
/export – スライス 3	このファイルシステムには、ディスクレスクライアント用のカーネルおよび実行可能プログラムが含まれます。	サイトの必要に応じて毎日 1 度ないし月に 1 度。 このファイルシステム内の情報はスライス 0 にあるサーバーのルートディレクトリと同様なので、ファイルシステムは頻繁に変化することはありません。サイトからメールをクライアントシステムに送信していない場合、このファイルシステムのバックアップは随時に実行するだけでかまいません。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ります。
/usr – スライス 6、/opt	/usr および /opt ファイルシステムには、ソフトウェアと実行可能ファイルが含まれます。/opt ディレクトリは、ルート (/) の一部であるか、独自のファイルシステムです。	サイトの必要に応じて毎日 1 度ないし月に 1 度。 ソフトウェアの追加や削除が頻繁に行われなければ、これらのファイルシステムは比較的静的です。

表 23–3 UFS ファイルシステムをサーバー用にバックアップする (続き)

バックアップするファイルシステム	説明	バックアップ間隔
/export/home – スライス 7	このファイルシステムには、システムの全ユーザーのホームディレクトリを格納できます。このファイルシステム内のファイルは、頻繁に変更されます。	毎日または毎週。

バックアップタイプの選択

ufsdump コマンドを使用すると、完全バックアップまたは増分バックアップを実行できます。fssnap コマンドを使用すると、ファイルシステムの一時イメージを作成できます。次の表に、各バックアップタイプの手順の違いを示します。

表 23–4 バックアップタイプの相違点

バックアップのタイプ	結果	長所	短所
完全 (フル)	ファイルシステムやディレクトリ全体をコピーします	すべてのデータが 1 カ所に格納されます	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ります。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ります。複数のテープを検索しなければならない場合もあります。
スナップショット	ファイルシステムの一時イメージを作成します	システムはマルチユーザーモードで動作可能です	スナップショットの作成中は、システムのパフォーマンスが低下する可能性があります。
増分	指定されたファイルシステム内で、前回のバックアップ以降に変更されたファイルのみをコピーします	ファイルシステム内の小さな変化を簡単に検索できます	どの増分テープにファイルが入っているかを探すのに時間がかかることがあります。最後の完全バックアップに戻らなければならない場合もあります。

テープデバイスの選択

次の表に、バックアップ処理中にファイルシステムを格納するための典型的なテープデバイスを示します。記憶容量は、ドライブのタイプとテープに書き込むデータのタイプによって異なります。テープデバイスの詳細は、第 29 章「テープドライブの管理 (手順)」を参照してください。

表 23–5　ファイルシステムのバックアップに使用する一般的なメディア

バックアップメディア	記憶容量
1/2 インチのリールテープ	140M バイト (6250 bpi)
2.5G バイト、1/4 インチのカートリッジ (QIC) テープ	2.5G バイト
DDS3 4mm カートリッジテープ (DAT)	12 - 24G バイト
14G バイト、8mm カートリッジテープ	14G バイト
DLT 7000 1/2 インチ カートリッジテープ	35 - 70G バイト

UFS ファイルシステムのバックアップおよび復元の概要 (作業マップ)

この作業マップを使用して、ファイルシステムのバックアップおよび復元作業をすべて確認できます。各作業は、実行するバックアップタイプの判別などの一連の追加作業を指し示します。

作業	説明	参照先
1. バックアップするファイルシステムを識別します。	毎日、毎週、または毎月、バックアップの必要なファイルシステムを識別します。	463 ページの「どの UFS ファイルシステムをバックアップするかの計画」
2. バックアップタイプを決定します。	サイトに適したファイルシステムのバックアップタイプを決定します。	466 ページの「バックアップタイプの選択」
3. バックアップを作成します。	次のいずれかの方法を使用します。	
	ファイルシステムの完全バックアップおよび増分バックアップを実行する場合は、`ufsdump` コマンドを使用します。 アクティブかつマウント中のファイルシステムのスナップショットを作成する場合は、`fssnap` コマンドの使用を考慮します。	第 24 章「UFS ファイルとファイルシステムのバックアップ (手順)」 第 25 章「UFS スナップショットの使用 (手順)」

作業	説明	参照先
	個人のホームディレクトリや、小規模で重要度の低いファイルシステムの完全バックアップを作成する場合は、tar、cpio、または pax コマンドを使用します。	第 28 章「UFS ファイルとファイルシステムのコピー (手順)」
4. (省略可能) ファイルシステムを復元します。	ファイルまたはファイルシステムのバックアップに使用するコマンドに基づいて、復元方法を選択します。 ufsdump コマンドを使用して作成したファイルシステムのバックアップを復元します。 tar、cpio、または pax コマンドを使用して作成したファイルシステムを復元します。	第 26 章「UFS ファイルとファイルシステムの復元 (手順)」 第 28 章「UFS ファイルとファイルシステムのコピー (手順)」
5. (省略可能) ルート (/) または /usr ファイルシステムを復元します。	ルート (/) または /usr ファイルシステムの復元は、それほど重要でないファイルシステムの復元よりも複雑です。これらのファイルシステムの復元中に、ローカルの Solaris DVD またはネットワークから起動する必要があります。	513 ページの「UFS ルート (/) および /usr ファイルシステムを復元する方法」

バックアップスケジュールに関するその他の注意事項

「バックアップスケジュール」とは、ufsdump コマンドを実行するように設定するスケジュールです。この節では、バックアップスケジュールの作成時に考慮するべき点について説明します。また、バックアップスケジュールの例もいくつか紹介します。

バックアップスケジュールは、次の点を考慮に入れて作成します。

- バックアップに使用するテープの数を最小限に抑える
- バックアップの実行に使用できる時間
- 損傷したファイルシステムの完全復元に使用できる時間
- 不注意に削除した個々のファイルの検索に使用できる時間

バックアップ頻度

バックアップに費やす時間とメディアの数を最小限度に抑える必要がない場合は、完全バックアップを毎日実行してもかまいません。しかし、多くのサイトの場合、このようなバックアップ方法は現実的ではないので、ほとんどの場合は増分

バックアップが使用されます。その場合は、サイトが過去 4 週間分のバックアップからファイルを十分復元できるようにしてください。そのためには、少なくとも 1 週分ごとに 1 組ずつ、合計 4 組のテープが必要です。各組のテープを毎月使い回すことになります。また、少なくとも 1 年分の月別のバックアップを保存する必要があります。そして、数年分の年度別バックアップを保管しておく必要があります。

バックアップ間隔の用語と定義

次の表に、バックアップ間隔の用語および定義を示します。

用語	定義
スナップショット	ファイルシステムの一時イメージを作成します。
完全バックアップ	ファイルシステムまたはディレクトリの完全なコピーを作成します。
増分バックアップ	指定したファイルシステムのファイルのうち、前回のバックアップ以降に変更されたファイルだけをコピーします。増分バックアップには、次の種類があります。 ■ 日単位累積 – 月曜日に 1 日分のファイル変更をコピーします。次に、月曜日のバックアップを火曜日、水曜日のように、各曜日のファイル変更で上書きしていきます。 ■ 日単位増分 – 1 日分のファイル変更をコピーします。つまり、月曜日、火曜日のように、各曜日の変更の差分テープを記録していきます。 ■ 週単位累積 – その週に変更されたファイルをコピーします。前の週のファイル変更も残しておきます。 ■ 週単位増分 – 前の週のバックアップ以降に変更されたファイルをコピーします。

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

次の表に、バックアップスケジュールのガイドラインを示します。バックアップスケジュールに関するその他の注意事項については、468 ページの「バックアップスケジュールに関するその他の注意事項」を参照してください。

表 23–6 バックアップスケジュールのガイドライン

必要なファイル復元操作	バックアップ間隔	Comments
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ります	作業日ごとに日単位増分バックアップを実行します。 日単位増分バックアップには同じテープを再利用しません。	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ります。ただし、このスケジュールの場合は、毎日異なるテープを使用する必要があります。このように運用しないと、必要なバージョンのファイルを復元できなくなることがあるためです。 たとえば、火曜日に変更があったファイルが木曜日にも変更されると、金曜 日の下のレベルのバックアップでは、火曜日の夜ではなく木曜日の夜に変更されたように見えます。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、それを復元できません。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ません。
ファイルシステム全体を短時間で復元する必要があります	下位のレベルのバックアップを頻繁に実行します。	—
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ます	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します。	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ります。
使用するテープ数を最小限に抑えます	1 週間に実行する増分バックアップのレベルを上げます。	毎日の変更のみが各日単位テープに保存されます。
	週末に実行するバックアップのレベルを上げます。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ます。	変更は (月単位ではなく) 週単位でしか週単位テープに保存されません。

表 23–6 バックアップスケジュールのガイドライン (続き)

必要なファイル復元操作	バックアップ間隔	Comments
	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ます。	この操作には、ufsdump の 'no rewind' オプションを使用します。たとえば、/dev/rmt/0n のように指定します。

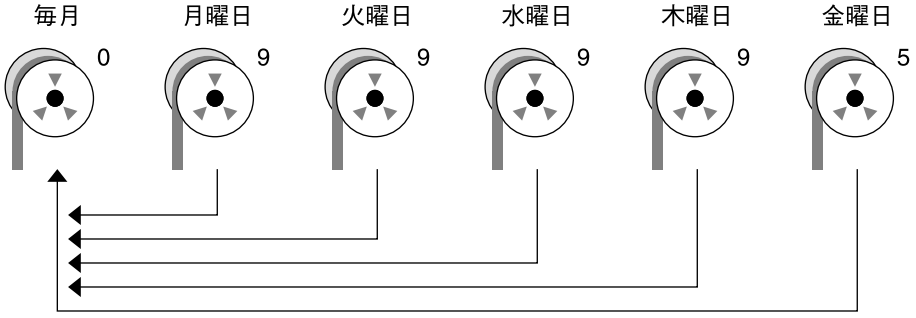
ダンプレベルを使用して増分バックアップを作成する

ufsdump コマンドで指定するダンプレベル (0 - 9) により、バックアップするファイルを決定できます。ダンプレベル 0 を指定すると、完全バックアップが作成されます。増分バックアップのスケジュール設定にレベル 1 から 9 までの番号が使用されますが、特に意味が定義されているわけではありません。レベル 1 から 9 は、累積バックアップまたは個別バックアップのスケジュール設定に使用する番号の範囲にすぎません。レベル 1 から 9 までが意味するのは、大小による相互関係だけです。下位ダンプレベル番号を使用した場合は、常に完全バックアップまたは累積バックアップが再開されます。次の例で、1 から 9 までのレベルを使用する増分ダンプの手順を示します。

例—日単位累積バックアップのダンプレベル

日単位の累積増分バックアップを実行するのが、もっとも一般に使用されるバックアップスケジュールで、ほとんどの場合に推奨できます。次の例で、月曜日から木曜日までレベル 9 のダンプを使用し、金曜日にはレベル 5 のダンプを使用してプロセスを再開するスケジュールを示します。

図 23–1 増分バックアップ: 日単位累積



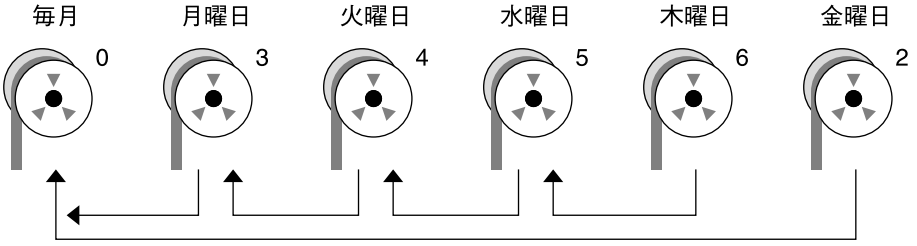
前の例では、1 から 9 までの範囲内で他の番号を使用しても同じ結果が得られます。ポイントは、月曜日から木曜日まで同じ番号を使用し、金曜日にはそれよ

り「小さい」番号を使用することです。たとえば、レベル 4、4、4、4、2 や 7、7、7、7、5 を指定してもかまいません。

例—日単位増分バックアップのダンプレベル

次の例で、1 日分の作業内容のみを別々のテープ上で保存するスケジュールを示します。このようなバックアップは、日単位増分バックアップと呼ばれます。この場合、月曜日から木曜日までは連続するダンプレベル番号(3、4、5、6) を使用し、金曜日にはそれより小さい番号(2) を使用します。金曜日はレベル番号が小さいので、プロセスが再開されます。

図 23–2 増分バックアップ: 日単位増分



前の例では、連番 6、7、8、9 の次に 2 を使用したり、5、6、7、8 の次に 3 を使用したりすることもできます。番号自体は特定の意味を持たないことに注意してください。この例が示しているように、連続した番号を並べることによって、それらの番号に意味が生まれます。

バックアップスケジュールの例

バックアップスケジュールの例をいくつか紹介します。どのスケジュールも、完全バックアップ (ダンプレベル 0) から始めることと、-u オプションを使用して /etc/dumpdates ファイルに各バックアップを記録することを前提としています。

例 — 日単位累積、週単位累積バックアップスケジュール

表 23–7 に、よく使用される増分バックアップスケジュールを示します。これは、ほとんどの場合に推奨できるスケジュールです。このスケジュールで実行される処理は次のとおりです。

- 前週の終わりの下位レベルのバックアップ以降に変更があったすべてのファイルが毎日保存されます。

- 月-金のレベル 9 のバックアップの場合は、直前のレベル 0 またはレベル 5 のバックアップがもっとも近い下位バックアップレベルになります。したがって、各週のテープには、前週の終わり (第 1 週の場合は初期レベル 0 のバックアップ) 以降に変更があったすべてのファイルが累積されます。
- 毎週金曜日のレベル 5 のバックアップの場合、もっとも近い下位レベルのバックアップは、月初めに実行されたレベル 0 のバックアップです。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ます。

表 23–7 日単位累積/週単位累積バックアップスケジュール

	開始日	月	火	水	木	金
月の 1 日目	0					
第 1 週		9	9	9	9	5
第 2 週		9	9	9	9	5
第 3 週		9	9	9	9	5
第 4 週		9	9	9	9	5

次の表に、日単位累積、週単位累積スケジュールを使用して、テープの内容が 2 週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 23–8 日単位累積、週単位累積スケジュールのテープの内容

	月	火	水	木	金
第 1 週	ab	abc	abcd	abcde	abcdef
第 2 週	g	gh	ghi	ghij	abcdefghijk

日単位累積、週単位累積スケジュールのテープ要件

このスケジュールでは、日単位テープを再利用する場合には、6 本のテープが必要になります。ただし、曜日ごとに 4 本の日単位テープを別々に使用する場合は、次に示すように 9 本のテープが必要になります。

- レベル 0 のバックアップ用に 1 本
- 金曜日用に 4 本
- 日単位テープ用に 1 本または 4 本

ファイルシステム全体を復元する必要があれば、次のテープが必要になります。

- レベル 0 用に 1 本
- 前週の金曜日用に 1 本

- 前週の金曜日以降の最新の日単位テープ 1 本

例 — 日単位累積、週単位増分バックアップスケジュール

次の表に、各曜日のテープに月曜日 (第 1 週の場合は初期レベル 0 のバックアップ) 以降に変更があったすべてのファイルが累積されるスケジュールを示します。また、各金曜日のテープには、その週に変更されたすべてのファイルが入ります。

表 23–9 日単位累積、週単位増分バックアップスケジュール

	開始日	月	火	水	木	金
月の 1 日目	0					
第 1 週		9	9	9	9	3
第 2 週		9	9	9	9	4
第 3 週		9	9	9	9	5
第 4 週		9	9	9	9	6

次の表に、日単位累積、週単位増分バックアップスケジュールの場合、テープの内容が 2 週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 23–10 日単位累積、週単位増分バックアップスケジュールのテープの内容

	月	火	水	木	金
第 1 週	a b	a b c	a b c d	a b c d e	a b c d e f
第 2 週	g	g h	g h i	g h i j	g h i j k

日単位累積、週単位増分バックアップスケジュールのテープ要件

このスケジュールでは、日単位テープを再利用する場合には、6 本のテープが必要になります。ただし、曜日ごとに 4 本の日単位テープを別々に使用する場合は、次に示すように 9 本のテープが必要になります。

- レベル 0 のバックアップ用に 1 本
- 金曜日用に 4 本
- 日単位テープ用に 1 本または 4 本

ファイルシステム全体を復元する必要があれば、次のテープが必要になります。

- レベル 0 用に 1 本
- すべての金曜日のテープ
- 前週の金曜日以降の最新の日単位テープ 1 本

例 — 日単位増分、週単位累積バックアップスケジュール

次の表に、各曜日のテープには前日以降に変更があったファイルのみが入るスケジュールを示します。また、各金曜日のテープには、月初めの初期レベル 0 バックアップ以降に変更されたすべてのファイルが入ります。

表 23–11 日単位増分、週単位累積バックアップスケジュール

	開始日	月	火	水	木	金
月の 1 日目	0					
第 1 週		3	4	5	6	2
第 2 週		3	4	5	6	2
第 3 週		3	4	5	6	2
第 4 週		3	4	5	6	2

次の表に、日単位増分、週単位累積スケジュールの場合、テープの内容が 2 週間でどのように変化するかを示します。各文字はそれぞれ異なるファイルを示します。

表 23–12 日単位増分、週単位累積バックアップスケジュールのテープの内容

	月	火	水	木	金
第 1 週	a b	c d	e f g	h i	a b c d e f g h i
第 2 週	j k l	m	n o	p q	a b c d e f g h i j k l m n o p q r s

日単位増分、週単位累積スケジュールのテープ要件

このスケジュールでは、少なくとも 9 本のテープが必要になります。これは日単位テープを毎週再利用することが前提となっていますが、これはお勧めできません。週単位テープを 1 カ月保存する場合は、21 本のテープが必要になります。可能であれば、こちらの方法をお勧めします。 内訳は、レベル 0 に 1 本、金曜日用に 4 本、日単位テープ 4 本または 16 本です。

- レベル 0 のバックアップ用に 1 本
- 金曜日のバックアップ用に 4 本
- 日単位のテープを 4 本または 16 本

ファイルシステム全体を復元する必要があれば、次のテープが必要になります。

- レベル 0 用に 1 本
- 前週の金曜日用に 1 本
- 前週の金曜日以降のその週のすべての日単位テープ

例—サーバーの月単位バックアップスケジュール

次の表に、ユーザーがプログラム開発や文書作成のようなファイル集約型の作業を実行する小型ネットワーク上の、使用頻度の高いファイルサーバーのバックアップ方針の例を示します。この例は、バックアップ期間が日曜日に始まり、1 週 7 日間を 4 週間行うものと想定しています。

表 23–13 例—サーバーの月単位バックアップスケジュール

ディレクトリ	日付	ダンプレベル	テープ名
ルート (/)	第 1 日曜日	0	*n* 本のテープ
/usr	第 1 日曜日	0	*n* 本のテープ
/export	第 1 日曜日	0	*n* 本のテープ
/export/home	第 1 日曜日	0	*n* 本のテープ
	第 1 月曜日	9	A
	第 1 火曜日	9	B
	第 1 水曜日	5	C
	第 1 木曜日	9	D
	第 1 金曜日	9	E
	第 1 土曜日	5	F
ルート (/)	第 2 日曜日	0	*n* 本のテープ
/usr	第 2 日曜日	0	*n* 本のテープ
/export	第 2 日曜日	0	*n* 本のテープ
/export/home	第 2 日曜日	0	*n* 本のテープ
	第 2 月曜日	9	G
	第 2 火曜日	9	H
	第 2 水曜日	5	I
	第 2 木曜日	9	J
	第 2 金曜日	9	K
	第 2 土曜日	5	L
ルート (/)	第 3 日曜日	0	*n* 本のテープ
/usr	第 3 日曜日	0	*n* 本のテープ
/export	第 3 日曜日	0	*n* 本のテープ

表 23–13 例—サーバーの月単位バックアップスケジュール (続き)

ディレクトリ	日付	ダンプレベル	テープ名
/export/home	第 3 日曜日	0	*n* 本のテープ
	第 3 月曜日	9	M
	第 3 火曜日	9	N
	第 3 水曜日	5	O
	第 3 木曜日	9	P
	第 3 金曜日	9	Q
	第 3 土曜日	5	R
ルート (/)	第 4 日曜日	0	*n* 本のテープ
/usr	第 4 日曜日	0	*n* 本のテープ
/export	第 4 日曜日	0	*n* 本のテープ
/export/home	第 4 日曜日	0	*n* 本のテープ
	第 4 月曜日	9	S
	第 4 火曜日	9	T
	第 4 水曜日	5	U
	第 4 木曜日	9	V
	第 4 金曜日	9	W
	第 4 土曜日	5	X

このスケジュールでは、4*n* 本のテープ (ルート (/)、/usr、/export、および /export/home ファイルシステムの 4 回の完全バックアップに必要な本数) を使用します。また、/export/home ファイルシステムの増分バックアップ用に 24 本のテープを使用します。このスケジュールは、増分バックアップごとに 1 本ずつテープを使用し、それを 1 カ月は保存することを前提としています。

このスケジュールの機能を、次に示します。

1. 日曜日ごとに、ルート (/)、/usr、/export、および /export/home ファイルシステムの完全バックアップ (レベル 0) を実行します。レベル 0 のテープを少なくとも 3 カ月は保存します。
2. 月の第 1 月曜日に、テープ A を使用して /export/home ファイルシステムのレベル 9 のバックアップを実行します。ufsdump コマンドは、下のレベルのバックアップ以降に変更があったすべてのファイルをコピーします。この場合、直前の下位レベルのバックアップは、日曜日に実行したレベル 0 のバックアップになります。

3. 月の第 1 火曜日に、テープ B を使用して /export/home ファイルシステムのレベル 9 のバックアップを実行します。この場合も、ufsdump コマンドは、直前の下のレベルのバックアップ (日曜日のレベル 0 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
4. 月の第 1 水曜日に、テープ C を使用して /export/home ファイルシステムのレベル 5 のバックアップを実行します。ufsdump コマンドは、日曜日以降に変更があったすべてのファイルをコピーします。
5. 木曜日と金曜日には、テープ D と E を使用して、/export/home ファイルシステムのレベル 9 のバックアップを実行します。ufsdump コマンドは、最後の下位レベルのバックアップ (水曜日のレベル 5 のバックアップ) 以降に変更があったすべてのファイルをコピーします。
6. 月の第 1 土曜日に、テープ F を使用して /export/home のレベル 5 のバックアップを実行します。ufsdump コマンドは、下のレベルのバックアップ (この場合は日曜日に実行したレベル 0 のバックアップ) 以降に変更があったすべてのファイルをコピーします。テープを再利用する場合は、テープ A から F までを次の 4 週間の第 1 月曜日までは保存しておきます。
7. 次の 3 週間は、テープ G から L までと、日曜日のレベル 0 のバックアップ用に 4*n* 本のテープを使用して、手順 1 から 6 までを繰り返します。
8. 4 週ごとに、レベル 0 のバックアップ用に新しいテープ 1 組と、増分バックアップ用のテープ A から X までを再利用して、手順 1 から 7 までを繰り返します。レベル 0 のテープは、3 カ月後に再利用できるようになります。

   このスケジュールでは、各ファイルを 1 カ月間で段階別に保存できます。多数のテープが必要ですが、テープのライブラリを確実に用意できます。テープの本数を減らすには、テープ A から F までを毎週再利用します。

◆ ◆ ◆ 第 24 章

UFS ファイルとファイルシステムのバックアップ (手順)

この章では、ufsdump コマンドを使用したファイルシステムのバックアップ手順について説明します。

手順の詳細は、479 ページの「UFS ファイルとファイルシステムのバックアップ (作業マップ)」.を参照してください。

バックアップの実行方法の概要については、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。

個々のファイルをフロッピーディスクにバックアップする方法については、第 28 章「UFS ファイルとファイルシステムのコピー (手順)」を参照してください。

ufsdump コマンドの追加情報については、第 27 章「UFS バックアップおよび復元コマンド (参照情報)」を参照してください。

UFS ファイルとファイルシステムのバックアップ (作業マップ)

作業	説明	参照先
1. ファイルシステムのバックアップの準備をします。	バックアップで使用するファイルシステム、バックアップタイプ、テープデバイスを確認します。	480 ページの「ファイルシステムバックアップの実行準備」
2. ファイルシステムのバックアップに必要なテープの数を決定します。	ファイルシステムの完全バックアップに必要なテープの数を決定します。	481 ページの「完全バックアップに必要なテープ数を決定する」

作業	説明	参照先
3. ファイルシステムのバックアップを実行します。	ファイルシステムの完全バックアップを実行して、全ファイルの基準コピーを取得します。 日単位で変更されたファイルのコピーを保持することが、サイトで重要かどうかに基づいて、ファイルシステムの増分バックアップを実行します。	482 ページの「UFS ファイルシステムのバックアップをテープに作成する方法」

ファイルシステムバックアップの実行準備

ファイルシステムのバックアップの準備は、計画 (第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照) から始め、さらに、次の項目を選択します。

- バックアップを作成するファイルシステム
- 実行するバックアップのタイプ (完全または増分)
- バックアップのスケジュール
- テープドライブ

詳細は、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。

この節では、ファイルシステムのバックアップを作成する前に実行する必要のある、次の 2 つの作業について説明します。

- バックアップを作成するファイルシステムの名前を検索する
- 完全バックアップの作成に必要なテープの本数を決定する

▼ UFS ファイルシステム名を検索する方法

1. `/etc/vfstab` ファイルの内容を表示します。

   ```
   $ more /etc/vfstab
   ```

2. `mount point` 列に表示されるファイルシステム名を調べます。

3. ファイルシステムのバックアップを作成する際、`mount point` 列に表示されたディレクトリ名を使用します。

例 24–1 ファイルシステム名を検索する

この例では、バックアップ対象のファイルシステムはルート (`/`)、`/usr`、および `/export/home` です。

```
# more /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s1       -       -       swap    -       no      -
/dev/dsk/c0t0d0s0       /dev/rdsk/c0t0d0s0      /       ufs     1       no      -
/dev/dsk/c0t0d0s6       /dev/rdsk/c0t0d0s6      /usr    ufs     1       no      -
/dev/dsk/c0t0d0s7       /dev/rdsk/c0t0d0s7      /export/home    ufs     2       yes     -
/devices        -       /devices        devfs   -       no      -
sharefs -       /etc/dfs/sharetab       sharefs -       no      -
ctfs    -       /system/contract        ctfs    -       no      -
objfs   -       /system/object  objfs   -       no      -
swap    -       /tmp    tmpfs   -       yes     -
```

▼ 完全バックアップに必要なテープ数を決定する

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. バックアップのサイズをバイト単位で予測します。

   # ufsdump [*0*]S *file-system*

   ファイルシステムの 1 回目のバックアップ時には、S オプションを使用して、バックアップに必要な予想バイト数を表示します。

   ファイルシステムの 2 回目以降のバックアップ時には、0S オプションを使用して、バックアップに必要な予想バイト数を表示します。
3. 予測サイズをテープの容量で割り、必要なテープの数を決定します。

   テープの容量の一覧は、表 23–5 を参照してください。

例 24–2 テープの本数を決定する

次の例では、150M バイトのテープにファイルシステムが入ります。

```
# ufsdump S /export/home
178176
```

UFS ファイルシステムのバックアップ

バックアップを実行する際の一般的なガイドラインは次のとおりです。

- シングルユーザーモードを使用するか、ファイルシステムをマウント解除します (ファイルシステムのスナップショットを作成している場合を除く)。UFS スナップショットの詳細は、第 25 章「UFS スナップショットの使用 (手順)」を参照してください。

- ディレクトリレベルの処理 (ファイルの作成、削除、名前変更など) とファイルレベルの処理が同時に行われているときにファイルシステムのバックアップを実行すると、バックアップに組み込まれないデータがあるので注意してください。
- 単一のシステムで ufsdump コマンドを実行し、リモートシェルまたはリモートログインを通じてネットワーク上のシステムグループのバックアップをリモート実行できます。また、出力をテープデバイスがあるシステムに転送できます。通常、テープデバイスは ufsdump コマンドを実行するシステム上にありますが、ほかの場所にあってもかまいません。

  ファイルのバックアップをリモートデバイスに作成するには、ufsdump コマンドの出力を dd コマンドにパイプする方法もあります。dd コマンドの使用方法については、第 28 章「UFS ファイルとファイルシステムのコピー (手順)」を参照してください。
- ネットワーク上でリモートバックアップを実行する場合、テープデバイスを持つシステムの /.rhosts ファイル中には、デバイスを使用する各クライアントのエントリが入っている必要があります。また、バックアップを実行する各システムの /.rhosts ファイルには、バックアップを開始するシステムのエントリが入っている必要があります。

▼ UFS ファイルシステムのバックアップをテープに作成する方法

ufsdump コマンドを使用してファイルシステムのバックアップを作成する一般的な手順を示します。この例では、オプションと引数の使用方法を示しています。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 システムをレベル S (シングルユーザーモード) に移行します。

次に例を示します。

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

3 (省略可能) ファイルシステムの整合性を検査します。

次に例を示します。

```
# fsck -m /dev/rdsk/c0t0d0s7
```

fsck -m コマンドを実行すると、ファイルシステムの整合性が検査されます。たとえば、電源障害が発生すると、ファイルが不整合になることがあります。fsck コマンドの詳細は、第 21 章「UFS ファイルシステムの整合性検査 (手順)」を参照してください。

4 ファイルシステムをリモートテープドライブにバックアップする場合は、次の手順を実行します。

a. テープドライブが接続されているシステム (テープサーバー) の `./rhosts` ファイルに、次のエントリを追加します。

host `root`

host エントリには、`ufsdump` コマンドを実行してバックアップを行うシステムの名前を指定します。

b. テープサーバー上で、上記の `/.rhosts` ファイルに追加したホストに、ネームサービス経由でアクセスできることを確認します。

5 テープドライブのデバイス名を確認します。

デフォルトのテープドライブは、`/dev/rmt/0` です。

6 書き込み可能なテープをテープドライブに挿入します。

7 システムのバックアップを実行します。

```
# ufsdump options arguments filenames
```

ファイルシステムやディレクトリ、またはファイルシステム内のファイルをバックアップできます。ファイルを個別にバックアップする方法については、`tar(1)` または `cpio(1)` のマニュアルページを参照してください。

次の例では、もっとも一般的な `ufsdump` オプションおよび引数の使用方法を示します。

- 例 24–3
- 例 24–4
- 例 24–5
- 例 24–6

その他の `ufsdump` オプションおよび引数については、第 27 章「UFS バックアップおよび復元コマンド (参照情報)」を参照してください。

8 プロンプトが表示されたら、テープを取り出して、次のテープを挿入します。

9 各テープにボリューム番号、ダンプレベル、日付、システム名、ディスクスライス、ファイルシステム名を記入したラベルを貼ります。

10 **Control + D** キーを押してシステムをレベル **3** の動作に戻します。

11 バックアップが正常に実行されたことを確認します。

```
# ufsrestore tf device-name
```

例 24–3 UFS ルート (/) ファイルシステムの完全バックアップの実行

次の例では、ルート (/) ファイルシステムの完全バックアップを実行する方法を示します。この例では、バックアップの実行前にシステムをシングルユーザーモードにしています。次の ufsdump オプションが含まれます。

- 0 は、レベル 0 のダンプ (完全バックアップ) であることを示します。
- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示します。
- c は、カートリッジテープデバイスを示します。
- f /dev/rmt/0 は、テープデバイスを示します。
- / はバックアップするファイルシステムを示します。

次に例を示します。

```
# init 0
ok boot -s
.
.
.
# ufsdump 0ucf /dev/rmt/0 /
  DUMP: Date of this level 0 dump: Mon Jul 12 10:00:50 2010
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 1843276 blocks (900.04MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 1843252 blocks (900.03MB) on 1 volume at 6602 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Mon Jul 12 10:22:50 MDT 2010
# ufsrestore tf /dev/rmt/0
          2      .
          3      ./lost+found
          4      ./usr
          5      ./var
          6      ./export
          7      ./bin
          8      ./dev
.
.
.
# (Press Control-D to bring system to run level 3)
```

例 24–4 UFS ルート (/) ファイルシステムの増分バックアップの実行

次の例は、ルート (/) ファイルシステムのシングルユーザーモードでの増分バックアップの方法を示します。次の ufsdump オプションが含まれます。

- 9 は、レベル 9 のダンプ (増分バックアップ) であることを示します。

- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示します。
- c は、カートリッジテープデバイスを示します。
- f /dev/rmt/0 は、テープデバイスを示します。
- / はバックアップするファイルシステムを示します。

```
# init 0
ok boot -s
# ufsdump 9ucf /dev/rmt/0 /
  DUMP: Date of this level 9 dump: Mon Jul 12 10:52:50 2010
  DUMP: Date of last level 0 dump: Mon Jul 12 10:50:50 2010
  DUMP: Dumping /dev/rdsk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 422 blocks (211KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 376 blocks (188KB) on 1 volume at 1843 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 9 dump on Mon Jul 12 10:55:50 2010
# ufsrestore tf /dev/rmt/0
         2      .
         9      ./etc
       304      ./etc/dumpdates
      1117      ./devices
      1118      ./devices/pseudo
      3381      ./devices/pseudo/pts@0:1
.
.
.

# init 0
ok boot -s
# ufsdump 9ucf /dev/rmt/0 /
  DUMP: Date of this level 9 dump: Mon Jul 12 10:33:14 2010
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c2t1d0s0 (v880-brm-03:/root) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 120694 blocks (58.93MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 120580 blocks (58.88MB) on 1 volume at 9026 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 9 dump on Mon Jul 12 10:33:14 2010
# ufsrestore tf /dev/rmt/0
         2      .
         9      ./etc
       304      ./etc/dumpdates
      1117      ./devices
      1118      ./devices/pseudo
```

```
      3381      ./devices/pseudo/pts@0:1
.
.
.
```

例 24–5 UFS ホームディレクトリの完全バックアップ

次の例では、/export/home/kryten ホームディレクトリの完全バックアップを実行する方法を示します。次の ufsdump オプションが含まれます。

- 0 は、レベル 0 のダンプ (完全バックアップ) であることを示します。
- u は、このバックアップの実行日に /etc/dumpdates ファイルが更新されることを示します。
- c は、カートリッジテープデバイスを示します。
- f /dev/rmt/0 は、テープデバイスを示します。
- /export/home/kryten は、バックアップするディレクトリを示します。

```
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
  DUMP: Date of this level 0 dump: Wed Jul 10:35:18 2010
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (starbug:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 105158 blocks (51.35MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 105082 blocks (51.31MB) on 1 volume at 5025 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 9 dump on Mon Jul 12 10:38:14 2010
# ufsrestore tf /dev/rmt/0
       232      ./kryten
       233      ./kryten/filea
       234      ./kryten/fileb
       235      ./kryten/filec
       236      ./kryten/letters
       237      ./kryten/letters/letter1
       238      ./kryten/letters/letter2
       239      ./kryten/letters/letter3
       240      ./kryten/reports
       241      ./kryten/reports/reportA
       242      ./kryten/reports/reportB
       243      ./kryten/reports/reportC
```

例 24–6 リモートシステムへの完全バックアップの実行 (Solaris 10 データを Solaris 10 システムへ)

次の例は、Solaris 10 システム (mars) 上のローカルの /export/home ファイルシステムをリモートの Solaris 10 システム (earth) へ、シングルユーザーモードで完全バックアップする方法を示します。リモートアクセスが有効になるようにシステムを設定する必要があります。次の ufsdump オプションが含まれます。

- `0` は、レベル `0` のダンプ (完全バックアップ) であることを示します。
- `u` は、このバックアップの実行日に `/etc/dumpdates` ファイルが更新されることを示します。
- `c` は、カートリッジテープデバイスを示します。
- `f earth:/dev/rmt/0` は、リモートシステム名およびテープデバイスを示します。
- `/export/home` は、バックアップするファイルシステムを示します。

```
# ufsdump 0ucf earth:/dev/rmt/0 /export/home
  DUMP: Date of this level 0 dump: Mon Jul 12 10:36:30 MDT 2010
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (mars:/export/home) to earth:/dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 105172 blocks (51.35MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 105082 blocks (51.31MB) on 1 volume at 4425 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Mon Jul 12 10:39:30 MDT 2010
# ufsrestore tf earth:/dev/rmt/0
         2      .
         3      ./lost+found
         4      ./kryten
         5      ./kryten/filea
         6      ./kryten/fileb
         7      ./kryten/filec
         8      ./kryten/letters
         9      ./kryten/letters/letter1
        10      ./kryten/letters/letter2
        11      ./kryten/letters/letter3
        12      ./kryten/reports
.
.
.
```

◆ ◆ ◆ 第 25 章

UFS スナップショットの使用 (手順)

この章では、UFS スナップショットの作成およびバックアップ方法について説明します。

UFS スナップショットの作成に関連した手順については、489 ページの「UFS スナップショットの使用 (作業マップ)」を参照してください。

バックアップの実行方法の概要については、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。

UFS スナップショットの使用 (作業マップ)

作業	説明	参照先
1. UFS スナップショットを作成します。	`fssnap` コマンドを使用して、ファイルシステムの読み取り専用コピーを作成します。	494 ページの「UFS スナップショットを作成する方法」
2. UFS スナップショットの情報を表示します。	UFS スナップショット情報 (raw スナップショットデバイスなど) を確認します。	495 ページの「UFS スナップショットの情報を表示する方法」
3. (省略可能) UFS スナップショットを削除します。	バックアップ済みまたは不要になったスナップショットを削除します。	496 ページの「UFS スナップショットを削除する方法」
4. (省略可能) UFS スナップショットのバックアップを作成します。	次の方法のいずれかを選択します。	

作業	説明	参照先
	ufsdump コマンドを使用して、UFS スナップショットの完全バックアップを作成します。	497 ページの「UFS スナップショットの完全バックアップの作成方法 (ufsdump)」
	ufsdump コマンドを使用して、UFS スナップショットの増分バックアップを作成します。	498 ページの「UFS スナップショットの増分バックアップの作成方法 (ufsdump)」
	tar コマンドを使用して、UFS スナップショットのバックアップを作成します。	498 ページの「UFS スナップショットのバックアップ方法 (tar)」
5. (省略可能) UFS スナップショットからデータを復元します。	ufsrestore コマンドを使用して、データを復元するのと同じ方法で、UFS スナップショットを復元します。	509 ページの「UFS ファイルシステム全体を復元する方法」

UFS スナップショットの概要

fssnap コマンドを使って、ファイルシステムをマウントした状態で、そのバックアップを作成できます。作成されるのは、ファイルシステムの読み取り専用スナップショットです。「スナップショット」は、バックアップ操作のためのファイルシステムの一時イメージです。

fssnap コマンドを実行すると、1 つの仮想デバイスと 1 つのバッキングストアファイルが作成されます。ユーザーは、既存の Solaris バックアップコマンドを使用して、実際のデバイスのように動作し実際のデバイスのように見える「仮想デバイス」をバックアップできます。「バッキングストアファイル」は、スナップショット作成後に変更されたデータのコピーを含むビットマップファイルです。

バッキングストアファイルを指定する際には、次の注意点に留意してください。

- バッキングストアファイルの格納先は、対象ファイルシステムのデータを格納できるだけの空き容量を備えている必要があります。バッキングストアファイルのサイズは、対象ファイルシステム上における活動量に応じて変わります。
- バッキングストアファイルの格納場所は、スナップショット作成対象のファイルシステム上であってはなりません。
- バッキングストアファイルの格納場所としては、別の UFS ファイルシステムや NFS ファイルシステムなど、任意の種類のファイルシステムを選択可能です。
- 512G バイトを超える UFS ファイルシステムのスナップショットを作成する場合、複数のバッキングストアファイルが作成されます。

- バッキングストアファイルは空白ファイルです。`ls` コマンドで表示される空白ファイルの論理サイズは、`du` コマンドで表示されるその割り当て容量と同一ではありません。

512G バイトを超える UFS ファイルシステムのスナップショットを作成する方法の詳細は、493 ページの「マルチテラバイト UFS のスナップショットの作成」を参照してください。

なぜ UFS スナップショットを使用するか

UFS スナップショット機能により、ファイルシステムのバックアップがより利用しやすく、簡単になりました。この機能を利用すれば、ファイルシステムをマウントした状態でバックアップを作成できます。また、このときマルチユーザーモードを保持できます。さらに、長期保存が必要な場合は、`tar` や `cpio` コマンドを使って、UFS スナップショットをテープにバックアップすることもできます。`ufsdump` コマンドでバックアップを実行する場合は、シングルユーザーモードにして、バックアップが完了するまでファイルシステムを非アクティブな状態にする必要があります。

`fssnap` コマンドは、企業レベルでないシステム管理者に、大規模な記憶容量を必要とせずに、Sun StorEdge Instant Image のような企業レベルのツールのパワーを提供します。

UFS スナップショットの機能は、Instant Image 製品の機能に似ています。UFS スナップショットは大規模なファイルシステムをコピーできますが、企業レベルのシステムには Instant Image の方が適しています。UFS スナップショットは、小さめのシステムに適しています。Instant Image は、取り込まれるファイルシステム全体のサイズに等しい容量を割り当てます。ただし、UFS スナップショットが作成するバッキングストアファイルは、必要なディスク容量だけを占有します。

次の表は、UFS スナップショットと Instant Image との特徴的な違いを示します。

UFS スナップショット	Sun StorEdge Instant Image
バッキングストアファイルのサイズは、スナップショットがとられた後のデータの変更量によります	バッキングストアファイルのサイズは、コピーされるファイルシステム全体のサイズと同じです
システムのリブート後は保持されません	システムのリブート後も保持されます
UFS ファイルシステムで動作します	ルート (/) または `/usr` ファイルシステムでは使用できません
Solaris 8 1/01 以降のリリースで使用できます	Sun StorEdge 製品の一部です

UFS スナップショットのパフォーマンス上の問題

UFS スナップショットの初回作成時に、ファイルシステムのユーザーが短い一時停止に気づく場合があります。一時停止の時間は、取り込まれるファイルシステムのサイズに応じて増加します。スナップショットがアクティブな間、ファイルシステムの書き込み中、パフォーマンスに若干の影響が出る可能性があります。ただし、ファイルシステムの読み込み中には、このような影響はありません。

UFS スナップショットの作成と削除

`fssnap` コマンドを使用して UFS スナップショットを作成する場合、バッキングストアファイルがどれだけのディスク容量を消費するかを監視してください。バッキングストアファイルは、はじめは容量をまったく使用しませんが、その後特によく使用されているシステムにおいて急速に拡大します。バッキングストアファイルが拡大するのに十分な容量を必ず確保してください。または `-o maxsize=`*n* `[k,m,g]` オプション (*n* `[k,m,g]` はバッキングストアファイル最大限のサイズ) でそのサイズを制限してください。

注意 – バッキングストアファイルに容量が不足する場合、スナップショットが削除されてしまうことがあり、バックアップが失敗します。スナップショットのエラーの可能性を調べるため、`/var/adm/messages` ファイルを検査してください。

バッキングストアパスのディレクトリを指定して、バッキングストアファイルは指定のディレクトリに作成することもできます。たとえば、`/var/tmp` を指定した場合、次のバッキングストアファイルが作成されます。

```
/var/tmp/snapshot0
```

`/export/home`、`/usr` などのファイルシステムを個別に作成する代わりに大容量のルート (`/`) ファイルシステムを 1 つ作成した場合、これらのファイルシステムのスナップショットを個別に作成することはできません。たとえば、このシステムは、`/usr` 用の個別のファイルシステムを持ちません。次の `Mounted on` カラムの下を参照してください。

```
# df -k /usr
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0    3618177 2190002 1391994    62%    /
```

`/usr` ファイルシステムのスナップショットを作成しようとすると、次のようなメッセージが表示されます。

```
# fssnap -F ufs -o bs=/snaps/usr.back.file /usr
snapshot error: Invalid backing file path
```

ここで、スナップショットを作成するファイルシステム (この例の場合、/usr ファイルシステム) 上にバッキングストアファイルを作成することはできません。

詳細は、fssnap_ufs(1M) のマニュアルページを参照してください。

マルチテラバイト UFS のスナップショットの作成

マルチテラバイト UFS のスナップショットの作成は、より小さいサイズの UFS ファイルシステムに対するスナップショットの作成とほとんど同じです。唯一の違いは、ファイルシステム領域の 512G バイトごとにバッキングストアファイルが作成されるという点です。

512G バイトを超えるファイルシステムのスナップショットを作成する際には、次の注意点に留意してください。

- 複数のバッキングストアファイルが作成されます。
  - スナップショットの作成時にバッキングストアファイル名を指定した場合、その指定されたファイル名に基づく名前が後続のバッキングストアファイルに繰り返し付けられます。後続のバッキングストアファイルには同じ名前が付けられますが、「.2」、「.3」などの接尾辞がそれぞれ付加されます。
  - バッキングストアの名前は指定せず、その格納先 (ディレクトリ) だけを指定した場合、複数のバッキングストアファイル名が、「.2」、「.3」などの接尾辞付きで繰り返し作成されます。
- 複数のバッキングストアファイルが作成された場合でも、fssnap -i コマンドの実行時に表示されるのは、最初のバッキングストアファイルの名前だけです。これに対し、バッキングストア長として表示されるのは、スナップショットのすべてのバッキングストアファイルの合計サイズです。

  注 – バッキングストアファイルは空白ファイルです。ls コマンドで表示される空白ファイルの論理サイズは、du コマンドで表示されるその割り当て容量と同一ではありません。

- スナップショットの作成時に unlink オプションを指定しなかった場合、スナップショットのバックアップ完了後や単にスナップショットを削除するときに、バッキングストアファイルを手動で削除する必要があります。

512G バイトを超えるファイルシステムのスナップショットの作成例については、例 25–2 を参照してください。

詳細は、fssnap_ufs(1M) のマニュアルページを参照してください。

▼ UFS スナップショットを作成する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 ファイルシステムに、バッキングストアファイル用の十分なディスク容量が存在することを確認してください。

```
# df -k
```

3 同じ場所に同じ名前の既存のバッキングストアファイルがないことを確認します。

```
# ls /backing-store-file
```

4 UFS スナップショットを作成します。

```
# fssnap -F ufs -o bs=/backing-store-file /file-system
```

注 – バッキングストアファイルは、UFS スナップショットを使用して取り込むファイルシステムとは異なるファイルシステムに存在する必要があります。

---

5 スナップショットが作成されたことを確認します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

例 25–1 UFS スナップショットを作成する

次の例は、/usr ファイルシステムのスナップショットを作成する方法を示します。バッキングストアファイルは /scratch/usr.back.file です。仮想デバイスは /dev/fssnap/1 です。

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

次の例は、バッキングストアファイルを 500M バイトに制限する方法を示します。

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

例 25–2 マルチテラバイト UFS のスナップショットの作成

次の例は、1.6T バイトの UFS ファイルシステムのスナップショットの作成方法を示しています。

```
# fssnap -F ufs -o bs=/var/tmp /datab
/dev/fssnap/2
# /usr/lib/fs/ufs/fssnap -i /datab
Snapshot number               : 2
Block Device                  : /dev/fssnap/2
Raw Device                    : /dev/rfssnap/2
```

```
Mount point                   : /datab
Device state                  : idle
Backing store path            : /var/tmp/snapshot3
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Mon Jul 12 10:37:50 2010
Copy-on-write granularity     : 32 KB
```

▼ UFS スナップショットの情報を表示する方法

`fssnap -i` オプションを使用すると、システムの現在のスナップショットを表示できます。1 つのファイルシステムを指定する場合、そのファイルシステムのスナップショットについての詳細な情報が表示されます。特定のファイルシステムを指定しない場合は、現在の UFS スナップショットすべておよび対応する仮想デバイスの情報が表示されます。

注 – 次の例に示すように、拡張スナップショット情報を表示する場合は、UFS ファイルシステム固有の `fssnap` コマンドを使用してください。

---

1. スーパーユーザーになるか、同等の役割を引き受けます。

2. 現在のすべてのスナップショットをリスト表示します。

   次に例を示します。

```
# /usr/lib/fs/ufs/fssnap -i
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /export/home
Device state                  : idle
Backing store path            : /var/tmp/snapshot0
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Mon Jul 12 10:37:50 2010
Copy-on-write granularity     : 32 KB
```

3. 特定のスナップショットについての詳細な情報を表示します。

   次に例を示します。

```
# /usr/lib/fs/ufs/fssnap -i /export
Snapshot number               : 1
Block Device                  : /dev/fssnap/1
Raw Device                    : /dev/rfssnap/1
Mount point                   : /export
Device state                  : idle
Backing store path            : /var/tmp/snapshot1
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Mon Jul 12 10:37:50 2010
Copy-on-write granularity     : 32 KB
```

UFS スナップショットの削除

UFS スナップショットを作成する際、バッキングストアファイルがリンクされないように指定できます。リンクされていないバッキングストアファイルは、スナップショットを削除した後で削除されます。UFS スナップショットを作成する際に -o unlink オプションを指定しない場合は、あとでバッキングストアファイルを手動で削除する必要があります。

バッキングストアファイルは、バッキングストアファイルの削除に -o unlink オプションを使用した場合はスナップショットが削除されるまで、またはバッキングストアファイルを手動で削除するまで、ディスク容量を使用します。

▼ UFS スナップショットを削除する方法

スナップショットは、システムをリブートするか、あるいは fssnap -d コマンドを使用して削除できます。このコマンドを使用するときには、UFS スナップショットを含むファイルシステムのパスを指定する必要があります。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 削除するスナップショットを特定します。

   ```
   # /usr/lib/fs/ufs/fssnap -i
   ```

3. スナップショットを削除します。

   ```
   # fssnap -d /file-system
   Deleted snapshot 1.
   ```

4. スナップショットの作成時に -o unlink オプションを使用しなかった場合は、そのバッキングストアファイルを手動で削除してください。

   ```
   # rm /file-system/backing-store-file
   ```

例 25–3 UFS スナップショットの削除

次の例では、-o unlink オプションを使用しなかった場合にスナップショットを削除する方法を示します。

```
# fssnap -i
    0    /export/home
    1    /export
   # fssnap -d /export
 Deleted snapshot 1.
# rm /var/tmp/snapshot1
```

UFS スナップショットのバックアップ

UFS スナップショットの完全バックアップまたは増分バックアップを作成できます。UFS スナップショットのバックアップ作成に、標準の Solaris バックアップコマンドを使用できます。

UFS スナップショットを含む仮想デバイスは、標準の読み取り専用デバイスとして動作します。つまり、仮想デバイスを、ファイルシステムのデバイスをバックアップするかのようにバックアップできます。

ufsdump コマンドを使用して UFS スナップショットをバックアップする場合、バックアップ時にスナップショットの名前を指定できます。詳細は、次の手順を参照してください。

▼ UFS スナップショットの完全バックアップの作成方法 (ufsdump)

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 バックアップする UFS スナップショットを確認します。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

次に例を示します。

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number               : 1
Block Device                  : /dev/fssnap/1
Raw Device                    : /dev/rfssnap/1
Mount point                   : /usr
Device state                  : idle
Backing store path            : /var/tmp/snapshot2
Backing store size            : 544 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Mon Jul 12 10:37:50 2010
Copy-on-write granularity     : 32 KB
```

3 UFS スナップショットをバックアップします。

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

次に例を示します。

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4 スナップショットのバックアップが作成されたことを確認します。

次に例を示します。

```
# ufsrestore tf /dev/rmt/0
```

▼ UFS スナップショットの増分バックアップの作成方法 (ufsdump)

UFS スナップショットの増分バックアップでは、最後のスナップショット以降に変更のあったファイルだけがバックアップされます。ufsdump コマンドと N オプションを組み合わせて使用します。このオプションは、増分ダンプをトラックするために /etc/dumpdates ファイルに挿入されるファイルシステムのデバイス名を指定します。

次の例では、ufsdump コマンド内で fssnap コマンドを組み込んでファイルシステムの増分バックアップを作成しています。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. UFS スナップショットの増分バックアップを作成します。

   次に例を示します。

   ```
   # ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c0t1d0s0 'fssnap -F ufs -o raw,bs=
   /export/scratch,unlink /dev/rdsk/c0t1d0s0'
   ```

   上記の例では、ブロックデバイスではなく raw デバイスの名前を表示するために -o raw オプションが使用されています。このオプションの使用により、raw デバイスを必要とするコマンド (ufsdump コマンドなど) に fssnap コマンドを組み込むことが簡単になります。
3. スナップショットのバックアップが作成されたことを確認します。

   ```
   # ufsrestore ta /dev/rmt/0
   ```

▼ UFS スナップショットのバックアップ方法 (tar)

tar コマンドを使用してスナップショットをバックアップする場合、バックアップを行う前にスナップショットをマウントします。

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. スナップショット用のマウントポイントを作成します。

   次に例を示します。

   ```
   # mkdir /backups/home.bkup
   ```
3. スナップショットをマウントします。

   ```
   # mount -F ufs -o ro /dev/fssnap/1 /backups/home.bkup
   ```
4. マウントスナップショットのディレクトリに移動します。

   ```
   # cd /backups/home.bkup
   ```

5 tar コマンドを使用して、スナップショットをバックアップします。

```
# tar cvf /dev/rmt/0 .
```

UFS スナップショットのバックアップからのデータの復元

仮想デバイスから作成されたバックアップは、基本的には、スナップショットがとられた時点でのオリジナルのファイルシステムの状態を表しています。ファイルシステムをバックアップから復元する場合は、オリジナルのファイルシステムから直接そのバックアップを作成したかのように復元します。そのような復元では、ufsrestore コマンドを使用します。ufsrestore コマンドを使用してファイルまたはファイルシステムを復元する方法については、第 26 章「UFS ファイルとファイルシステムの復元 (手順)」を参照してください。

◆ ◆ ◆ 第 26 章

UFS ファイルとファイルシステムの復元 (手順)

この章では、ufsdump コマンドを使用してバックアップを作成したファイルおよびファイルシステムを、ufsrestore コマンドを使用して復元する方法について説明します。

ファイルおよびファイルシステムの復元に関連した手順については、501 ページの「UFS ファイルおよびファイルシステムのバックアップを復元する (作業マップ)」を参照してください。

ファイルやファイルシステムの保存、復元、コピー、移動などに使用するその他のコマンドについては、第 28 章「UFS ファイルとファイルシステムのコピー (手順)」を参照してください。

ファイルシステムのバックアップと復元については、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。

UFS ファイルおよびファイルシステムのバックアップを復元する (作業マップ)

次の作業マップに、ファイルおよびファイルシステムの復元手順を示します。

作業	説明	参照先
ファイルとファイルシステムを復元するための準備を行います。	復元するファイルシステムやファイル、テープデバイス、およびそれらの復元方法を決定します。	502 ページの「UFS ファイルとファイルシステムを復元するための準備」
使用するテープを決定します。	バックアップテープを調べ、復元するファイルまたはファイルシステムを含む最新のバックアップの日付を確認します。	504 ページの「使用するテープを決定する方法」

作業	説明	参照先
ファイルを復元します。	次のいずれかの復元方法を選択します。	
	ファイルの対話的な復元 – メディアの内容を参照してファイルおよびディレクトリ単位で選択できるため、正確なファイル名を把握していない場合、この方法を使用します。	505 ページの「対話式で UFS ファイルを復元する方法」
	ファイルの非対話的な復元 – 復元するファイル名をすでに把握している場合には、この方法を使用します。	507 ページの「特定の UFS ファイルを復元する方法」
	ファイルシステムの復元 – 復元時の手順の一部として新規ディスクドライブを使用できる場合は、この方法を使用します。	509 ページの「UFS ファイルシステム全体を復元する方法」
ルート (/) または /usr ファイルシステムを復元します。	ルート (/) または /usr ファイルシステムを復元する場合には、ローカルの Solaris DVD またはネットワークからシステムを起動する必要があります。	513 ページの「UFS ルート (/) および /usr ファイルシステムを復元する方法」

UFS ファイルとファイルシステムを復元するための準備

ufsrestore コマンドは、ufsdump コマンドを使用して作成されたバックアップから、ディスクの現在の作業ディレクトリにファイルをコピーします。ufsrestore コマンドを使用すると、レベル 0 のダンプとそれ以降の増分ダンプからファイルシステム階層全体を読み込み直すことができます。また、このコマンドを使用して、任意のバックアップテープから個々のファイルを復元することもできます。ufsrestore コマンドをスーパーユーザーとして実行した場合には、ファイルの所有者、最新の変更時刻、モード (ファイルのアクセス権) は元のまま、ファイルが復元されます。

ファイルまたはファイルシステムの復元を開始する前に、次の点を確認してください。

- 復元に必要なテープ (またはフロッピーディスク)
- ファイルシステム全体を復元する raw デバイス名
- 使用するテープデバイスのタイプ
- テープデバイスのデバイス名 (ローカルまたはリモート)

UFS ファイルシステム名の確認

バックアップテープに適切な名前が付いている場合は、テープラベルに入っているファイルシステム名 (/dev/rdsk/*device-name*) が使えるはずです。詳細については、480 ページの「UFS ファイルシステム名を検索する方法」を参照してください。

必要なテープデバイスのタイプの決定

ファイルを復元するには、バックアップメディアと互換性のあるテープデバイスを使用する必要があります。バックアップメディアの形式によって、ファイルの復元にどんなドライブを使用しなければならないかが決まります。たとえば、使用するバックアップメディアが 8mm テープの場合、ファイルの復元には 8mm テープデバイスを使わなければなりません。

テープデバイス名の決定

テープデバイス名 (/dev/rmt/*n*) をバックアップテープラベル情報の一部として指定している可能性があります。同じドライブを使ってバックアップテープを復元しようとする場合には、ラベル内にあるデバイス名を使うことができます。メディアデバイスおよびデバイス名の詳細については、第 29 章「テープドライブの管理 (手順)」を参照してください。

UFS ファイルとファイルシステムの復元

バックアップを実行すると、ファイルやディレクトリは、それらが含まれるファイルシステムからの相対的な位置に保存されます。ファイルとディレクトリを復元するときは、ufsrestore コマンドが現在の作業ディレクトリにファイル階層を作成し直します。

たとえば、/export/doc/books ディレクトリ (/export はファイルシステム) からバックアップされたファイルは、/export からの相対的な位置に保存されます。つまり、books ディレクトリ内の book ファイルは、テープ上で ./doc/books/book1. として保存されます。あとで、./doc/books/book1 ファイルを /var/tmp ディレクトリに復元する場合、そのファイルは /var/tmp/doc/books/book1 に復元されます。

個別のファイルやディレクトリを復元するときには、/var/tmp などの一時的な場所に復元する必要があります。ファイルを確認したら、それを適当な位置に移動させてもかまいません。ただし、個別のファイルやディレクトリはそれぞれ元の位置に復元できます。その場合には、新しいファイルをバックアップテープからの古いバージョンで上書きしないかどうか確かめてください。

ほかのユーザーとの重複を防ぐために、/var/tmp/restore ファイルなどのサブディレクトリを作成し、そのディレクトリに移動して、ファイルを復元することをお勧めします。

階層を復元する場合、ファイルを配置するファイルシステム内の一時ディレクトリにファイルを復元する必要があります。復元後に、mv コマンドを使用して階層全体を適切な場所に移動します。

注 – 一時的な場合でも、/tmp ディレクトリにファイルを復元してはいけません。/tmp ディレクトリは通常、TMPFS ファイルシステムとしてマウントされます。TMPFS は ACL などの UFS ファイルシステム属性をサポートしていません。

▼ 使用するテープを決定する方法

1. 復元するファイルを最後に変更したのはいつごろだったかをユーザーに尋ねます。

2. バックアップ計画を参照し、該当するファイルまたはファイルシステムを含む最後のバックアップの日付を調べます。

   ファイルの最新バージョンを検索するには、特にユーザーの要求がない限り、増分バックアップファイルを最高のダンプレベルから最低のダンプレベルへ、最新の日付からもっとも古い日付へ逆方向に調べます。

3. オンラインアーカイブファイルを保持している場合、使用に適したメディアを判別します。

   # ufsrestore ta *archive-name ./path/filename ./path/filename*

t	テープ上の各ファイルを表示します。
a	内容一覧をテープからではなく、オンラインアーカイブファイルから読み取ります。
archive-name	オンラインアーカイブファイル名を指定します。
./path/filename	オンラインアーカイブ上で検索するファイル名を指定します。コマンドが成功した場合は、ufsrestore は i ノード番号とファイル名を出力します。成功しなかった場合、ufsrestore はエラーメッセージを出力します。

   詳細は、ufsrestore(1M) のマニュアルページを参照してください。

4. 復元するファイルを含むメディアをドライブに挿入し、適切なメディアであることを確認します。

   # ufsrestore tf /dev/rmt/*n ./path/filename ./path/filename*

各ファイル名に必ず完全パス名を使用してください。ファイルがバックアップに入っていれば、その名前と i ノード番号が表示されます。ファイルがバックアップに入っていない場合、ファイルがそのボリュームに入っていないことを示すメッセージが表示されます。

5 同じテープに複数のバックアップファイルが入っている場合は、使用するバックアップファイルの位置までテープを移動します。

```
# ufsrestore xfs /dev/rmt/n tape-number
```

例 **26–1** 使用するテープを決定する

次の例は、オンラインアーカイブに /etc/passwd ファイルが存在するかどうかを確認する方法を示します。

```
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

次の例は、バックアップテープに /etc/passwd ファイルが存在するかどうかを確認する方法を示します。

```
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

▼ 対話式で UFS ファイルを復元する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 (省略可能) 安全性のため、書き込み保護を設定します。

3 ボリューム 1 のテープをテープドライブに挿入します。

4 ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5 対話式で復元を開始します。

```
# ufsrestore if /dev/rmt/n
```

情報を伝えるためのメッセージと ufsrestore> プロンプトが表示されます。

6 復元するファイルのリストを作成します。

a. ディレクトリの内容を表示します。

```
ufsrestore> ls [directory-name]
```

b. ディレクトリを変更します。

```
ufsrestore> cd directory-name
```

c. 復元するファイルとディレクトリのリストを作成します。

```
ufsrestore> add filenames
```

d. (省略可能) 必要に応じ、復元するファイルのリストからディレクトリまたはファイルを削除します。

```
ufsrestore> delete filename
```

7 (省略可能) 復元処理中にファイル名を表示します。

```
ufsrestore> verbose
```

8 ファイルを復元します。

```
ufsrestore> extract
```

どのボリューム番号を使用するかを指定するプロンプトが表示されます。

9 ボリューム番号を入力して、**Return** キーを押します。ボリュームが **1** つしかない場合には、1 を入力して **Return** キーを押します。

```
Specify next volume #: 1
```

リスト内のファイルとディレクトリが抽出され、現在の作業ディレクトリに復元されます。

10 現在の作業ディレクトリのモードを変更しない場合は、`set owner/mode` プロンプトが表示されたときに n を入力します。

```
set owner/mode for '.'? [yn] n
```

`ufsrestore` コマンドによる最後の整理が完了すると、ufsrestore のプロンプトが表示されます。

11 `ufsrestore` プログラムを終了します。

```
ufsrestore> quit
```

シェルプロンプトが表示されます。

12 復元されたファイルを確認します。

a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

b. リストを検査して、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

13 ファイルを適切なディレクトリに移動します。

例 26–2 対話式で UFS ファイルを復元する

次の例では、バックアップテープから /etc/passwd ファイルと /etc/shadow ファイルを抽出する方法について説明します。

```
# cd /var/tmp
# ufsrestore if /dev/rmt/0
ufsrestore> ls
.:
 .:
 .sunw/        export/       net/          sbin/         usr/
 Sources/      etools/       opt/          scde/         var/
 b/            home/         ptools/       set/          vol/
 bin           kernel/       pkg/          share/
 dev/          lib/          platform/     shared/
 devices/      lost+found/   proc/         src/
 etc/          mnt/          rtools/       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
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

▼ 特定の UFS ファイルを復元する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 (省略可能) 安全性のため、書き込み保護を設定します。

3 ボリューム 1 のテープをテープドライブに挿入します。

4 ファイルを一時的に復元するためのディレクトリに移動します。

```
# cd /var/tmp
```

5 ファイルを復元します。

```
# ufsrestore xvf /dev/rmt/n filename
```

x	*filename* 引数内に指定されたファイルまたはディレクトリをコピーするように ufsrestore コマンドに指定します。
v	復元処理中にファイル名を表示します。
f /dev/rmt/*n*	テープデバイス名を識別します。
filename	1 つまたは複数のファイル名またはディレクトリ名を指定します。複数の場合は、空白で区切って指定します。次に例を示します。 ./export/home/user1/mail ./export/home/user2/mail .

6 ファイルが入っているボリューム番号を入力します。**Return** キーを押します。

```
Specify next volume #: 1
```

ファイルは現在の作業ディレクトリに復元されます。

7 現在のディレクトリのモードを変更しない場合は、set owner/mode プロンプトが表示されたときに n と入力して **Return** キーを押します。

```
set owner/mode for '.'? [yn] n
```

8 復元されたファイルを確認します。

a. 復元されたファイルとディレクトリを表示します。

```
# ls -l
```

ファイルとディレクトリのリストが表示されます。

b. リストを検査して、リストに指定したすべてのファイルとディレクトリが復元されていることを確認します。

9 ファイルを適切なディレクトリに移動します。

例 26–3 特定のファイルを復元する

次の例では、passwd ファイルと shadow ファイルを /var/tmp ディレクトリに非対話式で復元する方法を説明します。

```
# cd /var/tmp
# ufsrestore xvf /dev/rmt/0 ./etc/passwd ./etc/shadow
Verify volume and initialize maps
Media block size is 126
Dump   date: Wed Jul 14 13:36:17 2010
Dumped from: the epoch
Level 0 dump of / on starbug:/dev/dsk/c0t0d0s0
Label: none
```

```
Extract directories from tape
Initialize symbol table.
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./etc/passwd
extract file ./etc/shadow
Add links
Set directory mode, owner, and times.
Specify next volume #:1
extract file ./etc/passwd
extract file ./etc/shadow
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

例 26–4 リモートのテープデバイスからファイルを復元する

`ufsrestore` コマンドを使用するときにテープデバイス名の前に *remote-host*: を追加することにより、ファイルをリモートドライブから復元できます。

次の例では、システム `venus` 上のリモートテープドライブ `/dev/rmt/0` を使用してファイルを復元します。

```
# ufsrestore xf venus:/dev/rmt/0 ./etc/hosts
```

▼ UFS ファイルシステム全体を復元する方法

ときには、全面的に復元しなければならないほどファイルシステムが破壊される場合があります。一般的な例として、ディスク障害が発生した場合には、ファイルシステムを全面的に復元する必要があります。この場合、ハードウェアを交換してからソフトウェアを復元しなければならないこともあります。ディスクの交換方法については、239 ページの「SPARC: システムディスクまたは二次ディスクの追加 (作業マップ)」または251 ページの「x86: システムディスクまたは二次ディスクの追加 (作業マップ)」を参照してください。

`/export/home` などのファイルシステムを全面的に復元するには、時間がかかります。ファイルシステムを一貫性のある方法でバックアップしていれば、最後の増分バックアップ時の状態に復元できます。

注 – ルート (/) または /usr ファイルシステムの復元には、この手順は使用できません。これらのファイルシステムを復元する方法については、513 ページの「UFS ルート (/) および /usr ファイルシステムを復元する方法」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 必要に応じ、ファイルシステムをマウント解除します。

```
# umount /dev/rdsk/device-name
```

または

```
# umount /file-system
```

3 ファイルシステムを新規作成します。

```
# newfs /dev/rdsk/device-name
```

raw デバイス上に新しいファイルシステムを構築するかどうかを確認するプロンプトが表示されます。意図しないファイルシステムを間違って損失してしまわないように、*device-name* が正しいことを確認します。

詳細は、newfs(1M) のマニュアルページを参照してください。

4 新しいファイルシステムを作成しなければならないかどうかを確認します。

```
newfs: construct a new file system /dev/rdsk/cwtxdysz:(y/n)? y
```

新しいファイルシステムが作成されます。

5 新しいファイルシステムを一時的なマウントポイントにマウントします。

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

6 マウントポイントのディレクトリに移動します。

```
# cd /mnt
```

7 (省略可能) 安全性のため、書き込み保護を設定します。

8 レベル 0 テープの第 1 ボリュームをテープドライブに挿入します。

9 ファイルを復元します。

```
# ufsrestore rvf /dev/rmt/n
```

ダンプレベル 0 のバックアップが復元されます。バックアップの実行に複数のテープが必要な場合は、番号の順にテープをロードするようにプロンプトが表示されます。

10 テープを外し、次のレベルのテープをドライブにロードします。
テープの復元は必ずダンプレベル 0 から始め、一番高いダンプレベルまで続けてください。

11 ダンプレベルごとに、一番低いレベルから一番高いレベルまで、手順 8 から手順 10 までの操作を繰り返します。

12 ファイルシステムが復元されたことを確認します。

```
# ls
```

13 restoresymtable ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために ufsrestore コマンドが作成し、使用した restoresymtable ファイルを削除します。

14 別のディレクトリに移動します。

```
# cd /
```

15 新しく復元されたファイルシステムをマウント解除します。

```
# umount /mnt
```

16 最後のテープを取り出し、書き込み保護されていない新しいテープをテープドライブに挿入します。

17 新しく復元されたファイルシステムのレベル 0 のバックアップを作成します。

```
# ufsdump 0ucf /dev/rmt/n /dev/rdsk/device-name
```

レベル 0 のバックアップが実行されます。ufsrestore コマンドはファイルの位置を移動し、i ノード割り当てを変更するので、新しく作成されたファイルシステムの完全バックアップは、必ずすぐに作成するようにしてください。

18 復元されたファイルシステムをマウントします。

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

復元されたファイルシステムがマウントされ、使用できるようになります。

19 復元およびマウントされたファイルシステムが使用できることを確認します。

```
# ls mount-point
```

例 26–5 UFS ファイルシステム全体を復元する

次の例では、/export/home ファイルシステムを復元する方法を示します。

```
# newfs /dev/rdsk/c0t0d0s7
newfs: /dev/rdsk/c0t0d0s7 last mounted as /export/home
newfs: construct a new file system /dev/rdsk/c0t0d0s7: (y/n)? y
/dev/rdsk/c0t0d0s7:     143358064 sectors in 23334 cylinders of 48 tracks, 128 sectors
69999.1MB in 1459 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
............................
super-block backups for last 10 cylinder groups at:
 142447776, 142546208, 142644640, 142743072, 142841504, 142939936, 143038368,
 143136800, 143235232, 143333664
# mount /dev/dsk/c0t0d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Wed Jul 14 12:45:33 2010
Dumped from: the epoch
Level 0 dump of /export/home on starbug:/dev/dsk/c0t0d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./rimmer
Make node ./rimmer/wdir
Make node ./lister
Make node ./pmorph
Make node ./inquisitor
Make node ./kryten
Make node ./kryten/letters
Make node ./kryten/reports
Extract new leaves.
Check pointing the restore
extract file ./rimmer/words
extract file ./rimmer/words1
extract file ./rimmer/words2
extract file ./rimmer/words3
extract file ./rimmer/wdir/words
extract file ./rimmer/wdir/words1
extract file ./rimmer/wdir/words2
extract file ./rimmer/wdir/words3
.
.
.
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# rm restoresymtable
# cd /
# umount /mnt
# ufsdump 0ucf /dev/rmt/0 /export/home
                    .
                    .
                    .
```

```
# mount /dev/dsk/c0t0d0s7 /export/home
# ls /export/home
```

▼ UFS ルート (/) および /usr ファイルシステムを復元する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 新しいシステムディスクを、ルート (/) と /usr ファイルシステムが復元されるシステムに追加します。

   システムディスクの追加方法の詳細は、241 ページの「SPARC: システムディスクを接続してブートする方法」または 253 ページの「x86: システムディスクの接続方法」を参照してください。
3. 新しいファイルシステムを一時的なマウントポイントにマウントします。

   ```
   # mount /dev/dsk/device-name /mnt
   ```
4. /mnt ディレクトリに移動します。

   ```
   # cd /mnt
   ```
5. (省略可能) 安全性のため、書き込み保護を設定します。
6. テープデバイスのリンクを作成します。

   ```
   # tapes
   ```
7. ルート (/) ファイルシステムを復元します。

   ```
   # ufsrestore rvf /dev/rmt/n
   ```

   ダンプレベル 0 のテープが復元されます。
8. テープを外し、次のレベルのテープをドライブにロードします。

   テープの復元は必ずダンプレベル 0 から始め、もっとも低いダンプレベルからもっとも高いダンプレベルまで続けてください。
9. 必要に応じ、復元を続行します。

   ```
   # ufsrestore rvf /dev/rmt/n
   ```

   次のレベルのテープが復元されます。
10. テープを追加するたびに、手順 8 と手順 9 を繰り返します。

11 ファイルシステムが復元されたことを確認します。

```
# ls
```

12 restoresymtable ファイルを削除します。

```
# rm restoresymtable
```

復元のチェックポイントのために ufsrestore コマンドが作成し、使用した restoresymtable ファイルを削除します。

13 ルート (/) ディレクトリに変更します。

```
# cd /
```

14 新しく作成されたファイルシステムをマウント解除します。

```
# umount /mnt
```

15 新しいファイルシステムを検査します。

```
# fsck /dev/rdsk/device-name
```

復元されたファイルシステムの整合性が検査されます。

16 ルートパーティションにブートブロックを作成します。

SPARC システムの場合:

```
# installboot  /usr/platform/‘uname-i‘/lib/fs/ufs/bootblk
/dev/rdsk/device-name
```

詳細は、installboot(1M) のマニュアルページを参照してください。

SPARC システム上での installboot コマンドの使用例については、例 26–6 を参照してください。

x86 システムの場合:

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/cwtxdysz
```

詳細は、installgrub(1M) のマニュアルページを参照してください。

x86 システム上での installgrub コマンドの使用例については、例 26–7 を参照してください。

17 新しいテープをテープドライブに挿入します。

18 新しいファイルシステムのバックアップを作成します。

```
# ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
```

ダンプレベル 0 のバックアップが実行されます。ufsrestore コマンドはファイルの位置を移動し、i ノード割り当てを変更するので、新しく作成されたファイルシステムの完全バックアップは、必ずすぐに作成するようにしてください。

19 必要に応じ、/usr ファイルシステムについて手順 **5** から **16** を繰り返します。

20 システムを再起動します。

```
# init 6
```

システムがリブートされます。

例 26–6 SPARC: UFS ルート (/) ファイルシステムを復元する

次の例は、SPARC システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、システムはローカルの Solaris DVD またはネットワークから起動するものとします。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdsk/c0t3d0s0
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk
/dev/rdsk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6
```

例 26–7 x86: UFS ルート (/) ファイルシステムを復元する

次の例は、x86 システム上でルート (/) ファイルシステムを復元する方法を示します。この例では、GRUB フェイルセーフブートセッション、ローカルの Solaris DVD、またはネットワークからシステムが起動されていることを前提としています。

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdsk/c0t3d0s0
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t3d0s0
stage1 written to partition 0 sector 0 (abs 2016)
stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6
```

◆ ◆ ◆ 第 27 章

UFS バックアップおよび復元コマンド (参照情報)

この章では、ufsdump コマンドと ufsrestore コマンドの参照情報を示します。

この章で説明する情報は次のとおりです。

- 517 ページの「ufsdump コマンドの機能」
- 522 ページの「ufsdump コマンドオプションおよび引数の指定」
- 523 ページの「ufsdump とセキュリティーに関する注意事項」
- 523 ページの「ufsrestore オプションおよび引数の指定」

バックアップの実行方法の概要については、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。

バックアップの手順については、第 24 章「UFS ファイルとファイルシステムのバックアップ (手順)」を参照してください。

ufsdump コマンドの機能

ufsdump コマンドは、ファイルシステムのバックアップ作成時に 2 つのパスを作成します。最初のパスでは、このコマンドは raw デバイスファイル内でファイルシステムを走査し、メモリー内にディレクトリとファイルのテーブルを作成します。次に、そのテーブルをバックアップメディアに書き込みます。2 つ目のパスでは、ufsdump は i ノードに番号順にアクセスし、ファイルの内容を読み込んでバックアップメディアに書き込みます。

デバイス特性の判断

ufsdump コマンドに必要なことは、適切なテープブロックの大きさを認識することと、どのようにしてメディアの終わりを検出するかということです。

メディアの終わりの検出

ufsdump コマンドは、一連の固定長レコードを書き込みます。ufsdump コマンドは、レコードの一部にしか書き込まれていないという通知を受け取ると、メディアの物理的な終わりに達したものと判断します。この方法は、ほとんどのデバイスに有効です。部分的なレコードしか書き込まれなかったことをデバイスが ufsdump に通知できない場合、ufsdump が別のレコードの書き込みを試みると、メディアエラーが発生します。

注 – DAT デバイスと 8mm テープデバイスでは、メディアの終わりが検出されます。カートリッジテープデバイスと 1/2 インチテープデバイスでは、メディアの終わりは検出されません。

---

ufsdump は、ほとんどのデバイスのメディアの終わりを自動的に検出します。したがって、通常は -c、-d、-s、-t オプションを使用しなくても、複数のボリュームのバックアップを実行できます。

ufsdump コマンドで、デバイスがメディアの終わりを検出する方法を確認できない場合は、メディアの終わりに関するオプションを使用する必要があります。

restore コマンドとの互換性を確保するため、サイズオプションを使用すると、従来どおり、現在のテープやフロッピーディスクの終わりに達する前に、ufsdump を次のテープやフロッピーディスクに強制的に進ませることができます。

ufsdump コマンドを使用したデータのコピー

ufsdump コマンドは、raw ディスクスライスからデータのみをコピーします。ファイルシステムがまだ有効であれば、メモリーバッファー内のデータがコピーされていない可能性があります。ufsdump コマンドによるバックアップでは、空きブロックはコピーされず、ディスクスライスのイメージも作成されません。シンボリックリンクがほかのスライス上のファイルを指す場合は、リンク自体がコピーされます。

/etc/dumpdates ファイルの目的

ufsdump コマンドを -u オプション付きで使用すると、/etc/dumpdates というファイルを管理し、更新できます。/etc/dumpdates ファイル内の各行は、次の情報を表しています。

- バックアップが作成されたファイルシステム
- 前回実行されたバックアップのダンプレベル
- バックアップの曜日、日付、および時刻

次に例を示します。

```
# cat /etc/dumpdates
/dev/rdsk/c0t0d0s0              0 Wed Jul 7 13:36:17 2010
/dev/rdsk/c0t0d0s7              0 Thu Jul 8 12:53:12 2010
/dev/rdsk/c0t0d0s7              9 Thu Jul 8 13:41:48 2010
```

増分バックアップの実行時に、ufsdump コマンドは /etc/dumpdates ファイルを検査して、下のダンプレベルの最後のバックアップ日付を調べます。次に、下のレベルのバックアップ以降に更新されたすべてのファイルをメディアにコピーします。バックアップが完了すると、完了したばかりのバックアップを記述する新しい情報行によって、そのレベルの最後のバックアップの情報行が置き換えられます。

/etc/dumpdates ファイルを使用して、バックアップが実行中であるかどうかを検査してください。機器に問題が発生している場合は、この検査が特に重要です。機器の障害が原因でバックアップを完了できないと、そのバックアップは /etc/dumpdates ファイルに記録されません。

ディスク全体を復元する必要があれば、/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検査できるので、ファイルシステム全体を復元するために必要なファイルを判断できます。

注 – /etc/dumpdates ファイルは、編集可能なテキストファイルです。ただし、編集するかどうかはユーザーの判断によります。ファイルに変更を加えた結果、アーカイブテープと一致しなくなると、必要なテープ (またはファイル) がどれであるかわからなくなることがあります。

バックアップデバイス (*dump-file*) 引数

dump-file 引数 (-f オプションで使用) では、バックアップ先を指定します。バックアップ先は、次のいずれかになります。

- ローカルのテープドライブ
- ローカルのフロッピーディスクドライブ
- リモートのテープドライブ
- リモートのフロッピーディスクドライブ
- 標準出力

この引数は、バックアップ先がデフォルトのローカルテープドライブ /dev/rmt/0 でないときに使用します。-f オプションを使用する場合は、*dump-file* 引数の値を指定しなければなりません。

注 – *dump-file* 引数では、ローカルディスクまたはリモートディスク上のファイルを指すこともできます。誤用するとファイルシステムがいっぱいになる可能性があります。

ローカルのテープドライブまたはフロッピーディスクドライブ

通常、*dump-file* 引数には、テープデバイスかフロッピーディスク用の raw デバイスファイルを指定します。ufsdump コマンドは、出力デバイスへの書き込み時にバックアップファイルを 1 つ作成しますが、このファイルは複数のテープやフロッピーディスクにまたがってもかまいません。

デバイスの省略形を使用して、システム上のテープデバイスかフロッピーディスクデバイスを指定します。最初のデバイスは常に 0 です。たとえば、SCSI テープコントローラが 1 つと、中密度の形式を使用する QIC-24 テープドライブが 1 つある場合は、次のデバイス名を使用します。

```
/dev/rmt/0m
```

テープデバイス名を指定するときは、名前の末尾に文字 n を付けて、バックアップの完了後にテープドライブを巻き戻さないように指定することもできます。次に例を示します。

```
/dev/rmt/0mn
```

テープに複数のファイルを格納する場合は、no-rewind オプションを使用します。バックアップ中に領域を使い果たすと、ufsdump コマンドから新しいテープの挿入を促すプロンプトが表示されるまで、テープは巻き戻されません。デバイスの命名規則の詳細は、550 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

次の形式で、リモートのテープデバイスまたはフロッピーディスクを指定します。*host: device*。ローカルシステム上のスーパーユーザーがリモートシステムへのアクセス権を持っている場合、ufsdump コマンドはリモートデバイスに書き込みます。通常、スーパーユーザーとして ufsdump コマンドを実行するのであれば、ローカルシステム名をリモートシステムの /.rhosts ファイルに記述しておく必要があります。デバイスを *user@host*: *device* と指定した場合、ufsdump コマンドは指定されたユーザーでリモートシステム上のデバイスへのアクセスを試みます。この場合、指定されたユーザーの名前が、リモートシステム上の /.rhosts ファイル中に含まれている必要があります。

デバイスには、ufsdump コマンドを実行するシステムではなく、そのデバイスが存在するシステムのオペレーティングシステムに合った命名規則を使用してください。デバイスが SunOS の旧バージョン (4.1.1 など) を実行するシステム上にある場合は、SunOS 4.1 でのデバイス名 (/dev/rst0 など) を使用します。システムが Solaris ソフトウェアを実行中の場合は、SunOS 5.9 でのデバイス名 (/dev/rmt/0 など) を使用します。

ufsdump コマンドで標準出力を使用する

dump-file 引数としてダッシュ (-) を指定すると、ufsdump コマンドは標準出力に書き込みます。

注 – *dump-file* 引数として標準出力を指定すると、 -v オプション (検査) は機能しません。

ufsdump コマンドを使用して標準出力に書き込み、ufsrestore コマンドを使用して標準入力から読み込むと、パイプライン内でファイルシステムをコピーできます。次に例を示します。

```
# ufsdump 0f - /dev/rdsk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

バックアップを作成するファイルを指定する

コマンド行の最後の引数として、バックアップするファイル (*filenames*) を必ず指定してください。この引数は、バックアップのコピー元または内容を指定します。

ファイルシステムの場合、次のように raw デバイスファイルを指定します。

/dev/rdsk/*c0t0d0s7*

ファイルシステムは、そのエントリが /etc/vfstab ファイルに存在すれば、マウントポイントディレクトリ (/export/home など) を使用して指定できます。

デバイスの命名規則の詳細は、550 ページの「バックアップデバイス名」を参照してください。

個々のファイルやディレクトリごとに、1 つまたは複数の名前を空白で区切って入力します。

注 – ufsdump コマンドを使用して (ファイルシステム全体ではなく) 1 つまたは複数のディレクトリやファイルのバックアップを作成するときには、レベル 0 のバックアップが実行されます。増分バックアップは適用されません。

テープの性質を指定する

テープの性質を指定しなければ、ufsdump コマンドはデフォルト設定を使用します。テープカートリッジ (c)、密度 (d)、サイズ (s)、トラック数 (t) を指定できます。オプションの順序とその引数の順番が一致していれば、オプションはいくつでも指定できます。

ufsdump コマンドの制限

次に、ufsdump コマンドでは実行できない操作を示します。

- ファイルシステムのバックアップに必要なテープやフロッピーディスクの数を自動的に計算する。仮実行 (ドライラン) モード (S オプション) を使用すると、実際にファイルシステムをバックアップする前に必要な容量を判定できます。
- 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、組み込みエラー検査機能を提供する。
- サーバーからリモートにマウントされたファイルをバックアップする。サーバー上のファイルのバックアップは、そのサーバー上で実行しなければなりません。ユーザーがサーバー上で所有するファイル上で ufsdump コマンドを実行するアクセス権は拒否されます。

ufsdump コマンドオプションおよび引数の指定

この節では、ufsdump コマンドのオプションと引数の指定方法について説明します。ufsdump コマンドの構文を、次に示します。

/usr/sbin/ufsdump *options arguments filenames*

options 1 文字のオプション名からなる 1 つの文字列です。

arguments オプションの引数を指定します。複数の文字列も指定できます。オプション文字とそれに関連する引数は、同じ順序で並べる必要があります。

filenames バックアップするファイルを指定します。これらの引数は、空白で区切り、常に最後に指定します。

ufsdump のデフォルトオプション

オプションを指定せずに ufsdump コマンドを実行する場合は、次の構文を使用します。

 # ufsdump *filenames*

ufsdump コマンドでは、デフォルトで次のオプションと引数が使用されます。

ufsdump 9uf /dev/rmt/0 *filenames*

これらのオプションでは、デフォルトのテープドライブ上にその推奨密度でレベル 9 の増分バックアップが作成されます。

ufsdump オプションについては、ufsdump(1M) のマニュアルページを参照してください。

ufsdump とセキュリティーに関する注意事項

セキュリティー保護を適用するには、次の操作を実行する必要があります。

- ufsdump コマンドの実行には、スーパーユーザーのアクセス権を必要とします。
- 集中バックアップを実行する場合は、クライアント上とサーバー上の /.rhosts ファイルからスーパーユーザーアクセスのエントリを削除します。

  セキュリティーに関する一般的な情報は、『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

ufsrestore オプションおよび引数の指定

ufsrestore コマンドの構文を、次に示します。

/usr/sbin/ufsrestore *options arguments filenames*

options 1 文字のオプション名からなる 1 つの文字列です。次のオプションから 1 つだけ選択します。i、r、R、t、または x。ufsrestore オプションについては、ufsrestore(1M) のマニュアルページを参照してください。

arguments オプションに対応する引数です。オプション文字とそれに関連する引数は、同じ順序で並べる必要があります。

filenames 復元するファイルを、x または t オプションの引数として指定します。これらの引数は、空白で区切り、常に最後に指定します。

◆ ◆ ◆ 第 28 章

UFS ファイルとファイルシステムのコピー (手順)

この章では、各種のバックアップコマンドを使用して、UFS ファイルとファイルシステムをディスク、テープ、フロッピーディスクにコピーする方法について説明します。

この章で説明する手順は次のとおりです。

- 529 ページの「ディスクをコピーする方法 (dd)」
- 533 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)」
- 535 ページの「ファイルをテープにコピーする方法 (tar)」
- 536 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)」
- 536 ページの「テープからファイルを取り出す方法 (tar)」
- 537 ページの「pax コマンドを使用してファイルをテープにコピーする」
- 538 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)」
- 539 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)」
- 540 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)」
- 541 ページの「テープから特定のファイルを取り出す方法 (cpio)」
- 542 ページの「ファイルをリモートテープデバイスにコピーする方法 (tar と dd)」
- 543 ページの「ファイルをリモートテープデバイスから抽出する方法」
- 545 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)」
- 546 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)」
- 546 ページの「ファイルをフロッピーディスクから取り出す方法 (tar)」

ファイルシステムをコピーするためのコマンド

UFS ファイルシステム全体をバックアップして復元する必要があるときは、第 27 章「UFS バックアップおよび復元コマンド (参照情報)」で説明されている ufsdump および ufsrestore コマンドを使用します。個々のファイル、ファイルシステムの一部、またはファイルシステム全体をコピーまたは移動する場合は、ufsdump と ufsrestore コマンドの代わりに、この章で説明する手順を使用できます。

次の表に、各種バックアップコマンドの用途を示します。

表 28–1 バックアップコマンドの用途

作業	コマンド	参照先
UFS ファイルシステムをテープにバックアップします。	ufsdump	482 ページの「UFS ファイルシステムのバックアップをテープに作成する方法」
ファイルシステムのスナップショットを作成します。	fssnap	第 25 章「UFS スナップショットの使用 (手順)」
UFS ファイルシステムをテープから復元します。	ufsrestore コマンド	509 ページの「UFS ファイルシステム全体を復元する方法」
ファイルをほかのシステムに転送します。	pax、tar、または cpio	534 ページの「ファイルとファイルシステムをテープにコピーする」
ファイルまたはファイルシステムをディスク間でコピーします。	dd	529 ページの「ディスクをコピーする方法 (dd)」
ファイルをフロッピーディスクにコピーします。	tar	545 ページの「ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)」

次の表に、各種のバックアップおよび復元コマンドを示します。

表 28–2 バックアップコマンドの概要

コマンド名	ファイルシステム境界の認識	複数ボリュームバックアップのサポート	物理コピー / 論理コピー
volcopy	はい	はい	物理
tar	いいえ	いいえ	論理
cpio	いいえ	はい	論理
pax	はい	はい	論理

表 28–2 バックアップコマンドの概要 (続き)

コマンド名	ファイルシステム境界の認識	複数ボリュームバックアップのサポート	物理コピー/論理コピー
dd	はい	いいえ	物理
ufsdump/ufsrestore	はい	はい	論理
fssnap	なし	なし	論理

次の表では、これらのコマンドの一部の長所と短所を説明します。

表 28–3 tar、pax、cpio コマンドの長所と短所

コマンド	機能	長所	短所
tar	ファイルやディレクトリサブツリーを 1 本のテープにコピーする場合に使用します。	■ ほとんどの UNIX オペレーティングシステムで利用できます ■ パブリックドメインバージョンもすぐに利用できます	■ ファイルシステムの境界を認識しません ■ 完全パス名は最大 255 文字です ■ 複数のテープボリュームを作成する場合は使用できません
pax	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用します。または、POSIX 準拠システムとの間でファイルをコピーする場合に使用します。	■ POSIX 準拠システムに対する、tar コマンドや cpio コマンドよりも高い互換性 ■ 複数のベンダーサポート	tar コマンドの短所を参照してください。ただし、pax は、複数のテープボリュームを作成できます。

表 28–3 tar、pax、cpio コマンドの長所と短所 (続き)

コマンド	機能	長所	短所
cpio	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用します。または、最新の Solaris リリースを実行しているシステムから SunOS 4.0/4.1 リリースを実行しているシステムへファイルをコピーする場合に使用します。	■ tar コマンドよりも効率的に、データをテープに書き込みます ■ 復元時、テープ中の不良箇所をスキップします ■ 異なるシステムタイプ間の互換性のために、異なるヘッダー形式 (tar、ustar、crc、odc、bar など) でファイルを書き込むオプションを提供します ■ 複数のテープボリュームを作成します	コマンド構文が tar コマンドや pax コマンドよりも難解です。

次の節では、これらのコマンドの使用例を紹介します。

ファイルシステムをディスク間でコピーする

ファイルシステムをディスク間でコピーするには、次の 2 つのコマンドを使用します。

- volcopy
- dd

volcopy の詳細は、volcopy(1M) のマニュアルページを参照してください。

次の節では、dd コマンドを使用してファイルシステムをディスク間でコピーする方法について説明します。

ファイルシステムのリテラルコピーを作成する

dd コマンドでは、完全な UFS ファイルシステムのリテラル (ブロックレベル) コピーを別のファイルシステムやテープに作成します。デフォルトでは、dd コマンドはその標準入力を標準出力にコピーします。

注 – 可変長テープドライブで dd コマンドを使用するときは、必ず適切なブロックサイズを指定してください。

標準入力、標準出力、またはその両方の代わりに、デバイス名を指定できます。次の例では、フロッピーディスクの内容が /tmp ディレクトリ内のファイルにコピーされます。

```
$ dd < /floppy/floppy0 > /tmp/output.file
2400+0 records in
2400+0 records out
```

dd コマンドは、読み取りブロック数と書き込みブロック数をレポートします。+ の次の数値は、部分的にコピーされたブロックの数です。デフォルトのブロックサイズは 512 バイトです。

dd コマンドの構文は、ほかのほとんどのコマンドとは異なっています。オプションは *keyword=value* のペアで指定します。この場合、*keyword* は設定するオプションで、*value* はそのオプションの引数です。たとえば、標準入力と標準出力を次の構文に置き換えることができます。

```
$ dd if=input-file of=output-file
```

リダイレクト記号の代わりに *keyword=value* の形式で指定するには、次のように入力します。

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

▼ ディスクをコピーする方法 (dd)

ディスクをコピーする際には次の点に注意してください。

- ボリュームマネージャーの制御下にあるディスクをコピーする場合は、この手順を使わないでください。
- UFS ファイルシステムのデータをあるディスクまたはシステムから別のディスクまたはシステムへコピーする場合にもっとも多く使用されるのは、ufsdump コマンドおよび ufsrestore コマンドを使用する方法です。これらのコマンドの使用方法の詳細は、第 23 章「UFS ファイルシステムのバックアップと復元 (概要)」を参照してください。
- システムを複製するには、フラッシュアーカイブを作成し、それを複製先のシステムにコピーします。フラッシュアーカイブの作成方法の詳細は、『Oracle Solaris 10 9/10 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』を参照してください。
- EFI ディスクラベル付きディスクをコピーする場合は、例 28–2 を参照してください。

それでもなお、dd コマンドによるディスクコピーを検討する場合は、次の点に注意してください。

- コピー元とコピー先のディスクが、同じディスクジオメトリを保持していることを確認します。
- コピー対象の UFS ファイルシステムを fsck ユーティリティーを使って検査します。
- dd コマンドを使ってディスクをコピーする際に、システムがシングルユーザーモードになっていることを確認します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

2 (省略可能) システムがリブート時に必要に応じてコピー先ディスクを認識するように、/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。

```
# init 0
```

4 コピー先ディスクをシステムに接続します。

5 システムをブートします。

```
ok boot -s
```

6 ソースディスクをコピー先ディスクへコピーします。

```
# dd if=/dev/rdsk/device-name of=/dev/rdsk/device-name bs=block-size
```

if=/dev/rdsk/*device-name*	マスターディスクデバイスのオーバーラップスライスを指定します。通常はスライス 2 です。
of=/dev/rdsk/*device-name*	コピー先ディスクデバイスのオーバーラップスライスを指定します。通常はスライス 2 です。
bs=*blocksize*	ブロックサイズ (128K バイト、256K バイトなど) を指定できます。ブロックサイズの値を大きくすると、ディスクのコピーに要する時間を短縮できます。

詳細は、dd(1M) のマニュアルページを参照してください。

7 新しいファイルシステムを検査します。

```
# fsck /dev/rdsk/device-name
```

8 コピー先ディスクのルート (/) ファイルシステムをマウントします。

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

9 /etc/vfstab ファイルのあるディレクトリに移動します。

```
# cd /mnt/etc
```

10 テキストエディタを使用して、コピー先ディスクの /etc/vfstab ファイルを編集して、正しいデバイス名を参照するようにします。

たとえば、c0t3d0 のインスタンスをすべて c0t1d0 に変更します。

11 コピー先ディスクのルート (/) ディレクトリに移動します。

```
# cd /
```

12 コピー先ディスクのルート (/) ファイルシステムをマウント解除します。

```
# umount /mnt
```

13 システムをシャットダウンします。

```
# init 0
```

14 コピー先ディスクからシングルユーザーモードでブートします。

```
# boot disk n -s
```

注 – installboot コマンドをコピー先ディスクで実行する必要はありません。これは、ブートブロックがオーバーラップスライスの一部としてコピーされるためです。

---

15 コピー先ディスクの構成を解除します。

```
# sys-unconfig
```

構成を解除すると、システムが停止します。

16 再びコピー先ディスクからブートし、ホスト名や時間帯などのシステム情報を与えます。

```
# boot disk n
```

17 システムがブートしたら、スーパーユーザーとしてログインしてシステム情報を確認します。

hostname console login:

例 28–1 VTOC ラベル付きディスクをコピーする (dd)

次の例では、VTOC ラベル付きのマスターディスク /dev/rdsk/c0t0d0s2 をコピー先ディスク /dev/rdsk/c0t2d0s2 にコピーする方法を示します。

```
# touch /reconfigure
# init 0
```

```
ok boot
# dd if=/dev/rdsk/c0t0d0s2 of=/dev/rdsk/c0t2d0s2 bs=128k
# fsck /dev/rdsk/c0t2d0s2
# mount /dev/dsk/c0t2d0s2 /mnt
# cd /mnt/etc
# vi vfstab
(Modify entries for the new disk)
# cd /
# umount /mnt
# init 0
# boot disk2 -s
# sys-unconfig
# boot disk2
```

例 28–2 EFI ラベル付きディスクをコピーする (dd)

以前の Solaris リリースでは、ディスク全体がスライス 2 (s2) で表されていました。EFI ラベル付きディスクの場合、複製後のディスクに一意の UUID を割り当てる必要があるため、1T バイトを超えるサイズのディスクを複製 (コピー) する場合、若干異なる方法を使用する必要があります。複製済みディスク用の新しいラベルは、必ず作成してください。作成しないと、その他のソフトウェア製品によって UUID の重複が検出された時点で、EFI ラベル付きディスクのデータが破壊されることがあります。

次に例を示します。

1. EFI ラベル付きディスクを複製します。次に例を示します。

   ```
   # dd if=/dev/rdsk/c0t0d0 of=/dev/rdsk/c0t2d0 bs=128k
   ```

2. 複製対象のディスクの prtvtoc 出力を fmthard コマンドにパイプして、複製済みディスク用の新しいラベルを作成します。次に例を示します。

   ```
   # prtvtoc /dev/rdsk/c0t0d0 | fmthard -s - /dev/rdsk/c0t2d0
   ```

EFI ディスクラベルの詳細は、198 ページの「EFI ディスクラベル」を参照してください。

cpio を使用してファイルシステム間でディレクトリをコピーする

cpio (コピーインとコピーアウト) コマンドを使用して、個々のファイル、ファイルグループ、またはファイルシステム全体をコピーできます。この節では、cpio コマンドを使ってファイルシステム全体をコピーする方法について説明します。

cpio コマンドは、ファイルのリストを 1 つの大型出力ファイルにコピーするアーカイブプログラムです。また、復元しやすいように、個々のファイルの間に

ヘッダーを挿入します。cpio コマンドを使用すると、ファイルシステム全体を別のスライス、別のシステム、またはテープやフロッピーディスクなどのメディアデバイスにコピーできます。

cpio コマンドは、メディアの終わりを認識し、別のボリュームを挿入するように促すプロンプトを表示するので、複数のテープやフロッピーディスクが必要なアーカイブを作成するにはもっとも効率のよいコマンド (ufsdump 以外では) です。

cpio コマンドの使用時には、しばしば ls や find のコマンドを使用してコピーするファイルを選択し、その出力を cpio コマンドにパイプします。

▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)

1. スーパーユーザーになるか、同等の役割を引き受けます。
2. 目的のディレクトリに移動します。

   # cd *filesystem1*
3. find コマンドと cpio コマンドを組み合わせて使用して、*filesystem1* から *filesystem2* へディレクトリツリーをコピーします。

   # find . -print -depth | cpio -pdm *filesystem2*

   .　現在の作業ディレクトリで処理を始めます。

   -print　ファイル名を出力します。

   -depth　ディレクトリ階層を下降し、すべてのファイル名を出力します。

   -p　ファイルのリストを作成します。

   -d　必要に応じてディレクトリを作成します。

   -m　ディレクトリ上で正しい変更時刻を設定します。

   詳細は、cpio(1) のマニュアルページを参照してください。

   指定したディレクトリ名からファイルがコピーされます。シンボリックリンクは保持されます。

   また、-u オプションも指定できます。このオプションは、無条件にコピーを実行します。u オプションを指定しない場合、古いファイルが、新しいファイルで置換されません。このオプションは、ディレクトリごとコピーするときに、コピーするファイルの一部がすでにターゲットのディレクトリ中に存在する場合に便利です。

4 コピー先ディレクトリの内容を表示して、コピーに成功したかどうかを確認します。

```
# cd filesystem2
# ls
```

5 ソースディレクトリが不要な場合は削除します。

```
# rm -rf filesystem1
```

例 28–3 ファイルシステム間でディレクトリをコピーする (cpio)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

ファイルとファイルシステムをテープにコピーする

tar、pax、および cpio コマンドを使用すると、ファイルとファイルシステムをテープにコピーできます。どのコマンドを選択するかは、コピーする目的に応じて異なります。3 つのコマンドはすべて raw デバイスを使用するので、使用する前にテープ上でファイルシステムをフォーマットまたは作成する必要はありません。

使用するテープドライブとデバイス名は、各システムのハードウェアの構成によって異なります。テープデバイス名の詳細は、549 ページの「使用するメディアの選択」を参照してください。

tar を使用してファイルをテープにコピーする

tar コマンドを使用してファイルをテープにコピーする前に、次の情報について知っておかなければなりません。

- tar コマンドに -c オプションを指定してファイルをテープにコピーすると、テープに入っているすべての既存のファイルまたはテープの現存の位置以降にある既存のファイルすべてが破壊 (上書き) されます。
- ファイルをコピーするとき、ファイル名の一部にファイル名置換ワイルドカード文字 (? と *) を使用できます。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 と入力します。
- 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ません。

▼ ファイルをテープにコピーする方法 (tar)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

   ```
   $ tar cvf /dev/rmt/n filenames
   ```

c	アーカイブの作成を指定します。
v	各ファイルがアーカイブされるたびに、その名前を表示します。
f /dev/rmt/ *n*	アーカイブを指定したデバイスまたはファイルに書き込むように指定します。
filenames	コピーするファイルとディレクトリを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

   指定した名前のファイルがテープにコピーされ、テープ上の既存のファイルが上書きされます。
4. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。
5. コピーしたファイルがテープ上に存在することを確認します。

   ```
   $ tar tvf /dev/rmt/n
   ```

   tar テープ上のファイルを表示する方法については、536 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

例 28–4 ファイルをテープにコピーする (tar)

次の例では、3 つのファイルをテープドライブ 0 のテープにコピーします。

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/0
```

▼ テープ上のファイルのリストを表示する方法 (tar)

1. テープをテープドライブに挿入します。
2. テープの内容を表示します。

   ```
   $ tar tvf /dev/rmt/n
   ```

t	テープ上のファイルのリストが表示されます。
v	t オプションと一緒に使用すると、テープ上のファイルに関する詳細情報が表示されます。
f /dev/rmt/*n*	テープデバイスを示します。

例 28–5 テープ上のファイルのリストを表示する (tar)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ tar tvf /dev/rmt/0
drwxr-xr-x   0/0        0 Jul 14 13:50 2010 reports/
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportC
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportB
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportA
```

▼ 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へ移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルを取り出します。

   ```
   $ tar xvf /dev/rmt/n [filenames]
   ```

x	指定したアーカイブファイルからのファイルの抽出を指定します。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ます。
v	各ファイルを取り出すたびに、その名前を表示します。
f /dev/rmt/ *n*	アーカイブを含むテープデバイスを示します。
filenames	取り出すファイルを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

   詳細は、tar(1) のマニュアルページを参照してください。

4 ファイルがコピーされたことを確認します。

```
$ ls -l
```

例 28–6 テープ上のファイルを取り出す (tar)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

注意事項 テープから抽出されるファイル名は、アーカイブに格納されているファイル名と同一でなければなりません。ファイルの名前やパス名が不明な場合は、まずテープ上のファイルのリストを表示します。テープ上のファイルをリスト表示する方法については、536 ページの「テープ上のファイルのリストを表示する方法 (tar)」を参照してください。

pax コマンドを使用してファイルをテープにコピーする

▼ ファイルをテープにコピーする方法 (pax)

1 コピーするファイルの存在するディレクトリに移動します。

2 書き込み可能なテープをテープドライブに挿入します。

3 ファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/n filenames
```

`-w`	書き込みモードを有効にします。
`-f /dev/rmt/n`	テープドライブを識別します。
filenames	コピーするファイルとディレクトリを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

詳細は、pax(1) のマニュアルページを参照してください。

4 ファイルがテープにコピーされたことを確認します。

```
$ pax -f /dev/rmt/n
```

5 テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 28–7 ファイルをテープにコピーする (pax)

次の例は、pax コマンドを使用して、現在のディレクトリ内のファイルをすべてコピーする方法を示します。

```
$ pax -w -f /dev/rmt/0 .
$ pax -f /dev/rmt/0
filea fileb filec
```

cpio コマンドを使用してファイルをテープにコピーする

▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)

1 コピーするファイルの存在するディレクトリに移動します。

2 書き込み可能なテープをテープドライブに挿入します。

3 ファイルをテープにコピーします。

```
$ ls | cpio -oc > /dev/rmt/n
```

ls	cpio コマンドにファイル名のリストを渡します。
cpio -oc	cpio コマンドがコピーアウトモード (-o) で動作し、ASCII 文字形式 (-c) でヘッダー情報を書き込むように指定します。これらのオプションによりほかのベンダーのシステムとの互換性を保ちます。
> /dev/rmt/n	出力ファイルを指定します。

ディレクトリ内のすべてのファイルは、指定したドライブ内のテープにコピーされ、テープ上の既存のファイルが上書きされます。コピーされた合計ブロック数が表示されます。

4 ファイルがテープにコピーされたことを確認します。

```
$ cpio -civt < /dev/rmt/n
```

-c	cpio コマンドがファイルを ASCII 文字形式で読み込むように指定します。

`-i`	cpio コマンドがコピーインモードで動作することを指定します。この時点ではファイルをリストするだけです。
`-v`	`ls -l` コマンドの出力と同様の形式で出力を表示します。
`-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リストを表示します。
`< /dev/rmt/`*n*	既存の cpio アーカイブの入力ファイルを指定します。

5 テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 28–8 ディレクトリ内のすべてのファイルをテープにコピーする (cpio)

次の例では、 `/export/home/kryten` ディレクトリ内のすべてのファイルをテープドライブ 0 のテープにコピーする方法を示します。

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
1280 blocks
$ cpio -civt < /dev/rmt/0
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filea
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, fileb
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filec
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, letters
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, reports
1280 blocks
```

▼ テープ上のファイルのリストを表示する方法 (cpio)

注 – テープの内容リストを表示するには、cpio コマンドがアーカイブ全体を処理する必要があるため、かなりの時間がかかります。

---

1 テープをテープドライブに挿入します。

2 テープ上のファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/n
```

例 28–9 テープ上のファイルのリストを表示する (cpio)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/0
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filea
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, fileb
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filec
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, letters
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, reports
1280 blocks
```

▼ テープからすべてのファイルを取り出す方法 (cpio)

相対パス名を使用してアーカイブを作成した場合、入力ファイルはそれを取り出すときに現在のディレクトリ内のディレクトリとして作成されます。ただし、絶対パス名を指定してアーカイブを作成した場合は、それと同じ絶対パス名を使用してシステム上でファイルが再び作成されます。

注意 – 絶対パス名を使用すると、自分のシステム上にある元のファイルを上書きすることになるので危険です。

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからすべてのファイルを抽出します。

   ```
   $ cpio -icvd < /dev/rmt/n
   ```

`-i`	標準入力からファイルを取り出します。
`-c`	`cpio` コマンドがファイルを ASCII 文字形式で読み込むように指定します。
`-v`	取り出されたファイルを `ls` コマンドの出力と同様の形式で表示します。
`-d`	必要に応じてディレクトリを作成します。
`< /dev/rmt/n`	出力ファイルを指定します。

4. ファイルがコピーされたことを確認します。

   ```
   $ ls -l
   ```

例 28–10 テープからすべてのファイルを取り出す (cpio)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

▼ テープから特定のファイルを取り出す方法 (cpio)

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルのサブセットを取り出します。

   ```
   $ cpio -icv "*file" < /dev/rmt/n
   ```

`-i`	標準入力からファイルを取り出します。
`-c`	cpio コマンドを使用してヘッダーを ASCII 文字形式で読み込むように指定します。
`-v`	取り出されたファイルを ls コマンドの出力と同様の形式で表示します。
"*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します。複数のパターンを指定できますが、個々のパターンを二重引用符で囲まなければなりません。
`< /dev/rmt/`*n*	入力ファイルを指定します。

   詳細は、cpio(1) のマニュアルページを参照してください。
4. ファイルがコピーされたことを確認します。

   ```
   $ ls -l
   ```

例 28–11 指定したファイルをテープから取り出す (cpio)

次の例では、末尾に接尾辞 chapter が付いているすべてのファイルをドライブ 0 のテープから取り出します。

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
```

```
31 blocks
$ ls -l
```

ファイルをリモートテープデバイスにコピーする

▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd)

1 リモートテープドライブを使用するには、次の前提条件を満たしている必要があります。

   a. ローカルホスト名 (および、オプションでコピーを実行するユーザーのユーザー名) が、リモートシステムの /etc/hosts.equiv ファイルに記述されている必要がある。または、コピーを実行するユーザーが、リモートマシン上の自分のホームディレクトリをアクセス可能にし、かつ $HOME/.rhosts 内にローカルマシン名を記述しておく必要がある。

      詳細は、hosts.equiv(4) のマニュアルページを参照してください。

   b. リモートシステムのエントリがローカルシステムの /etc/inet/hosts ファイル内またはネームサービスの hosts ファイル内になければならない。

2 リモートコマンドの実行に必要なアクセス権を保持していることを確認するには、次のように入力します。

```
$ rsh remotehost echo test
```

   test と表示された場合、リモートコマンドの実行に必要なアクセス権を保持しています。Permission denied と表示された場合は、手順 1 の内容を確認してください。

3 ファイルを置きたいディレクトリに移動します。

4 テープをテープドライブに挿入します。

5 ファイルをリモートテープドライブへコピーします。

```
$ tar cvf - filenames | rsh remote-host dd of=/dev/rmt/n obs=block-size
```

tar cf	テープアーカイブを作成し、アーカイブに含まれるファイルをリスト表示し、テープデバイスを指定します。
v	tar ファイルのエントリに関する追加情報を表示します。
- (ハイフン)	可変部としてテープデバイスの代わりに指定します。

filenames コピーするファイルを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

rsh | *remote-host* tar コマンドの出力をパイプを通してリモートシェルに渡します。

dd of= /dev/rmt/*n* 出力デバイスを指定します。

obs=*block-size* ブロック係数を指定します。

6 テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 **28–12** ファイルをリモートテープドライブにコピーする (tar と dd)

```
# tar cvf - * | rsh mercury dd of=/dev/rmt/0 obs=126b
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

▼ ファイルをリモートテープデバイスから抽出する方法

1 テープをテープドライブに挿入します。

2 一時ディレクトリに移動します。

```
$ cd /var/tmp
```

3 リモートテープデバイスからファイルを抽出します。

```
$ rsh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

rsh *remote-host* dd コマンドを使用してテープデバイスからファイルを取り出すために起動するリモートシェルです。

dd if=/dev/rmt/*n* 入力デバイスを指定します。

| tar xvBpf - dd コマンドの出力を tar コマンドにパイプして、ファイルを復元します。

4 ファイルが抽出されたことを確認します。

```
$ ls -l
```

例 **28–13** ファイルをリモートのテープドライブから抽出する

```
$ cd /var/tmp
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```

ファイルとファイルシステムをフロッピーディスクにコピーする

ファイルやファイルシステムをフロッピーディスクにコピーする前に、フロッピーディスクをフォーマットする必要があります。フロッピーディスクをフォーマットする方法については、第 2 章「リムーバブルメディアの管理 (手順)」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れば、cpio コマンドを使用します。cpio コマンドはメディアの終わりを認識し、次のフロッピーディスクの挿入を促すプロンプトを表示します。

ファイルをフロッピーディスクにコピーする際の注意事項

- tar に -c オプションを指定してフォーマット済みフロッピーディスクにファイルをコピーすると、フロッピーディスク上の既存のファイルは破壊 (上書き) されます。
- tar イメージを格納済みのフロッピーディスクはマウントできません。
- 複数のボリュームを扱う場合は、cpio コマンドを使用します。tar コマンドは 1 つのボリュームに対してのみ使用するユーティリティーです。

詳細は、tar(1) のマニュアルページを参照してください。

▼ ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする方法 (tar)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み保護されていないフォーマット済みフロッピーディスクをドライブに挿入します。
3. フロッピーディスクを使用可能な状態にします。

   ```
   $ volcheck
   ```

4. 必要に応じて、再度フォーマットします。

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

5. ファイルをフロッピーディスクにコピーします。

   ```
   $ tar cvf /vol/dev/aliases/floppy0 filenames
   ```

   指定した名前のファイルがフロッピーディスクにコピーされ、フロッピーディスク上の既存のファイルがすべて上書きされます。
6. ファイルがコピーされたことを確認します。

   ```
   $ tar tvf /vol/dev/aliases/floppy0
   ```

   ファイルのリストを表示する方法については、546 ページの「フロッピーディスク上のファイルのリストを表示する方法 (tar)」を参照してください。
7. フロッピーディスクをドライブから取り出します。
8. ファイル名をフロッピーディスクラベルに記入します。

例 28–14 ファイルを 1 枚のフォーマット済みフロッピーディスクにコピーする (tar)

次の例では、evaluation* という名前のファイルをフロッピーディスクにコピーする方法を示します。

```
$ cd /home/smith
$ volcheck
$ ls evaluation*
evaluation.doc   evaluation.doc.backup
$ tar cvf /vol/dev/aliases/floppy0 evaluation*
a evaluation.doc 86 blocks
```

```
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/aliases/floppy0
```

▼ フロッピーディスク上のファイルのリストを表示する方法 (tar)

1. フロッピーディスクをドライブに挿入します。
2. フロッピーディスクを使用可能な状態にします。

   ```
   $ volcheck
   ```

3. フロッピーディスク上のファイルのリストを表示します。

   ```
   $ tar tvf /vol/dev/aliases/floppy0
   ```

例 **28–15** フロッピーディスク上のファイルのリストを表示する (tar)

次の例では、フロッピーディスク上のファイルのリストを表示します。

```
$ volcheck
$ tar tvf /vol/dev/aliases/floppy0
rw-rw-rw-6693/10  44032 Jun  9 15:45 evaluation.doc
rw-rw-rw-6693/10  43008 Jun  9 15:55 evaluation.doc.backup
$
```

▼ ファイルをフロッピーディスクから取り出す方法 (tar)

1. ファイルを置きたいディレクトリに移動します。
2. フロッピーディスクをドライブに挿入します。
3. フロッピーディスクを使用可能な状態にします。

   ```
   $ volcheck
   ```

4. フロッピーディスクからファイルを取り出します。

   ```
   $ tar xvf /vol/dev/aliases/floppy0
   ```

   フロッピーディスク上のすべてのファイルが現在のディレクトリにコピーされます。

5 ファイルが取り出されたことを確認します。

```
$ ls -l
```

6 フロッピーディスクをドライブから取り出します。

例 28–16 ファイルをフロッピーディスクから取り出す (tar)

次の例では、フロッピーディスクからすべてのファイルを取り出します。

```
$ cd /home/smith/Evaluations
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

次の例では、フロッピーディスクから個々のファイルを取り出します。指定したファイルはフロッピーディスクから取り出され、現在の作業ディレクトリに格納されます。

```
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0 evaluation.doc
x evaluation.doc, 44032 bytes, 86 tape blocks
$ ls -l
```

複数のフロッピーディスクへファイルをアーカイブする

大量のファイルをフロッピーディスクにコピーする場合は、いっぱいになったフロッピーディスクを別のフォーマット済みフロッピーディスクと交換するように促すプロンプトを表示させることができます。cpio コマンドにはこの機能があります。使用する cpio コマンドはファイルをテープにコピーする場合と同じですが、テープデバイス名ではなく /vol/dev/aliases/floppy0 をデバイスとして指定する点が異なります。

cpio コマンドの使用方法については、538 ページの「ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)」を参照してください。

◆ ◆ ◆ 第 29 章

テープドライブの管理 (手順)

この章では、Solaris オペレーティングシステム (Oracle Solaris OS) でテープドライブを管理する方法について説明します。

この章で説明する手順は次のとおりです。

- 552 ページの「テープドライブの状態を表示する方法」
- 553 ページの「磁気テープカートリッジのたるみを直す」
- 554 ページの「磁気テープカートリッジを巻き戻す」

この章の内容は次のとおりです。

- 549 ページの「使用するメディアの選択」
- 550 ページの「バックアップデバイス名」
- 552 ページの「テープドライブの状態を表示する」
- 554 ページの「ドライブの管理とメディア処理のガイドライン」

使用するメディアの選択

通常は、次のテープメディアを使用して Solaris システムのバックアップを作成します。

- 1/2 インチのリールテープ
- 1/4 インチのストリームカートリッジテープ
- 8mm のカートリッジテープ
- 4mm のカートリッジテープ (DAT)

フロッピーディスクを使用してバックアップを実行することもできますが、時間がかかり煩雑です。

どのメディアを選択するかは、メディアをサポートする機器とファイルの格納に使用するメディア (通常はテープ) の可用性によって決まります。バックアップはローカルシステムから実行しなければなりませんが、ファイルはリモートデバイスに書き込めます。

次の表に、ファイルシステムのバックアップに使用する標準的なテープデバイスを示します。各デバイスの記憶容量は、ドライブのタイプとテープに書き込むデータのタイプによって異なります。

表 29–1 メディアの記憶容量

バックアップメディア	記憶容量
1/2 インチのリールテープ	140M バイト (6250 bpi)
2.5G バイト、1/4 インチのカートリッジ (QIC) テープ	2.5G バイト
DDS3 4mm カートリッジテープ (DAT)	12 - 24G バイト
14G バイト、8mm カートリッジテープ	14G バイト
DLT 7000 1/2 インチ カートリッジテープ	35 - 70G バイト

バックアップデバイス名

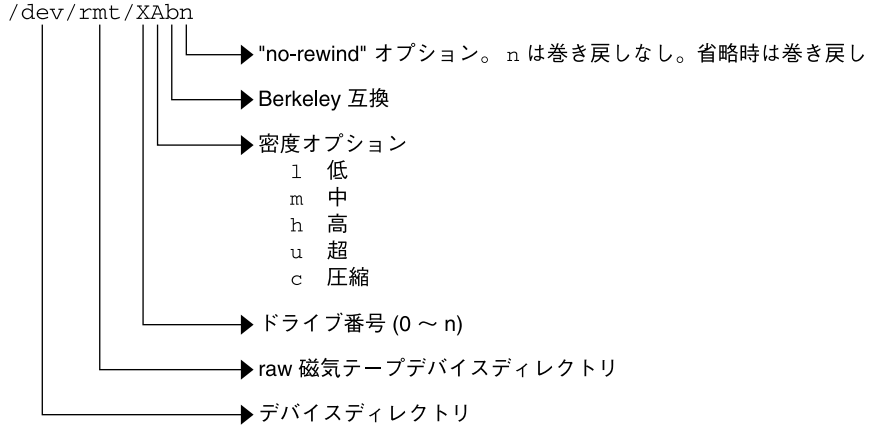
バックアップに使用するテープまたはフロッピーディスクドライブに論理デバイス名を与えて指定します。この名前は、「raw」デバイスファイルの格納されたサブディレクトリを指し、ドライブの論理ユニット番号が含まれます。テープドライブの命名規則に従い、物理デバイス名ではなく論理デバイス名を使用します。次の表に、この命名方式を示します。

表 29–2 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/`*n*
フロッピーディスク	`/vol/dev/rdiskette0/unlabeled`

通常は、次の図に示す方法で、テープデバイスを指定します。

図 29–1 テープドライブデバイス名



密度を指定しないと、テープドライブは通常その「推奨」密度で書き込みます。推奨密度は、一般にそのテープドライブでサポートされる最大密度です。ほとんどの SCSI ドライブはテープ上の密度やフォーマットを自動的に検出し、それに従って読み取りを実行します。ドライブでサポートされる密度を調べるには、`/dev/rmt` サブディレクトリを確認してください。このサブディレクトリには、各テープで異なる出力密度をサポートするためのテープデバイスファイルのセットが含まれます。

SCSI コントローラは、最大 7 台の SCSI テープドライブを持つこともできます。

テープドライブの巻き戻しオプションを指定する

通常は、テープドライブを 0 から *n* までの論理デバイス番号で指定します。次の表に、「巻き戻し」または「巻き戻しなし」のオプションを付けてテープデバイス名を指定する方法を示します。

表 29–3 テープドライブの「巻き戻し」または「巻き戻しなし」オプション

ドライブおよび巻き戻し	使用するオプション
第 1 のドライブ、巻き戻し	`/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`

テープドライブに別の密度を指定する

デフォルトでは、テープドライブはその「推奨」密度で書き込みますが、これは一般にそのテープドライブでサポートされる最大密度です。テープデバイスを指定しなければ、コマンドはデバイスでサポートされるデフォルト密度でドライブ番号 0 に書き込みます。

テープを特定の密度しかサポートされないテープドライブが付いたシステムに転送するには、目的の密度で書き込むデバイス名を指定します。次の表に、テープドライブに別の密度を指定する方法を示します。

表 29–4 テープドライブに別の密度を指定する

ドライブ、密度、巻き戻し	使用するオプション
第 1 のドライブ、低密度、巻き戻し	`/dev/rmt/0l`
第 1 のドライブ、低密度、巻き戻しなし	`/dev/rmt/0ln`
第 2 のドライブ、中密度、巻き戻し	`/dev/rmt/1m`
第 2 のドライブ、中密度、巻き戻しなし	`/dev/rmt/1mn`

密度のオプションについては、550 ページの「バックアップデバイス名」を参照してください。

テープドライブの状態を表示する

`mt` コマンドの `status` オプションを使用すると、テープドライブに関する状態情報を表示できます。`mt` コマンドは、`/kernel/drv/st.conf` ファイルに記述されているすべてのテープドライブの情報を表示します。

▼ テープドライブの状態を表示する方法

1. 情報を表示するドライブにテープをロードします。
2. テープドライブの状態を表示します。

   ```
   # mt -f /dev/rmt/n status
   ```

3. テープドライブ番号を **0**、**1**、**2**、**3** というように置き換えて手順 **1** と **2** を繰り返し、使用可能なすべてのテープドライブに関する情報を表示します。

例 29–1 テープドライブの状態を表示する

次の例は、QIC-150 テープドライブ (`/dev/rmt/0`) の状態を示します。

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 0
```

次の例は、Exabyte テープドライブ (/dev/rmt/1) の状態を示します。

```
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0  retries= 0
file no= 0   block no= 0
```

次の方法を使用すると、システムを手早くポーリングしてすべてのテープドライブを検査できます。

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

磁気テープカートリッジの取り扱い

テープの読み込み中にエラーが発生した場合は、テープのたるみを直し、テープドライブを掃除してからやり直してください。

磁気テープカートリッジのたるみを直す

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。

次に例を示します。

```
$ 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

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。

次に例を示します。

```
$ mt -f /dev/rmt/1 rewind
$
```

ドライブの管理とメディア処理のガイドライン

バックアップテープは読み込めなければ役に立ちません。このため、定期的に掃除および検査を行い、テープドライブが正常に動作するようにしてください。テープドライブのクリーニング手順については、ハードウェアのマニュアルを参照してください。次のいずれかの方法で、テープハードウェアを検査できます。

- テープにファイルをコピーし、ファイルを読み込んで、オリジナルのファイルとコピーしたファイルを比較します。
- ufsdump コマンドの -v オプションを使用して、メディアの内容をソースファイルシステムと比較して検査します。-v オプションを有効にするには、ファイルシステムをマウント解除するか、完全にアイドル状態にする必要があります。

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 23 章「UFS ファイルシステムのバックアップと復元 (概要)」で説明したバックアップ方法を採用する場合は、ラベルを「テープ A」、「テープ B」という具合に指定する必要があります。このラベルは変更しないでください。バックアップを実行するたびに、次の情報を記入した別のテープラベルを作成します。

- バックアップ日付
- マシン名およびバックアップを作成したファイルシステム
- バックアップレベル
- テープ番号 (バックアップが複数のボリュームにまたがる場合は *n* 本のうちの 1 本目)
- サイト特有の情報

テープは、磁気を発生させる機器から離れた埃のない安全な場所に保管してください。たとえば、アーカイブしたテープを遠隔地の防火キャビネットに保管します。

各ジョブ (バックアップ) がどのメディア (テープボリューム) に格納されているかということと、各バックアップファイルがどこに保管されているかを記録したログを作成し、管理する必要があります。

索引

数字・記号

1394 (FireWire) のサポート, 説明，71
4.3 Tahoe ファイルシステム，329
9660 CD フォーマット，47

A

AutoFS，351-352

B

BSD Fat Fast ファイルシステム，329

C

`cachefspack` コマンド
  概要，400
  使用方法，400
CacheFS ファイルシステム
  `cachefspack` エラーの障害追跡，405
  `cachefspack` コマンドによるパック (概要)，400
  `cachefspack` コマンドによるパック (手順)，400
  CacheFS 統計情報の表示，413
  CacheFS の統計情報の収集 (概要)，409
  CacheFS ロギングの設定 (手順)，410
  CacheFS ロギングの停止，412
  CacheFS ログファイルの場所の調査，411
  `fsck` コマンドによる検査 (例)，399
  (概要)，387
  削除 (手順)，397
CacheFS ファイルシステム *(続き)*
  作成 (手順)，389
  情報の表示(手順)，395
  パッキングリストの作成 (手順)，403
  パックされたファイルの表示 (手順)，401
  パックされたファイルの表示 (例)，401
  パラメータ，387
  マウント (手順)，390
CD
  ISO 9660 フォーマット，47
  UFS CD
    SPARC vs. x86 フォーマット，47
  名前，46
`cdrw` コマンド
  アクセスの制限 (手順)，59
  説明，55
  データ CD、データ DVD、およびオーディオ CD の書き込み (概要)，58
`cfgadm`
  PCI ホットプラグ (概要)，90
  SCSI ホットプラグ (概要)，90
`cfsadmin` コマンド，389, 397
`clri` コマンド，336
`cpio` コマンド
  (概要)，532
  テープからすべてのファイルを抽出 (手順)，540
  テープ上のファイルのリスト (手順)，539
  ファイルシステム間でのディレクトリのコピー (手順)，533
CTFS ファイルシステム，334

D

datadm コマンド，189
dd コマンド
  (概要)，528
  ディスクの複製 (手順)，530
/dev/dsk ディレクトリ，85
/dev/rdsk ディレクトリ，85
devfsadm コマンド，84
dfstab ファイル, 共有されたローカルのリムーバブルメディアの設定 (手順)，52
df コマンド，86, 336
DOS, ファイルシステム，329
driver not attached メッセージ，77
DVD-RAM, ファイルシステムの作成 (手順)，40
DVD-ROM，330

E

EFI ラベル
  VTOC ラベルとの比較，199
  (概要)，198
  システムのインストール，200
  制限，199
  問題のトラブルシューティング，201
eject コマンド, リムーバブルメディア (手順)，51
/etc/dfs/dfstab ファイル, 共有されたリムーバブルメディアの設定 (手順)，52
/etc/dumpdates ファイル，518-519
/export/home ディレクトリ，338

F

FDFS ファイルシステム，334
ff コマンド，336
FIFOFS ファイルシステム，334
FIFO の i ノード，434
format.dat ファイル
  エントリの作成 (概要)，231
  エントリの作成 (手順)，231
  キーワード，311, 314
  構文規則，311
format ユーティリティー
  analyze メニュー，308
format ユーティリティー (続き)
  defect メニュー，309-310
  fdisk メニュー，307
  partition メニュー，306, 307
  SCSI ディスクドライブの自動構成 (概要)，232
  SCSI ディスクドライブの自動構成 (手順)，234
  Solaris fdisk パーティションの作成 (手順)，258
  (概要)，207
  機能と利点，207
  コマンド名の入力 (手順)，316
  システム上のディスクの確認 (手順)，216
  システム上のディスクの確認 (例)，218
  使用上のガイドライン，208-209
  使用する場合，207
  ディスクがフォーマット済みかどうかの確認 (手順)，219
  ディスクスライス情報の表示 (例)，223
  ディスクスライスとディスクラベルの作成 (手順)
    SPARC，243
    x86，263
  ディスクのフォーマット (例)，221
  ディスクラベルの作成
    例，225
  入力，315, 317
  破損したディスクラベルの復元 (手順)，228
  ブロック番号の指定 (手順)，316
  ヘルプ機能の使用，317
  メインメニュー，304
free hog スライス，「提供側のスライス」を参照
fsck コマンド，86, 336
  FSACTIVE 状態フラグ，430
  FSBAD 状態フラグ，430
  FSCLEAN 状態フラグ，430
  FSSTABLE 状態フラグ，430
  検査
    i ノードリストのサイズ，433
    空き i ノード数，433
    空きブロック数，433
    スーパーブロック，432
  構文とオプション，450
  修復，443
  修復する条件，431
  状態フラグ，430

fsck コマンド *(続き)*
対話式で使用，438
fsdb コマンド，336
fssnap コマンド, UFS スナップショットの作成 (手順)，494
fsstat コマンド, 説明，320
fsstat コマンド (例)，320
fstypes ファイル，352
fstyp コマンド，336
fuser コマンド
リムーバブルメディアが使用中かどうかの調査 (手順)，50
リムーバブルメディアにアクセスしているプロセスの強制終了 (手順)，50

G

grep コマンド，352
GRUB
GRUB によるディスク管理
x86，195
Solaris フェイルセーフブート
x86，195

H

High Sierra ファイルシステム，329
/home (自動マウント)，352
HSFS,「High Sierra ファイルシステム」を参照

I

x86 システム, UFS フォーマット，47
InfiniBand デバイス
HCA に接続されている IB デバイスの構成解除 (手順)，184
HCA のカーネル IB クライアントの表示 (手順)，184
IB Port、HCA_SVC、または VPPA デバイスの構成解除 (手順)，182
IB Port、HCA_SVC、または VPPA デバイスの構成 (手順)，182
InfiniBand デバイス *(続き)*
IB 擬似デバイスの構成解除 (手順)，183
IB 擬似デバイスの構成 (手順)，183
IB 通信サービスの表示 (手順)，185
IOC デバイスの構成解除 (手順)，181
IOC デバイスの構成 (手順)，181
IOC の構成の更新 (手順)，187
IP p_key テーブルの更新 (手順)，185
VPPA 通信サービスの追加 (手順)，186
概要，175
既存の IB Port、HCA_SVC、VPPA 通信サービスの削除 (手順)，186
動的構成 (概要)，178
表示 (手順)，179
installboot コマンド，249, 265
iSCSI
iSCSI 構成に関する問題の障害追跡 (手順)，294
iSCSI 構成の監視 (手順)，284
iSCSI ターゲット発見の構成 (手順)，279
iSCSI ディスクへのアクセス (手順)，282
iSCSI のイニシエータとターゲットのパラメータの変更 (手順)，287
Solaris iSCSI 構成の準備 (手順)，273
一般的な iSCSI エラーメッセージ，296
(概要)，268
静的および動的ターゲット発見，272
ソフトウェア要件およびハードウェア要件，269
単方向または双方向 CHAP 認証の構成 (手順)，276
発見された iSCSI ターゲットの削除 (手順)，280
iscsiadm list, ISCSI 構成情報の表示 (例)，285
iscsiadm add コマンド, 静的または動的ターゲットの追加 (例)，280
iscsiadm modify コマンド
CHAP を有効にする (例)，277
静的または動的ターゲットの有効化または無効化 (例)，281
iscsiadm remove コマンド, 静的または動的ターゲットの削除 (例)，281
ISO 9660 ファイルシステム，329
ISO 標準, 9660 CD フォーマット，47
i ノード，454-455
FIFO，434

i ノード (続き)
キャラクタ型特殊ファイル，434
形式とタイプの検査，434
ごとのバイト数，459-460
サイズ，435
シンボリックリンク，434
通常，434
ディレクトリ，434
不正な i ノード番号，437
ブロック型特殊ファイル，434
リンク数，434
i ノードの形式，434
i ノードの状態，434
i ノードのタイプ，434
i ノードのリンク数，434
i ノードリストのサイズ，433

K

/kernel/drv ディレクトリ，76

L

labelit コマンド，336
lost+found ディレクトリ，430

M

「media was found」メッセージ，36
mkfile コマンド，424, 425
mkfs コマンド，336, 357
mkisofs コマンド, データ CD または DVD の
ファイルシステムを作成する (手順)，61
MNTFS ファイルシステム，339
mnttab ファイル，348
mountall コマンド，336
mount コマンド，86
mt コマンド，553

N

NAMEFS ファイルシステム，334
ncheck コマンド，336
newfs コマンド，86, 357
NFS
vfstab エントリ，375
サーバーの説明，351
説明，350
nfsd デーモン
起動，52
実行中かどうかの確認，52
「no media was found」メッセージ，36

O

OBJFS ファイルシステム，334
/opt ディレクトリ，338

P

passwd ファイル, テープから復元 (例)，508
PCFS ファイルシステム，329
PCI Express (PCIe) のサポート, 説明，70
PCI デバイス
PCI アダプタカードの取り付け (手順)，109
PCI アダプタカードの取り外し (手順)，108
PCI 構成に関する問題のトラブルシューティン
グ，111
PCI スロット構成情報の表示 (手順)，106
PROCFS ファイルシステム, (概要)，333
/proc ディレクトリ，333, 339
prtvtoc コマンド，86
例，227

R

raw ディスクデバイスインタフェース，86
RCM スクリプト
概要，112
コマンド，112
Rock Ridge 拡張 (HSFS ファイルシステム)，329

S

SCSI ディスクドライブ，232
SCSI テープドライブ，551
SCSI デバイス
  cfgadm コマンドによる切り離し (手順)，98
  SCSI 構成に関する問題の障害追跡，103
  SCSI コントローラ上の同一 SCSI デバイスの交換 (手順)，101
  SCSI コントローラの構成解除 (手順)，96
  SCSI コントローラの構成 (手順)，97
  SCSI コントローラの接続 (手順)，99
  SCSI デバイスの SCSI バスへの取り付け (手順)，100
  SCSI デバイスの構成 (手順)，97
  SCSI デバイスの取り外し (手順)，102
  失敗した SCSI 構成解除操作の解決 (手順)，104
  に関する情報の表示 (手順)，96
shareall コマンド，351
SHAREFS ファイルシステム，334
share コマンド，351
  リムーバブルメディアをほかのシステムで使用可能にする (手順)，52
Solaris fdisk パーティション, ガイドライン，257-258
Solaris フェイルセーフブート
  x86，195
  (手順)
    x86，253
space、最適化のタイプ，459
SPARC システム, UFS フォーマット，47
SPECFS ファイルシステム，334
svcadm disable, (例)，27
swapadd コマンド，420
SWAPFS ファイルシステム，334
swap コマンド，424
sysdef コマンド，77

T

tar コマンド
  dd コマンドによるファイルのリモートテープへのコピー (手順)，542
  dd コマンドによるリモートテープからのファイルの取り出し (手順)，543
tar コマンド (続き)
  (概要)，534
  テープからのファイルの取り出し (手順)，536
  テープ上のファイルのリスト (手順)，536
  フロッピーディスクからのファイルの取り出し (手順)，546
  フロッピーディスクのファイルのリストの表示 (方法)，546
time (最適化のタイプ)，459
TMPFS ファイルシステム, 概要，332
/tmp ディレクトリ，332, 339

U

uDAPL
  DAT 静的レジストリ内のサービスプロバイダの登録解除 (手順)，190
  DAT 静的レジストリの更新 (手順)，189
  DAT 静的レジストリへのサービスプロバイダの登録 (手順)，190
  (概要)，188
  有効にする (手順)，188
UDF ファイルシステム，330
UFS CD, SPARC と x86 のフォーマットの比較，47
ufsdump コマンド
  オプションと引数，522
  完全バックアップの例，484, 486
  機能，517-522
  制限，521
  増分バックアップの例，484
  データのコピー方法，518
  メディアの終わりの検出，518
ufsdump コマンド (概要)，482
ufsrestore コマンド，523
  使用するための準備 (概要)，502
  使用するテープの決定 (手順)，504
UFS スナップショット
  完全バックアップの作成 (手順)，497
  作成 (手順)，494
  説明，491
UFS ファイルシステム，329, 339
  /，338
  /etc/vfstab によるマウント，376
  拡張基礎タイプ，340

UFS ファイルシステム (続き)
状態フラグ，340
大規模ファイルシステム，340
大規模ファイルを持たない UFS ファイルシステムのマウント (手順)，378
マウント，375
マルチテラバイトファイルシステム，340
ロギング，340
UFS ロギング、概要，344
umountall コマンド，336
umount コマンド，336
UNIX ファイルシステム，329
USB デバイス
2.0 デバイスの概要，133
CDC ACM デバイスのサポート，122
EHCI、OHCI、および UHCI のサポート，131
Prolific および Keyspan シリアルアダプタのサポート，130
Solaris USB アーキテクチャー (USBA)，132
USB 2.0 のサポート，133
USB カメラの追加 (手順)，146
USB デバイスサブツリーの論理的な接続解除 (手順)，172
USB デバイス情報の表示 (手順)，149
USB デバイスの構成 (手順)，170
USB デバイスの接続 (手順)，171
USB デバイスのリセット (手順)，172
USB デバイスの論理的な接続解除 (手順)，171
vold を使用しないで大容量ストレージデバイスを追加する (手順)，146
vold を使用しないで大容量ストレージデバイスを取り外す (手順)，148
vold を使用しないで大容量ストレージデバイスをフォーマットする (手順)，152
vold を使用しないで大容量ストレージデバイスをマウント解除する (手順)，159
vold を使用しないで大容量ストレージデバイスをマウントする (手順)，159
ZFS ファイルシステムのサポート，123
オーディオ
(概要)，163
主オーディオデバイスの変更 (手順)，166
デバイスの所有権，167
オーディオデバイスの追加 (手順)，165
USB デバイス (続き)
オーディオデバイスの問題のトラブルシューティング，166
(概要)，126
仮想キーボードおよびマウスのサポート，124
キーボードとマウスデバイス，135
ケーブル，138
合成デバイス，128
主オーディオデバイスの識別 (手順)，165
大容量ストレージデバイス上のファイルシステムの作成 (手順)，150
大容量ストレージデバイスのトラブルシューティングヒント，161
デバイスクラス，129
デバイスノード，132
デバイスの構成解除 (手順)，170
電源管理，137
電源割り当てのサポート，123
ドライバ，129
名前，128
バス情報の表示 (手順)，169
バス電源供給方式のデバイス，135
バスの説明，127
非準拠大容量ストレージデバイスの使用 (概要)，144
複合デバイス，128
物理デバイス階層，127
フロッピーディスクデバイス (概要)，143
ホイール付きマウスのサポート，136
ホストコントローラとルートハブ，136
ホットプラグ (概要)，145
ホットプラグ対応属性，122
リムーバブル大容量ストレージ (概要)，142
略語，126
/usr ファイルシステム，338

V

/var ディレクトリ，338
vfstab ファイル，352, 420
LOFS のエントリ，366
エントリの追加 (手順)，374
すべてのファイルをマウント，375
スワップの追加，420

vfstab ファイル (続き)
デフォルト，349
volcopy コマンド，336

あ
アーカイブ, cpio コマンドを使ってファイルを複数のフロッピーディスクにアーカイブ (手順)，547
空き i ノード数，433
空きブロック，456
空きブロック数，433
空き容量 (最小)，458-459
アクセス
iSCSI ディスク (手順)，282
ディスクデバイス，85
テープデバイス，88
リムーバブルメディア (手順)，49

い
一時ファイルシステム (TMPFS), 概要，332

え
エラーメッセージ, iSCSI，296

お
オプション, ufsdump コマンド，522

か
解決, 失敗した SCSI 構成解除操作 (手順)，104
書き込み, データ CD、データ DVD、およびオーディオ CD (概要)，58
拡張基礎タイプ (UFS ファイルシステム)，340
確認
nfsd デーモンが実行中，52
システム上のディスク (手順)，217
確認 (続き)
ファイルシステム名，503
仮想ファイルシステムテーブル，349
仮想メモリー記憶域、定義，416
監視, iSCSI 構成 (手順)，284
間接ブロック，436
完全バックアップ
定義，466
例，484, 486
管理
GRUB によるディスク管理
x86，195

き
記憶域 (仮想メモリー)、定義，416
記憶容量 (メディア)，467, 550
擬似ファイルシステム, (概要)，331
起動, nfsd デーモン，52
キャラクタ型特殊ファイルの i ノード，434
強制終了
ファイルシステムを使用中のすべてのプロセス (手順)，382
リムーバブルメディアにアクセスしているプロセス (手順)，50
共有, ファイル，350-351
切り離し, SCSI コントローラ (手順)，98
記録
増分バックアップ，519
ダンプ，518-519

け
決定
テープデバイスのタイプ，503
テープデバイス名，503
ファイルシステムのタイプ，352
検査
CacheFS ファイルシステム (例)，399
i ノードの形式とタイプ，434
i ノードリストの整合性，433
空き i ノード数，433
空きブロック数，433

検査 (続き)
ファイルシステムサイズ，433
ファイルシステムの修復，438
リムーバブルメディア上のファイルシステム (手順)，41
検出、メディアの終わり, ufsdump コマンド，518

こ

交換, SCSI コントローラ上の同一 SCSI デバイス (手順)，101
更新
DAT 静的レジストリ (手順)，189
IOC の構成 (手順)，187
IP p_key テーブル (手順)，185
構成
IB Port、HCA_SVC、または VPPA デバイス (手順)，182
IB 擬似デバイス (手順)，183
IOC デバイス (手順)，181
iSCSI ターゲット発見 (手順)，279
iSCSI の単方向または双方向 CHAP 認証 (手順)，276
SCSI コントローラ (手順)，97
SCSI デバイス (手順)，97
USB デバイス (手順)，170
構成解除
HCA に接続されている IB デバイス (手順)，184
IB Port、HCA_SVC、VPPA デバイス (手順)，182
IB 擬似デバイス (手順)，183
IOC デバイス (手順)，181
SCSI コントローラ (手順)，96
USB デバイス (手順)，170
構造、シリンダグループの，453-456
構文, fsck コマンド，450
コピー
cpio コマンドを使って個々のファイルを (概要)，532
cpio コマンドを使ってファイルシステム間の (概要)，532
cpio コマンドを使ってファイルのグループを (概要)，532
ファイルシステム全体 (dd)，528
コピー (続き)
ファイルをフロッピーディスクにコピーする (手順)，544

さ

再構成ブート，233
x86 の例，256
SPARC の例，242
最小空き容量，458-459
サイズ
i ノード，435
ファイルシステムの検査，433
フラグメント，458
削除
CacheFS ファイルシステム (手順)，397
UFS スナップショット情報 (例)，496
既存の IB Port、HCA_SVC、VPPA 通信サービス (手順)，186
発見された iSCSI ターゲット (手順)，280
不要なスワップファイル，428
作成
DVD-RAM 上のファイルシステム (手順)，40
UFS スナップショット 例，494
UFS スナップショット情報の完全バックアップ (手順)，497
UFS スナップショット (手順)，494
USB 大容量ストレージデバイス上のファイルシステム (手順)，150
スワップファイル，424
データ CD または DVD のファイルシステム (手順)，61
パッキングリスト (手順)，403
ファイルシステム (概要)，357
リムーバブルメディア上のファイルシステム (手順)，38
ループバックファイルシステム (概要)，365
サポートされていないデバイス，76

し
識別
　主 USB オーディオデバイス (手順)，165
　デバイス，77
システムディスク
　接続 (手順)
　　x86，253
　説明，205
自動構成プロセス，76
自動マウント, /home，352
収集, CacheFS の統計情報 (概要)，409
修復, リムーバブルメディア上の不良ブロック (手順)，41
重複ブロック，435
終了
　ファイルシステムに対するすべてのプロセス (手順)，382
　リムーバブルメディアにアクセスしているプロセスの強制終了 (手順)，50
準備
　Solaris iSCSI 構成 (手順)，273
　バックアップ (概要)，480-481
　ファイルの復元 (概要)，502-503
障害追跡
　cachefspack エラー，405
　iSCSI 構成に関する問題 (手順)，294
　SCSI 構成に関する問題，103
　失敗した SCSI 構成解除操作，104
状態フラグ
　fsck，430
　UFS ファイルシステム，340
シリンダグループ，453-456
シンボリックリンク，434

す
スーパーブロック，432, 445, 454
スライス (定義)，202
スワップパーティション、定義，416
スワップファイル
　vfstab へ追加，420
　作成，424
　表示，422-424
　不要なスワップファイルの削除，428

せ
制限, リムーバブルメディアへのアクセス (手順)，59
接続
　SCSI コントローラ (手順)，99
　USB デバイスの論理的な接続 (手順)，171
接続解除
　USB デバイスサブツリーの論理的な接続解除 (手順)，172
　USB デバイスの論理的な接続解除 (手順)，171
設定, CacheFS ロギング，410

そ
増分バックアップ，466, 519
　例，484

た
大規模ファイルオプション，370
ダンプレベル
　定義，471
　日単位増分バックアップ，472

ち
調査
　ファイルシステムのタイプ，352
　マウント済みのファイルシステム，373
直接入出力，346

つ
追加
　>新しいリムーバブルメディアドライブの追加 (手順)，48
　USB オーディオデバイス (手順)，165
　USB カメラ (手順)，146
　vold を使用しないで USB 大容量ストレージデバイスを追加する (手順)，146

追加 (続き)
vold を使用しないで USB 大容量ストレージデバイスを取り外す (手順), 148
vold を使用しないで USB 大容量ストレージデバイスをフォーマットする (手順), 152
VPPA 通信サービス (手順), 186
エントリ /etc/vfstab ファイル (手順), 374
スワップを vfstab へ, 420
ディスク (概要)
SPARC, 240
x86, 253-266
通常のノード, 434

て

停止, CacheFS ロギング, 412
ディスク
SCSI ドライブの自動構成, 232
欠陥セクターの修復, 234, 236
追加 (概要)
SPARC, 240
x86, 253-266
ディスクスライスとディスクラベルの作成 (手順)
SPARC, 243
ディスクスライスとディスクラベルの作成 (例)
SPARC, 245
二次ディスクの接続 (例)
SPARC, 246
破損したディスクラベルの復元 (概要), 228
破損したディスクラベルの復元 (手順), 228
フォーマット (概要), 209
フォーマット済みかを調べる (手順), 220
フォーマットする場合 (概要), 219
ディスクコントローラ, 86
ディスクスライス
システム構成の要件, 206
使用するスライスの決定, 206
情報の表示 (概要), 222-223
定義, 202
ディスクスライスの指定, 86
ディスクのフォーマット (概要), 209
ディスクベースのファイルシステム, 328
ディスクラベル
作成 (概要), 224
説明, 197
ディレクトリ
cpio コマンドを使ってファイルシステム間のコピー (概要), 532
i ノード, 434
/proc, 333
/tmp, 332
未割り当てブロック, 436
ディレクトリデータブロック, 436
データブロック, 437, 456
テープ
tar コマンドによるファイルの取り出し (手順), 536
記憶容量, 467, 550
サイズ, 467, 550
性質, 521
容量, 521
テープデバイス, 復元するタイプの決定, 503
テープデバイス (命名), 88
テープドライブ
管理, 554
最大の SCSI テープドライブ台数, 551
巻き戻し, 551-552
テープドライブの管理, 554
デバイス, アクセス, 83
デバイスドライバ
追加, 83
定義, 75
デバイスの電力管理, 説明, 73
デバイス名
テープデバイス名の決定, 503
バックアップ, 550-552
ファイルシステム名の確認, 503
デフォルト
SunOS ファイルシステム, 338
/tmp 用のファイルシステム (TMPFS), 332
デフォルトの SunOS ファイルシステム, 338

と

動的構成, InfiniBand デバイス, 178
動的再構成 (概要), 90

登録, DAT 静的レジストリへのサービスプロバイダの登録 (手順), 190
登録解除, DAT 静的レジストリ内のサービスプロバイダの登録解除 (手順), 190
トラブルシューティング
EFI ディスクラベル, 201
PCI 構成に関する問題, 111
USB オーディオデバイスの問題, 166
USB 大容量ストレージデバイス, 161
取り出し
tar コマンドによるテープからのファイルの (手順), 536
リムーバブルメディア (手順), 51
取り付け
PCI アダプタカード (手順), 109
SCSI デバイスの SCSI バスへの取り付け (手順), 100
取り外し
PCI アダプタカード (手順), 108
SCSI デバイス (手順), 102

に

二次ディスク
システムへの接続 (手順)
SPARC, 243
x86, 256
説明, 205
入出力、直接, 346

ね

ネットワークベースのファイルシステム, 330

は

パーティション (スワップ)、定義, 416
バイト (i ノードごとの数), 459-460
場所の調査, CacheFS ログファイル, 411
バス指向コントローラ, 87
バス指向ディスクコントローラ, 87
バックアップ
完全バックアップおよび増分バックアップの定義, 466
準備 (概要), 480-481
増分バックアップの記録, 519
タイプ, 466
デバイス名, 550-552
ファイルシステムの選択, 463
ファイルシステムのバックアップと復元
コマンド, 462
定義, 462
理由, 463
バックアップスケジュール
ガイドライン, 468, 469
サーバー用, 476-478
ダンプレベルの使用, 471
日単位増分、週単位累積バックアップ, 475
日単位累積、週単位増分バックアップ, 474
日単位累積、週単位累積バックアップ, 472
例, 472, 478
バックアップのスケジュール, 468
パラメータ (ファイルシステム), 457-460

ひ

日単位個別バックアップ, 472
表示
CacheFS 統計情報, 413
HCA のカーネル IB クライアント (手順), 184
IB 通信サービス (手順), 185
InfiniBand デバイス情報 (手順), 179
PCI スロット構成情報 (手順), 106
SCSI デバイスに関する情報, 96
USB デバイス情報 (手順), 149
USB バス情報 (手順), 169
システム構成情報, 77, 80
スワップ空間, 422-424
ディスクスライス情報 (概要), 222
パックされたファイル (手順), 401
パックされたファイル (例), 401
リムーバブルメディアのユーザー (手順), 50

ふ

ファイル
- cpio コマンドを使って複数のフロッピーディスクにアーカイブ (手順), 547
- /etc/default/fs, 352
- /etc/dfs/fstypes, 352
- /proc ディレクトリ内, 333
- tar コマンドによるテープからの取り出し (手順), 536
- 共有, 350-351
- メディアにコピーするためのコマンド (概要), 526

ファイルシステム
- 4.3 Tahoe, 329
- BSD Fat Fast, 329
- CTFS, 334
- DOS, 329
- /export/home, 338
- FDFS, 334
- FIFOFS, 334
- High Sierra, 329
- ISO 9660, 329
- MNTFS, 339
- NAMEFS, 334
- OBJFS, 334
- /opt, 338
- PCFS, 329
- /proc, 339
- PROCFS、(概要), 333
- SHAREFS, 334
- SPECFS, 334
- SWAPFS, 334
- TMPFS, 332
- UFS, 329
- UNIX, 329
- /usr, 338
- /var, 338
- カスタムパラメータ, 457-460
- 管理コマンドの説明, 336
- 擬似、(概要), 331
- キャッシュ (概要), 387
- 共有, 350-351
- 検査と修復, 438
- サイズの検査, 433

ファイルシステム (続き)
- 作成 (概要)
  - ループバック (LOFS), 365
- 修正, 444
- 修復, 443, 444
- 使用中のすべてのプロセスを終了(手順), 382
- シリンダグループの構造, 453-456
- 全体のコピー (dd), 528
- 損傷, 430
- 大規模, 370
- タイプ, 328
- タイプの調査, 352
- ディスクベース, 328
- デフォルトの SunOS ファイルシステム, 338
- ネットワークベース, 330
- バックアップする理由, 463
- バックアップ対象, 463
- 不整合の原因, 432
- プロセス、(概要), 333
- マウントテーブル, 348
- マニュアルページ, 337
- 利用可能にする (概要), 367-373

ファイルシステムテーブル, 仮想, 349
ファイルシステムの修復, 443, 444
ファイルシステムの損傷, 430
ファイルシステムの損傷の原因, 430
ファイルシステムのタイプ, 328
ファイルシステムの復元
- 完全 (手段), 510
- 準備 (概要), 502-503
- 使用するテープの決定 (手順), 504
- 全体 (例), 511
- テープドライブのタイプ, 503
- ルートと /usr (SPARC) (例), 515
- ルートと /usr (x86) (例), 515

ファイルシステムの不整合, 432
ファイルシステム名, 503
ファイルシステム用のカスタムパラメータ, 457-460
ファイルの復元
- 対話式で復元 (例), 507
- 非対話式復元 (例), 508

ブートブロック, 454

ブートブロックのインストール (手順),
SPARC, 249
フォーマット, rmformat によるフロッピーディスクのフォーマット (手順), 37
不整合なファイルシステムの修正, 444
不正な「.」と「..」エントリ, 437
不正な i ノード番号, 437
不正なスーパーブロック, 445
不正なスーパーブロックの復元, 445
不正なブロック番号, 435
物理デバイス名
定義, 84
フラグメントサイズ, 458
プロセスファイルシステム (PROCFS), 333
ブロック
空き, 456
間接, 436
重複, 435
通常データ, 437
ディレクトリデータ, 436
データ, 456
特殊ファイル i ノード, 434
ブート, 454
不正, 435
論理サイズ, 457
ブロックディスクデバイスインタフェース
いつ使用するか, 86
定義, 86
フロッピーディスク
cpio コマンドを使ってファイルを複数のフロッピーディスクにアーカイブ (手順), 547
rmformat によるフォーマット (手順), 37
ファイルシステムの作成 (手順), 38
ボリューム管理による読み込み (手順), 36

へ
変更
iSCSI のイニシエータとターゲットのパラメータ (手順), 287
主 USB オーディオデバイス (手順), 166

ほ
ホットプラグ
cfgadm コマンドによる SCSI コントローラの切り離し (手順), 98
PCI アダプタカードの取り付け (手順), 109
PCI アダプタカードの取り外し (手順), 108
PCI デバイス (概要), 105
SCSI コントローラ上の同一 SCSI デバイスの交換 (手順), 101
SCSI コントローラの構成 (手順), 97
SCSI コントローラの接続 (手順), 99
SCSI デバイスの SCSI バスへの取り付け (手順), 100
SCSI デバイスの構成解除 (手順), 96
SCSI デバイスの構成 (手順), 97
SCSI デバイスの取り外し (手順), 102
USB デバイスサブツリーの論理的な接続解除 (手順), 172
USB デバイスの構成解除 (手順), 170
USB デバイスの構成 (手順), 170
USB デバイスの論理的な接続解除 (手順), 171
USB デバイスの論理的な接続 (手順), 171
概要, 90

ま
マウント
/etc/vfstab によるファイルシステムの, 376
NFS ファイルシステム, 375
UFS ファイルシステム, 375
UFS ファイルシステム (手順)
大規模ファイルを持たない, 378
vfstab ファイル内のすべてのファイル, 375
vold を使用しないで USB 大容量ストレージデバイスをマウントする (手順), 159
ファイルシステムの自動マウント, 351-352
リムーバブルメディア
自動マウントと手動マウントの比較, 30, 31
リモートのリムーバブルメディアを手動で (例), 54
リモートリムーバブルメディアの手動マウント (例), 54
ループバックファイルシステム (LOFS), 375

マウント解除, vold を使用しないで USB 大容量ストレージデバイスをマウント解除する (手順), 159
マウントテーブル, 348
マウントポイント定義, 346
マニュアルページ、ファイルシステム用, 337

み
未割り当てディレクトリブロック, 436
未割り当ての i ノード, 434

む
無効にする
  リムーバブルメディアサービス, 27
  リムーバブルメディアサービス (手順), 48
  リムーバブルメディアの書き込み保護 (手順), 42

め
メディアの終わりの検出
  cpio コマンド, 533
  ufsdump コマンド, 518
メモリー記憶域 (仮想)、定義, 416

ゆ
有効にする
  uDAPL, 188
  リムーバブルメディアサービス (手順), 48
  リムーバブルメディアの書き込み保護 (手順), 42

よ
読み込み, フロッピーディスク (手順), 36

り
リセット, USB デバイス (手順), 172
リムーバブルメディア
  DVD-RAM 上のファイルシステムの作成 (手順), 40
  rmformat によるフロッピーディスクのフォーマット (手順), 37
  アクセスしているプロセスの強制終了 (手順), 50
  アクセス (手順), 49
  アクセス (例), 49
  新しいリムーバブルメディアドライブの追加 (手順), 48
  書き込み保護を有効にする (手順), 42
  管理、利点, 29
  サービス、無効にする, 27
  取り出し (手順), 51
  名前, 46
  ファイルシステムの検査 (手順), 41
  ファイルシステムの作成 (手順), 38
  フロッピーディスクの読み込み (手順), 36
  ほかのシステム上のメディアへのアクセス (例), 54
  マウント
    手動と自動の比較, 30, 31
  メディアが使用中かどうかの調査 (手順), 50
  リムーバブルメディアサービスを無効にするまたは有効にする (手順), 48
  リムーバブルメディア上の不良ブロックの修復 (手順), 41
  リムーバブルメディアのマウント (例), 54
  リモートのメディアのマウント (例), 54

る
ルート (/) ファイルシステムまたはディレクトリ, 338
ループバックファイルシステム (LOFS)
  作成 (概要), 365
  マウント, 375

れ

レベル 0 バックアップ，471

ろ

ログ (ダンプの記録)，518-519
論理デバイス名
　定義，85
　ディスク，85
　テープ，88
　リムーバブルメディア，88
論理ブロックサイズ，457

わ

割り当て済みの i ノード，434