

Tru64 UNIX

システム管理ガイド

Part Number: AA-RK3PE-TE

2002 年 11 月

ソフトウェア・バージョン: Tru64 UNIX Version 5.1B 以上

本書は、Tru64 UNIX オペレーティング・システムのシステム管理について説明しています。本書で説明している管理作業は、UNIX コマンド、シェル・スクリプト、SysMan Menu または SysMan Station のユーザ・インタフェースを使用して行うことができます。

© 2002 日本ヒューレット・パッカート株式会社

本書の著作権は日本ヒューレット・パッカート株式会社が保有しており、本書中の解説および図、表は日本ヒューレット・パッカートの文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

日本ヒューレット・パッカートは、弊社または弊社の指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、弊社は一切その責任を負いかねます。

本書で解説するソフトウェア(対象ソフトウェア)は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

COMPAQ, Compaq ロゴ, Digital ロゴは U.S. Patent and Trademark Office に登録されています。Alpha, AlphaServer, NonStop, TruCluster, および Tru64 は米国 Compaq Computer Corporation の商標です。

Microsoft, Windows および Windows NT は米国 Microsoft 社の登録商標です。Intel は米国 Intel 社の登録商標です。Motif, OSF/1, UNIX, The Open Group および X/Open は、The Open Group の米国ならびに他の国における商標です。

このドキュメントに記載されているその他の会社名および製品名は、各社の商標または登録商標です。

原典: System Administration (AA-RH9FE-TE)
©2002 Hewlett-Packard Company

目次

まえがき

1 システム管理メソッドおよびユーティリティ

| | | |
|----------|--|------|
| 1.1 | SysMan Menu およびその他のユーティリティの概要 | 1-2 |
| 1.2 | 関連ドキュメント | 1-4 |
| 1.2.1 | リファレンス・ページ | 1-4 |
| 1.2.2 | オンライン・ヘルプ | 1-4 |
| 1.2.3 | Web ベースのヘルプ | 1-5 |
| 1.3 | システムのセットアップ | 1-7 |
| 1.4 | 管理メソッド | 1-11 |
| 1.5 | CDE での管理ユーティリティ | 1-14 |
| 1.5.1 | CDE での SysMan へのアクセス | 1-15 |
| 1.5.2 | システム・セットアップ | 1-18 |
| 1.5.2.1 | クイック・セットアップ | 1-18 |
| 1.5.2.2 | カスタム・セットアップ | 1-19 |
| 1.6 | SysMan Menu | 1-23 |
| 1.7 | SysMan コマンド行の使用 | 1-26 |
| 1.8 | SysMan Station | 1-30 |
| 1.8.1 | SysMan Station の「Status」オプションの使用 | 1-32 |
| 1.8.2 | SysMan Station の「views」の使用 | 1-33 |
| 1.8.3 | SysMan Station のメニュー・オプションの使用 | 1-37 |
| 1.9 | HP Insight Manager | 1-38 |
| 1.10 | パーソナル・コンピュータでの SysMan の使用 | 1-40 |
| 1.11 | シリアル回線コンソールのセットアップ | 1-42 |
| 1.11.1 | コンソール・ポートの設定 | 1-43 |
| 1.11.1.1 | モデムの COMM1 への接続 | 1-43 |

| | | |
|------------|---|------|
| 1.11.1.2 | 構成可能な DCD タイマ値の設定 | 1-43 |
| 1.11.1.3 | コンソール環境変数の設定 | 1-44 |
| 1.11.1.4 | モデム設定の確認 | 1-45 |
| 1.11.2 | コンソール・ポート接続の開始 | 1-45 |
| 1.11.2.1 | コンソール・ポートの使用 | 1-45 |
| 1.11.2.1.1 | コンソール・ログ・メッセージのオフ | 1-46 |
| 1.11.2.1.2 | リモート・システムのシャットダウン | 1-46 |
| 1.11.2.1.3 | リモート・セッションの終了 | 1-47 |
| 1.11.3 | トラブルシューティング | 1-47 |
| 2 | システムの起動とシャットダウン | |
| 2.1 | シャットダウンとブート操作の概要 | 2-1 |
| 2.1.1 | シャットダウンの方法 | 2-3 |
| 2.1.2 | ブート方式 | 2-3 |
| 2.1.3 | 関連ドキュメント | 2-4 |
| 2.1.3.1 | マニュアル | 2-4 |
| 2.1.3.2 | リファレンス・ページ | 2-5 |
| 2.1.3.3 | オンライン・ヘルプ | 2-6 |
| 2.1.4 | システム・ファイル | 2-7 |
| 2.1.5 | 関連ユーティリティ | 2-7 |
| 2.2 | ブート操作の概要 | 2-8 |
| 2.2.1 | 自動ブートと手動ブート | 2-8 |
| 2.2.2 | シングルユーザ・モードまたはマルチユーザ・モードへの ブート | 2-9 |
| 2.3 | インストール済みシステムのブート | 2-10 |
| 2.3.1 | 電源が切られているシステムのブート | 2-11 |
| 2.3.2 | 電源は入っているが停止しているシステムのブート | 2-12 |
| 2.3.3 | シングルユーザ・モードからのブート | 2-12 |
| 2.3.4 | クラッシュしたシステムのブート | 2-13 |

| | | |
|---------|---------------------------------|------|
| 2.3.5 | ネットワークから切り離されたシステムのブート準備 | 2-14 |
| 2.4 | システムのブート | 2-16 |
| 2.4.1 | コンソール環境変数の設定とブート・コマンドの使用 | 2-17 |
| 2.4.2 | ブート・コマンドの変更 | 2-20 |
| 2.4.3 | 対話式ブートを使うルート・ファイル・システムの検証 . | 2-21 |
| 2.5 | システム実行レベルの識別 | 2-23 |
| 2.6 | システム実行レベルの変更 | 2-23 |
| 2.6.1 | シングルユーザ・モードでの実行レベルの変更 | 2-24 |
| 2.6.2 | マルチユーザ・モードからの実行レベルの変更 | 2-24 |
| 2.6.2.1 | 別のマルチユーザ実行レベルへの変更 | 2-25 |
| 2.6.2.2 | シングルユーザ・モードへの変更 | 2-25 |
| 2.6.2.3 | inittab ファイルの再読み取り | 2-26 |
| 2.7 | シンメトリック・マルチプロセッシング | 2-26 |
| 2.7.1 | 既存システムへの CPU の追加 | 2-27 |
| 2.7.2 | マルチプロセッサ・システムでの自動リブート | 2-27 |
| 2.8 | システム・クロックの設定および再設定 | 2-27 |
| 2.9 | ブート時の問題のトラブルシューティング | 2-28 |
| 2.9.1 | ハードウェア障害 | 2-28 |
| 2.9.2 | ソフトウェア障害 | 2-28 |
| 2.10 | システムのシャットダウン | 2-29 |
| 2.11 | マルチユーザ・モードにおけるシステムの停止 | 2-30 |
| 2.11.1 | SysMan シャットダウンの使用 | 2-31 |
| 2.11.2 | システムのシャットダウンとユーザへの警告 | 2-34 |
| 2.11.3 | システムのシャットダウンと停止 | 2-36 |
| 2.11.4 | システムのシャットダウンと自動リブート | 2-37 |
| 2.11.5 | 即時シャットダウンとシステムの停止 | 2-38 |
| 2.12 | シングルユーザ・モードでのシステムの停止 | 2-38 |
| 2.12.1 | reboot コマンドでのシステムの停止とリブート | 2-39 |
| 2.12.2 | fasthalt コマンドによるシステムの停止 | 2-39 |

| | | |
|--------|-------------------------------|------|
| 2.12.3 | fastboot コマンドによるシステムの停止 | 2-40 |
|--------|-------------------------------|------|

3 システム環境のカスタマイズ

| | | |
|---------|--|------|
| 3.1 | システム初期化ファイルの確認と変更 | 3-2 |
| 3.1.1 | /etc/inittab ファイル | 3-5 |
| 3.1.1.1 | 初期化時の省略時実行レベルの指定 | 3-8 |
| 3.1.1.2 | 待ち実行レベルの指定 | 3-8 |
| 3.1.1.3 | コンソール実行レベルの指定 | 3-9 |
| 3.1.1.4 | 端末と端末実行レベルの指定 | 3-9 |
| 3.1.1.5 | プロセス実行レベルの指定 | 3-10 |
| 3.1.1.6 | 端末回線の保護 | 3-11 |
| 3.1.2 | init および rc ディレクトリの構造 | 3-11 |
| 3.1.2.1 | init.d ディレクトリ | 3-11 |
| 3.1.2.2 | rc0.d ディレクトリと rc0 コマンド実行スクリプト ... | 3-12 |
| 3.1.2.3 | rc2.d ディレクトリと rc2 コマンド実行スクリプト ... | 3-13 |
| 3.1.2.4 | rc3.d ディレクトリと rc3 コマンド実行スクリプト ... | 3-15 |
| 3.1.3 | crontabs ディレクトリ | 3-16 |
| 3.2 | 各国語サポート | 3-19 |
| 3.2.1 | ロケールの設定 | 3-21 |
| 3.2.2 | ロケール・カテゴリの変更 | 3-22 |
| 3.2.3 | ロケール変数の制限事項 | 3-24 |
| 3.2.4 | メッセージ・カタログとロケールについての環境変数の設定 | 3-24 |
| 3.3 | 国際化機能のカスタマイズ | 3-24 |
| 3.4 | 時間帯のカスタマイズ | 3-25 |
| 3.5 | 電力管理のカスタマイズ | 3-28 |
| 3.5.1 | dxfpower ユーティリティのグラフィカル・ユーザ・インタフェースの使用 | 3-29 |
| 3.5.2 | sysconfig コマンドの使用 | 3-31 |

| | | |
|---------|----------------------------|------|
| 3.5.2.1 | 電力管理値の変更 | 3-31 |
| 3.5.2.2 | 実行中のカーネルまたは X サーバの変更 | 3-33 |
| 3.5.3 | SysMan Station の使用 | 3-34 |
| 3.6 | スワップ領域の追加 | 3-34 |
| 3.6.1 | 関連ドキュメントおよびユーティリティ | 3-36 |
| 3.6.1.1 | 関連ドキュメント | 3-36 |
| 3.6.1.2 | 関連ユーティリティ | 3-37 |
| 3.6.2 | スワップ領域の割り当て | 3-38 |
| 3.6.3 | スワップ領域の必要量の見積り | 3-40 |
| 3.6.4 | スワップ領域割り当て方法の選択 | 3-41 |
| 3.6.5 | 明白なスワップ領域不足の修正 | 3-42 |

4 カーネルの構成

| | | |
|---------|----------------------------|------|
| 4.1 | 概要 | 4-1 |
| 4.2 | 関連ドキュメントとユーティリティ | 4-2 |
| 4.2.1 | マニュアル | 4-2 |
| 4.2.2 | リファレンス・ページ | 4-3 |
| 4.2.3 | オンライン・ヘルプ | 4-7 |
| 4.3 | インストレーション時のシステム構成 | 4-8 |
| 4.4 | カーネルを再構成する時期と方法の決定 | 4-9 |
| 4.5 | 動的システム構成 | 4-11 |
| 4.5.1 | サブシステムの構成 | 4-12 |
| 4.5.2 | 構成サブシステムのリスト | 4-12 |
| 4.5.3 | サブシステムのタイプの判断 | 4-13 |
| 4.5.4 | サブシステムのアンロード | 4-14 |
| 4.5.5 | 自動構成サブシステムのリストの保守 | 4-14 |
| 4.5.6 | サブシステム属性の管理 | 4-15 |
| 4.5.6.1 | サブシステム属性の現在値の確認 | 4-16 |
| 4.5.6.2 | 実行時に構成可能なサブシステム属性の識別 | 4-18 |

| | | |
|---------|-------------------------------------|------|
| 4.5.6.3 | 属性値の実行時における変更 | 4-18 |
| 4.5.7 | サブシステムおよび属性のリモート管理 | 4-19 |
| 4.5.8 | サブシステム属性データベースの管理 | 4-20 |
| 4.5.8.1 | データベース内の属性のリスト | 4-22 |
| 4.5.8.2 | データベースへの属性の追加 | 4-22 |
| 4.5.8.3 | 新しい定義の既存のデータベース・エントリへのマージ | 4-22 |
| 4.5.8.4 | データベース内の属性の更新 | 4-23 |
| 4.5.8.5 | データベースからの属性定義の削除 | 4-24 |
| 4.5.8.6 | データベースからのサブシステム・エントリの削除 .. | 4-25 |
| 4.6 | 静的システム構成 | 4-25 |
| 4.6.1 | 新しいデバイスをサポートするためのカーネルの構築 | 4-27 |
| 4.6.2 | 選択したカーネル・オプションを追加するためのカーネルの構築 | 4-32 |
| 4.6.3 | システム・ファイルを変更した後のカーネルの構築 | 4-35 |
| 4.7 | 構成ファイル | 4-37 |
| 4.7.1 | /usr/sys/conf の構成ファイル | 4-38 |
| 4.7.1.1 | ターゲット構成ファイル | 4-38 |
| 4.7.1.2 | GENERIC 構成ファイル | 4-38 |
| 4.7.2 | ターゲット構成ファイルの機能拡張 | 4-39 |
| 4.7.3 | param.c ファイル | 4-41 |
| 4.7.4 | システム構成ファイル・エントリ | 4-42 |

5 ディスクの管理

| | | |
|-------|---|-----|
| 5.1 | Disk Configuration ユーティリティを用いたディスク・パーティションの作成 | 5-1 |
| 5.1.1 | パーティション・ウィンドウの構成 | 5-4 |
| 5.1.2 | パーティション・テーブル・ウィンドウ | 5-5 |
| 5.2 | 手動によるディスク・パーティションの作成 | 5-5 |
| 5.2.1 | ユーティリティ | 5-5 |

| | | |
|-------|-------------------------------------|------|
| 5.2.2 | disklabel ユーティリティの使用 | 5-6 |
| 5.2.3 | newfs コマンドによるパーティションの重なりのチェック | 5-9 |
| 5.3 | ディスクのコピー | 5-10 |

6 UNIX ファイル・システム (UFS) の管理

| | | |
|---------|----------------------------------|------|
| 6.1 | ファイル・システムの概要 | 6-2 |
| 6.1.1 | ファイル・システムのディレクトリ階層 | 6-4 |
| 6.1.2 | ディスク・パーティション | 6-5 |
| 6.1.3 | UFS バージョン 4.0 | 6-7 |
| 6.1.4 | ファイル・システムの構造: UFS | 6-10 |
| 6.1.4.1 | ブート・ブロック | 6-11 |
| 6.1.4.2 | スーパーブロック | 6-11 |
| 6.1.4.3 | i ノード・ブロック | 6-11 |
| 6.1.4.4 | データ・ブロック | 6-13 |
| 6.1.5 | ディレクトリおよびファイルのタイプ | 6-13 |
| 6.1.6 | デバイス特殊ファイル | 6-14 |
| 6.2 | コンテキスト依存シンボリック・リンクとクラスタ | 6-15 |
| 6.2.1 | 関連ドキュメント | 6-16 |
| 6.2.2 | CDSL の説明 | 6-16 |
| 6.2.2.1 | CDSL の構造 | 6-18 |
| 6.2.3 | CDSL の管理 | 6-19 |
| 6.2.3.1 | CDSL インベントリの検査 | 6-20 |
| 6.2.3.2 | CDSL の作成 | 6-20 |
| 6.3 | UFS ファイル・システムの手動作成 | 6-21 |
| 6.3.1 | newfs による新しいファイル・システムの作成 | 6-21 |
| 6.3.2 | ファイル・システムをユーザからアクセス可能とする | 6-25 |
| 6.3.3 | /etc/fstab ファイルの使用 | 6-26 |
| 6.3.4 | 手動による UFS ファイル・システムのマウント | 6-30 |
| 6.3.5 | 手動による UFS ファイル・システムのアンマウント | 6-31 |

| | | |
|---------|---|------|
| 6.3.6 | UFS ファイル・システムの拡張 | 6-31 |
| 6.3.6.1 | ディスマウント状態のファイル・システムの拡張 | 6-34 |
| 6.3.6.2 | マウント状態のファイル・システムの拡張 | 6-35 |
| 6.4 | SysMan Menu による UFS ファイル・システムの管理 | 6-35 |
| 6.4.1 | SysMan Menu でのファイル・システムの作業 | 6-38 |
| 6.4.2 | SysMan によるファイル・システムのディスマウント | 6-39 |
| 6.4.3 | SysMan によるマウントされているファイル・システムの 表示 | 6-40 |
| 6.4.4 | SysMan によるファイル・システムのマウント | 6-43 |
| 6.4.5 | SysMan によるローカル・ディレクトリの共用 | 6-48 |
| 6.4.5.1 | ファイル・システムの共用 | 6-49 |
| 6.4.5.2 | 共用ファイル・システムの削除 | 6-50 |
| 6.4.6 | SysMan によるネットワーク・ファイル・システムのマウ ント | 6-50 |
| 6.4.6.1 | 共用ネットワーク・ファイル・システムのマウント .. | 6-52 |
| 6.4.6.2 | ネットワーク・ディレクトリの追加 | 6-53 |
| 6.4.7 | SysMan による UFS ファイル・システムの作成 | 6-53 |
| 6.5 | クォータの管理 | 6-55 |
| 6.5.1 | 物理クォータ制限と論理クォータ制限 | 6-56 |
| 6.5.2 | ファイル・システム・クォータの設定 | 6-57 |
| 6.5.3 | ユーザ・アカウントへのファイル・システム・クォータの 設定 | 6-59 |
| 6.5.4 | ファイル・システム・クォータの確認 | 6-60 |
| 6.6 | ファイル・システムのバックアップとリストア | 6-60 |
| 6.7 | ファイル・システムのモニタリングとチューニング | 6-61 |
| 6.7.1 | UFS の一貫性の検査 | 6-61 |
| 6.7.2 | ファイル・システムのディスク使用量のモニタリング | 6-62 |
| 6.7.2.1 | 使用可能な空きスペースの確認 | 6-63 |
| 6.7.2.2 | ディスク使用量の確認 | 6-64 |

| | | |
|----------|--|------|
| 6.7.3 | UFS 読み取り効率の改善 | 6-66 |
| 6.8 | ファイル・システムのトラブルシューティング | 6-66 |
| 7 | ユーザ・アカウントおよびグループの管理 | |
| 7.1 | アカウント管理のオプションと制限 | 7-1 |
| 7.1.1 | 管理ユーティリティ | 7-2 |
| 7.1.2 | ユーティリティの使用についての説明と制限 | 7-3 |
| 7.1.3 | 関連ドキュメント | 7-6 |
| 7.1.3.1 | マニュアル | 7-6 |
| 7.1.3.2 | リファレンス・ページ | 7-7 |
| 7.1.3.3 | オンライン・ヘルプ | 7-7 |
| 7.1.4 | 関連ユーティリティ | 7-7 |
| 7.2 | アカウント管理 - クイック・スタート | 7-9 |
| 7.2.1 | システム・セットアップの際の初期アカウントの作成 | 7-9 |
| 7.2.2 | アカウント・マネージャ (dxaccounts) GUI の使用 | 7-9 |
| 7.2.3 | SysMan Menu の [アカウント] オプションの使用 | 7-10 |
| 7.2.4 | コマンド行ユーティリティの使用 | 7-12 |
| 7.2.5 | Advanced Server for UNIX | 7-12 |
| 7.3 | ユーザ・アカウントおよびグループの概要 | 7-13 |
| 7.3.1 | システム・ファイル | 7-13 |
| 7.3.2 | 識別子 (UID と GID) の概要 | 7-15 |
| 7.3.3 | パスワード・ファイル | 7-17 |
| 7.3.4 | グループ・ファイル | 7-20 |
| 7.4 | ユーザ・アカウントの管理 | 7-21 |
| 7.4.1 | SysMan Menu の [アカウント] オプションの使用 | 7-22 |
| 7.4.1.1 | アカウント情報の収集 | 7-22 |
| 7.4.1.2 | フィルタ・オプションと表示オプションの設定 | 7-24 |
| 7.4.1.3 | フィルタ・オプションの使用 | 7-25 |
| 7.4.1.4 | ローカル・アカウントの作成と変更 | 7-27 |

| | | |
|---------|--|------|
| 7.4.1.5 | ローカル・アカウントの削除 | 7-29 |
| 7.4.1.6 | LDAP アカウントと NIS アカウントの作成と変更 ... | 7-29 |
| 7.4.1.7 | LDAP アカウントと NIS アカウントの削除 | 7-31 |
| 7.4.2 | アカウント・マネージャ (dxaccounts) の使用 | 7-31 |
| 7.4.2.1 | アカウントの追加と変更 | 7-32 |
| 7.4.2.2 | アカウントの削除 | 7-34 |
| 7.4.2.3 | アカウントの検索と選択 | 7-34 |
| 7.4.2.4 | アカウントのコピー | 7-35 |
| 7.4.2.5 | password オプションの使用 | 7-36 |
| 7.4.2.6 | アカウント・マネージャ (dxaccounts) の一般オプション | 7-36 |
| 7.5 | グループの管理 | 7-38 |
| 7.5.1 | SysMan Menu のグループ管理オプションの使用 | 7-38 |
| 7.5.1.1 | グループ情報の収集 | 7-39 |
| 7.5.1.2 | グループの作成と変更 | 7-40 |
| 7.5.2 | アカウント・マネージャ (dxaccounts) の使用 | 7-41 |
| 7.5.2.1 | グループの追加 | 7-42 |
| 7.5.2.2 | グループの変更 | 7-43 |
| 7.5.2.3 | グループの削除 | 7-43 |
| 7.5.2.4 | グループの検索 | 7-44 |
| 7.6 | Windows ドメイン・アカウントとグループの管理 | 7-44 |
| 7.6.1 | 同期化されたアカウントの管理 | 7-47 |
| 7.6.1.1 | SysMan Menu のアカウント・オプションとグループ・オプションの使用 | 7-47 |
| 7.6.1.2 | アカウント・マネージャ (dxaccounts) の使用 | 7-48 |
| 7.6.1.3 | コマンド行ユーティリティの使用 | 7-49 |
| 7.6.1.4 | ASU ドメイン・ユーザ・マネージャの使用 | 7-53 |
| 7.6.1.5 | ASU net コマンドの使用 | 7-54 |
| 7.6.2 | Windows 2000 シングル・サイン・オン | 7-54 |

| | | |
|----------|---|------|
| 7.6.2.1 | シングル・サイン・オンのインストール要件 .. | 7-55 |
| 7.6.2.2 | シングル・サイン・オン・ソフトウェアのインストール | 7-56 |
| 7.6.2.3 | シングル・サイン・オン・アカウントの作成についての UNIX の要件 | 7-56 |
| 7.6.2.4 | シングル・サイン・オンのアカウントおよびグループの作成 | 7-58 |
| 7.6.2.5 | シングル・サイン・オンのシステム・ファイル | 7-59 |
| 8 | プリント・サービスの管理 | |
| 8.1 | プリント管理タスク | 8-2 |
| 8.1.1 | プリンタの接続方法 | 8-2 |
| 8.1.2 | プリンタの管理方法 | 8-3 |
| 8.1.2.1 | Printer Configuration ユーティリティ (printconfig) の使用 | 8-3 |
| 8.1.2.2 | lprsetup ユーティリティの使用 | 8-4 |
| 8.1.2.3 | 手動によるシステム・ファイルの編集 | 8-4 |
| 8.1.3 | Advanced Printing Software | 8-5 |
| 8.1.4 | 関連ドキュメント | 8-5 |
| 8.1.4.1 | マニュアル | 8-5 |
| 8.1.4.2 | リファレンス・ページ | 8-6 |
| 8.1.4.3 | オンライン・ヘルプ | 8-8 |
| 8.1.5 | システム・ファイル | 8-8 |
| 8.1.6 | 関連ユーティリティ | 8-10 |
| 8.2 | 情報収集 | 8-11 |
| 8.2.1 | 直接接続とネットワーク接続のプリンタ | 8-12 |
| 8.2.2 | リモート・プリンタ | 8-15 |
| 8.3 | プリンタの構成 | 8-16 |
| 8.3.1 | printconfig を使用した TCP/IP プリントの構成 | 8-17 |
| 8.3.1.1 | printconfig を使用した TCP/IP プリンタの構成 | 8-18 |

| | | |
|-----------|---|------|
| 8.3.1.2 | TCP/IP プrint の設定に必要な手動による追加手順 | 8-19 |
| 8.3.2 | printconfig を使用した直接接続のプリンタのインストール | 8-21 |
| 8.3.3 | printconfig を使用したリモート・プリンタのセットアップ | 8-24 |
| 8.3.4 | printconfig による PC プrint ・キューの構成 | 8-25 |
| 8.3.5 | lprsetup によるプリンタのインストール | 8-26 |
| 8.3.6 | Advanced Printing Software のための Print ・シンボル | 8-30 |
| 8.4 | Print ・システムの日常保守 | 8-31 |
| 8.4.1 | プリンタの追加 | 8-31 |
| 8.4.2 | プリンタ構成の変更 | 8-32 |
| 8.4.3 | プリンタの削除 | 8-32 |
| 8.4.4 | ローカル・Print ・ジョブおよびキューの制御 | 8-33 |
| 8.4.5 | プリンタ課金機能の設定 | 8-35 |
| 8.5 | 参照情報 | 8-37 |
| 8.5.1 | /etc/printcap ファイル | 8-37 |
| 8.5.2 | /etc/printcap のデータ | 8-41 |
| 8.5.2.1 | プリンタ名 | 8-41 |
| 8.5.2.2 | プリンタ・タイプ | 8-42 |
| 8.5.2.3 | プリンタの別名 | 8-42 |
| 8.5.2.4 | デバイス特殊ファイル | 8-43 |
| 8.5.2.5 | 接続タイプ | 8-44 |
| 8.5.2.6 | スプール・ディレクトリ | 8-45 |
| 8.5.2.6.1 | スプール・ディレクトリのファイル | 8-46 |
| 8.5.2.6.2 | スプール・ディレクトリの作成 | 8-48 |
| 8.5.2.7 | ポー・レート | 8-48 |
| 8.5.3 | ライン・プリンタ・デーモン | 8-48 |
| 8.5.4 | エラー・ロギング | 8-49 |
| 8.5.5 | Print ・フィルタとフィルタ・ディレクトリ | 8-50 |

| | | |
|-------|----------------------------|------|
| 8.5.6 | フラグ・ビット | 8-52 |
| 8.5.7 | モード・ビット | 8-54 |
| 8.5.8 | リモート・プリンタの特性 | 8-55 |
| 8.6 | プリント・フィルタ | 8-56 |
| 8.6.1 | pcfof プリント・フィルタ | 8-56 |
| 8.6.2 | wwpsmf プリント・フィルタ | 8-56 |
| 8.6.3 | フィルタ使用における既知の制限事項 | 8-57 |
| 8.7 | プリンタのテストとトラブルシューティング | 8-58 |

9 アーカイブ・サービスの管理

| | | |
|---------|---------------------------------|------|
| 9.1 | バックアップ作業の概要 | 9-2 |
| 9.2 | データ・ファイルとシステム・ファイルのバックアップ | 9-4 |
| 9.3 | バックアップ・スケジュールの選択 | 9-5 |
| 9.4 | バックアップの方法 | 9-6 |
| 9.5 | バックアップ実行の準備 | 9-7 |
| 9.5.1 | 関連ドキュメント | 9-8 |
| 9.5.1.1 | マニュアル | 9-8 |
| 9.5.1.2 | リファレンス・ページ | 9-8 |
| 9.5.1.3 | オンライン・ヘルプ | 9-9 |
| 9.5.2 | システム・ファイル | 9-9 |
| 9.5.3 | 関連ユーティリティ | 9-10 |
| 9.5.4 | 事前に行う作業 | 9-11 |
| 9.6 | dump コマンドの使用 | 9-15 |
| 9.6.1 | フル・バックアップの実行 | 9-16 |
| 9.6.2 | 増分バックアップの実行 | 9-18 |
| 9.6.3 | リモート・バックアップの実行 | 9-19 |
| 9.6.4 | バックアップ・スクリプトの使用 | 9-19 |
| 9.7 | データのリストア | 9-20 |
| 9.7.1 | ファイル・システムのリストア | 9-21 |

| | | |
|----------|---|------|
| 9.7.2 | 手動によるファイルのリストア | 9-22 |
| 9.7.3 | ファイルの対話式リストア | 9-23 |
| 9.7.4 | ファイルのリモート・リストア | 9-26 |
| 9.7.5 | システム (ルート) ディスクの回復または複製 | 9-27 |
| 9.7.5.1 | 回復または複製の準備 | 9-28 |
| 9.7.5.2 | リストア要件の判断 | 9-30 |
| 9.7.5.3 | 手順の適用 | 9-33 |
| 9.7.5.4 | その他のルート・ディスク複製方法の使用 | 9-38 |
| 9.7.6 | /usr および /var ファイル・システムのリストア | 9-39 |
| 9.8 | コマンド行ユーティリティ tar , pax , および cpio の使用 | 9-40 |
| 9.9 | dxarchiver の使用 | 9-42 |
| 9.10 | スタンドアロン・システム・カーネルのテープ上での作成 ... | 9-45 |
| 9.10.1 | テープ・デバイスの必要条件 | 9-46 |
| 9.10.2 | btcreate ユーティリティの使用 | 9-47 |
| 9.10.2.1 | 情報の収集 | 9-47 |
| 9.10.2.2 | SAS カーネルの作成 | 9-49 |
| 9.10.3 | btextract ユーティリティの使用 | 9-49 |
| 9.10.4 | SysMan Menu の [ブータブル・テープの作成] オプション の使用 | 9-50 |

10 システム課金サービスの管理

| | | |
|--------|---|-------|
| 10.1 | 課金機能の概要 | 10-1 |
| 10.1.1 | 課金シェル・スクリプトと課金コマンド | 10-3 |
| 10.1.2 | 課金ファイル | 10-5 |
| 10.2 | 課金機能の設定 | 10-11 |
| 10.2.1 | rc.config ファイルの編集 | 10-12 |
| 10.2.2 | qacct ファイル , pacct ファイル , および fee ファイルの確 認 | 10-12 |
| 10.2.3 | holidays ファイルの編集 | 10-13 |

| | | |
|--------|---------------------------|-------|
| 10.2.4 | crontab ファイルの変更 | 10-13 |
| 10.3 | 課金機能のスタートアップと停止 | 10-15 |
| 10.4 | 接続セッション課金機能 | 10-15 |
| 10.4.1 | wtmpfix コマンド | 10-18 |
| 10.4.2 | fwtmp コマンド | 10-19 |
| 10.4.3 | acwttmp コマンド | 10-20 |
| 10.4.4 | ac コマンド | 10-21 |
| 10.4.5 | acctcon1 コマンド | 10-22 |
| 10.4.6 | acctcon2 コマンド | 10-23 |
| 10.4.7 | prctmp シェル・スクリプト | 10-23 |
| 10.4.8 | lastlogin シェル・スクリプト | 10-24 |
| 10.4.9 | last コマンド | 10-24 |
| 10.5 | プロセス課金 | 10-25 |
| 10.5.1 | accton コマンド | 10-27 |
| 10.5.2 | turnacct シェル・スクリプト | 10-28 |
| 10.5.3 | ckpacct シェル・スクリプト | 10-28 |
| 10.5.4 | acctcom コマンド | 10-29 |
| 10.5.5 | sa コマンド | 10-30 |
| 10.5.6 | acctcms コマンド | 10-32 |
| 10.5.7 | acctpre1 コマンド | 10-33 |
| 10.5.8 | acctpre2 コマンド | 10-35 |
| 10.5.9 | lastcomm コマンド | 10-35 |
| 10.6 | ディスク使用量課金機能 | 10-36 |
| 10.6.1 | dodisk シェル・スクリプト | 10-36 |
| 10.6.2 | diskusg コマンド | 10-37 |
| 10.6.3 | acctdusg コマンド | 10-38 |
| 10.6.4 | acctdisk コマンド | 10-39 |
| 10.7 | システム管理サービス課金機能 | 10-39 |
| 10.8 | プリンタ課金機能 | 10-40 |

| | | |
|-----------|---|-------|
| 10.9 | 日次, 要約, 月次の各レポート・ファイルの作成 | 10-41 |
| 10.9.1 | runacct シェル・スクリプト | 10-42 |
| 10.9.1.1 | runacct シェル・スクリプト・エラーの修正 | 10-43 |
| 10.9.1.2 | エラーの例と対策 | 10-45 |
| 10.9.2 | acctmerg コマンド | 10-46 |
| 10.9.3 | prtacct シェル・スクリプト | 10-47 |
| 10.9.4 | prdaily シェル・スクリプト | 10-48 |
| 10.9.5 | monacct シェル・スクリプト | 10-48 |
| 11 | システムのモニタリングとテスト | |
| 11.1 | モニタリングとテストの概要 | 11-1 |
| 11.1.1 | システムのモニタリングの概要 | 11-2 |
| 11.1.2 | コマンドおよびユーティリティの概要 | 11-4 |
| 11.1.2.1 | コマンド行ユーティリティ | 11-5 |
| 11.1.2.2 | SysMan Menu のモニタリング作業およびチューニング作業 | 11-6 |
| 11.1.2.3 | SysMan Station | 11-8 |
| 11.1.2.4 | X11 準拠のグラフィカル・ユーザ・インタフェース .. | 11-8 |
| 11.1.2.5 | 高度なモニタリング・ユーティリティ | 11-10 |
| 11.1.3 | 関連ドキュメント | 11-11 |
| 11.2 | モニタリング・ユーティリティの構成および使用 | 11-12 |
| 11.2.1 | collect ユーティリティによるシステム・データの記録 ... | 11-12 |
| 11.2.2 | sys_check ユーティリティの使用 | 11-14 |
| 11.2.3 | Monitoring Performance History ユーティリティの使用 . | 11-15 |
| 11.3 | 環境モニタリングと envmond/envconfig | 11-17 |
| 11.3.1 | ロード可能カーネル・モジュール | 11-19 |
| 11.3.1.1 | ロード可能カーネル属性の指定 | 11-19 |
| 11.3.1.2 | プラットフォーム固有の機能の取得 | 11-20 |
| 11.3.2 | Server System MIB サブエージェント | 11-21 |

| | | |
|-----------|---|-------|
| 11.3.3 | 環境モニタリング・デーモン | 11-22 |
| 11.3.4 | envconfig を使用した envmond デーモンの構成 | 11-22 |
| 11.3.5 | ユーザ定義可能なメッセージ | 11-23 |
| 11.4 | システム・エクササイズの使用 | 11-23 |
| 11.4.1 | システム・エクササイズの実行 | 11-24 |
| 11.4.2 | エクササイズ診断の使用 | 11-25 |
| 11.4.3 | ファイル・システムのエクササイズ | 11-26 |
| 11.4.4 | システム・メモリのエクササイズ | 11-26 |
| 11.4.5 | 共用メモリのエクササイズ | 11-27 |
| 11.4.6 | ターミナル通信システムのエクササイズ | 11-28 |
| 12 | 基本システム・イベント・チャネルの管理 | |
| 12.1 | 基本イベント・ロギング機能 | 12-1 |
| 12.1.1 | システム・イベント・ロギング | 12-3 |
| 12.1.2 | バイナリ・イベント・ロギング | 12-4 |
| 12.2 | イベント・ロギングの構成 | 12-5 |
| 12.2.1 | 構成ファイルの編集 | 12-6 |
| 12.2.1.1 | syslog.conf ファイルの編集 | 12-6 |
| 12.2.1.2 | Event Manager を使用するための syslog の構成 | 12-11 |
| 12.2.1.3 | binlog.conf ファイルの編集 | 12-12 |
| 12.2.2 | リモート・メッセージと syslog のセキュリティ | 12-15 |
| 12.2.3 | 特殊ファイルの作成 | 12-17 |
| 12.2.4 | イベント・ロギング・デーモンの起動と停止 | 12-17 |
| 12.2.4.1 | syslogd デーモン | 12-17 |
| 12.2.4.2 | binlogd デーモン | 12-19 |
| 12.2.5 | カーネルのバイナリ・イベント・ロガーの設定 | 12-20 |
| 12.3 | システム・クラッシュ後のイベント・ログの回復 | 12-21 |
| 12.4 | ログ・ファイルの管理 | 12-22 |
| 12.5 | /var/adm/messages 内の起動ログ・メッセージ | 12-23 |

13 Event Manager の使用方法

| | | |
|----------|---|-------|
| 13.1 | Event Manager の概要 | 13-1 |
| 13.1.1 | Event Manager の機能 | 13-3 |
| 13.1.2 | Event Manager イベントとは | 13-4 |
| 13.1.3 | Event Manager のコンポーネント | 13-6 |
| 13.1.3.1 | Event Manager のコマンド行ユーティリティ | 13-8 |
| 13.1.3.2 | Event Manager のアプリケーション・プログラミング・インタフェース | 13-10 |
| 13.1.3.3 | Event Manager のシステム・ファイル | 13-11 |
| 13.1.4 | 関連ユーティリティ | 13-14 |
| 13.2 | Event Manager の管理 | 13-15 |
| 13.2.1 | Event Manager の起動と停止 | 13-16 |
| 13.2.2 | Event Manager の構成 | 13-17 |
| 13.2.2.1 | Event Manager のデーモンの構成 | 13-17 |
| 13.2.2.2 | Event Manager のチャンネルの構成 | 13-19 |
| 13.2.2.3 | Event Manager ロガーの構成 | 13-21 |
| 13.2.2.4 | ロガーの 2 次構成ファイル | 13-24 |
| 13.2.2.5 | イベントの紛失を防ぐためのバッファ・サイズの変更 | 13-25 |
| 13.2.3 | セキュリティについて | 13-25 |
| 13.2.3.1 | ユーザの認証 | 13-26 |
| 13.2.3.2 | ユーザの権限 | 13-26 |
| 13.2.3.3 | 認証を含むリモート・アクセス | 13-28 |
| 13.2.4 | ログ・ファイルの管理 | 13-32 |
| 13.2.5 | イベント・テンプレート | 13-33 |
| 13.2.6 | 新しい Event Manager クライアントのインストール | 13-34 |
| 13.2.7 | binlog イベント変換ユーティリティの構成 | 13-35 |
| 13.3 | システム管理での Event Manager の使用 | 13-39 |
| 13.3.1 | evmshow を使用した , イベントの表示 | 13-39 |

| | | |
|-----------|--|-------|
| 13.3.2 | イベント・フィルタの概要 | 13-42 |
| 13.3.3 | evmget を使用した, 格納されているイベントの取り出し | 13-44 |
| 13.3.4 | evmsort を使用した, イベントのソート | 13-47 |
| 13.3.5 | -A オプションを使用した, コマンド文字列の簡略化 | 13-48 |
| 13.3.6 | evmwatch を使用した, イベントの監視 | 13-49 |
| 13.3.7 | evmpost を使用した, クイック・メッセージ・イベントの ポスト | 13-51 |
| 13.3.8 | 登録済みのイベントのリスト表示 | 13-52 |
| 13.3.9 | シェル・スクリプトからのイベントのポスト | 13-53 |
| 13.3.10 | Event Manager のマーク・イベントについて | 13-57 |
| 13.3.11 | SysMan イベント・ビューアを使用した, イベントの表示 | 13-58 |
| 13.3.12 | 高度な選択とフィルタリング技法 | 13-59 |
| 13.3.12.1 | 時刻によるフィルタリング | 13-60 |
| 13.3.12.2 | 詳細表示のイベントを選択する event-id の使用 | 13-61 |
| 13.3.12.3 | 予約済みのコンポーネント名の検索 | 13-62 |
| 13.3.12.4 | フィルタ・ファイルの使用 | 13-63 |
| 13.3.13 | イベントのロギングと転送 | 13-64 |
| 13.3.13.1 | イベントのロギング | 13-65 |
| 13.3.13.2 | 転送を使用したイベントの自動処理 | 13-65 |
| 13.3.13.3 | リモート・システムからのイベントのロギング | 13-67 |
| 13.4 | Event Manager のトラブルシューティング | 13-68 |
| 14 | クラッシュ・ダンプの管理 | |
| 14.1 | クラッシュ・ダンプの概要 | 14-1 |
| 14.1.1 | 関連ドキュメントとユーティリティ | 14-2 |
| 14.1.1.1 | マニュアル | 14-2 |
| 14.1.1.2 | リファレンス・ページ | 14-3 |
| 14.1.1.3 | SysMan Menu アプリケーション | 14-4 |
| 14.1.2 | クラッシュ・ダンプ時に使用されるファイル | 14-4 |

| | | |
|--------|--|-------|
| 14.2 | クラッシュ・ダンプのアプリケーション | 14-5 |
| 14.2.1 | 「システム・ダンプの構成」アプリケーションの使用方法 | 14-5 |
| 14.2.2 | 「ダンプ・スナップショットの作成」アプリケーションの 使用方法 | 14-8 |
| 14.3 | クラッシュ・ダンプの作成 | 14-10 |
| 14.3.1 | generic サブシステムのダンプ・カーネル属性の設定 | 14-10 |
| 14.3.2 | クラッシュ・ダンプ・ファイルの作成 | 14-13 |
| 14.3.3 | クラッシュ・ダンプのロギング | 14-14 |
| 14.3.4 | スワップ領域 | 14-16 |
| 14.3.5 | クラッシュ・ダンプ領域の計画 | 14-20 |
| 14.3.6 | クラッシュ・ダンプ・ファイルに対するファイル・システム 領域の割り当て | 14-21 |
| 14.4 | クラッシュ・ダンプの内容と方式の選択 | 14-22 |
| 14.4.1 | 1 次スワップ・パーティションにおけるクラッシュ・ダンプ のしきい値の調整 | 14-23 |
| 14.4.2 | 部分クラッシュ・ダンプにユーザ・ページ・テーブルを含 める | 14-24 |
| 14.4.3 | 部分クラッシュ・ダンプかフル・クラッシュ・ダンプかの 選択 | 14-25 |
| 14.4.4 | ダンプ圧縮の期待値 | 14-25 |
| 14.4.5 | 圧縮されていないクラッシュ・ダンプの選択と使用 | 14-26 |
| 14.4.6 | 除外メモリへのダンプ | 14-27 |
| 14.4.7 | リモート・ホストへのダンプ | 14-29 |
| 14.5 | 手動によるクラッシュ・ダンプの生成 | 14-29 |
| 14.5.1 | 実行中のシステムでの継続ダンプ | 14-30 |
| 14.5.2 | ハングしたシステムでの強制クラッシュ・ダンプ | 14-31 |
| 14.6 | クラッシュ・ダンプ・ファイルの保存とアーカイブ | 14-32 |
| 14.6.1 | クラッシュ・ダンプ・ファイルの圧縮 | 14-32 |
| 14.6.2 | 部分クラッシュ・ダンプ・ファイルの圧縮解除 | 14-33 |

A 管理ユーティリティ

| | | |
|--------|---|------|
| A.1 | X11 グラフィカル・ユーザ・インタフェース (CDE アプリケーション・マネージャ) | A-1 |
| A.2 | SysMan Menu タスクと関連ユーティリティ | A-8 |
| A.2.1 | アカウント | A-9 |
| A.2.2 | ハードウェア | A-10 |
| A.2.3 | メール | A-10 |
| A.2.4 | モニタリング/チューニング | A-11 |
| A.2.5 | ネットワーク | A-12 |
| A.2.6 | プリント | A-16 |
| A.2.7 | セキュリティ | A-16 |
| A.2.8 | ソフトウェア | A-17 |
| A.2.9 | ストレージ | A-18 |
| A.2.10 | サポートとサービス | A-19 |
| A.2.11 | 一般的なタスク | A-20 |

索引

例

| | | |
|------|---------------------------------------|-------|
| 2-1 | 典型的なシャットダウン・シーケンス | 2-34 |
| 6-1 | 省略時のパーティション | 6-6 |
| 7-1 | usermod による省略時の環境変数の変更 | 7-45 |
| 12-1 | 変換されたイベントの例 | 12-4 |
| 12-2 | syslog_evm.conf ファイルのエントリの例 | 12-11 |
| 13-1 | Event Manager デーモン構成ファイルのエントリの例 | 13-18 |
| 13-2 | Event Manager チャネル構成ファイルの例 | 13-19 |
| 13-3 | Event Manager ロガー構成ファイルのエントリの例 | 13-21 |
| 13-4 | Event Manager 権限ファイルのエントリの例 | 13-27 |
| 13-5 | DECevent 変換を表示する binlogd イベント | 13-36 |

| | | |
|------|---------------------------------------|-------|
| 13-6 | リモート・ロギングを行うためのロガー構成ファイルのエントリの例 | 13-67 |
|------|---------------------------------------|-------|

図

| | | |
|------|---|-------|
| 1-1 | 「システム・セットアップ」のグラフィカル・ユーザ・インタフェース | 1-8 |
| 1-2 | クイック・セットアップ | 1-9 |
| 1-3 | カスタム・セットアップ | 1-10 |
| 1-4 | CDE のツール・ドロワおよび SysMan Station アイコン | 1-15 |
| 1-5 | 「SysMan Applications」パネル | 1-16 |
| 1-6 | SysMan Menu | 1-25 |
| 1-7 | SysMan Station のメイン・ウィンドウ | 1-31 |
| 1-8 | AdvFS_Fileystems View | 1-34 |
| 1-9 | Hardware View | 1-35 |
| 4-1 | 構成ファイルのディレクトリの階層構造 | 4-37 |
| 8-1 | Printconfig のメイン・ウィンドウ | 8-17 |
| 13-1 | イベント・モデル | 13-5 |
| 13-2 | Event Manager コンポーネントのモデル | 13-6 |
| 14-1 | 「システム・ダンプの構成」アプリケーション | 14-6 |
| 14-2 | 「ダンプ・スナップショットの作成」アプリケーション | 14-9 |
| 14-3 | 省略時の dump_sp_threshold 属性の設定 | 14-18 |
| 14-4 | 複数のデバイスに書き込まれるクラッシュ・ダンプ | 14-19 |

表

| | | |
|-----|-----------------------------|------|
| 2-1 | コンソール環境変数 | 2-17 |
| 2-2 | boot_osflags 変数のオプション | 2-18 |
| 2-3 | date コマンドのパラメータ | 2-27 |
| 3-1 | ロケール・サポート・ファイル | 3-19 |
| 3-2 | ロケール環境変数 | 3-23 |
| 6-1 | ディスク・パーティション・テーブル | 6-24 |

| | | |
|------|---|-------|
| 7-1 | アカウントおよびグループの管理ユーティリティ | 7-2 |
| 7-2 | アカウント管理ワークシート | 7-23 |
| 7-3 | データの例を記入したアカウント管理ワークシート | 7-24 |
| 7-4 | グループ管理ワークシート | 7-40 |
| 8-1 | TCP/IP ソケット番号 | 8-20 |
| 8-2 | lprsetup のオプション | 8-27 |
| 8-3 | lpc コマンドの引数 | 8-34 |
| 8-4 | 通信ポートとプリンタのデバイス特殊ファイル | 8-43 |
| 8-5 | フラグ・ビット | 8-53 |
| 8-6 | モード・ビット | 8-54 |
| 8-7 | 非 PostScript および PostScript フィルタ | 8-57 |
| 9-1 | 回復の準備 | 9-30 |
| 10-1 | 課金コマンドおよび課金シェル・スクリプト | 10-4 |
| 10-2 | /var/adm ディレクトリ内のデータベース・ファイル | 10-6 |
| 10-3 | /var/adm/acct/nite ディレクトリの日次ファイル | 10-8 |
| 10-4 | /var/adm/acct/sum ディレクトリ内の要約ファイル | 10-10 |
| 10-5 | /var/adm/acct/fiscal ディレクトリ内の月次ファイル | 10-10 |
| 10-6 | utmp ASCII 形式によるファイル構成要素 | 10-18 |
| 10-7 | tacct ファイル・フォーマット | 10-26 |
| 11-1 | カーネル・モジュールで定義されているパラメータ | 11-20 |
| 11-2 | get_info() 関数タイプ | 11-20 |
| 11-3 | サーバ・サブシステム変数のマッピング | 11-21 |
| 13-1 | Event Manager のコマンド行ユーティリティ | 13-9 |
| 13-2 | Event Manager の管理ユーティリティ | 13-9 |
| A-1 | システム管理のシステム設定アプリケーション | A-4 |
| A-2 | システム管理の日常管理アプリケーション | A-5 |
| A-3 | システム管理のモニタリング/チューニング・アプリケーション | A-6 |
| A-4 | システム管理のソフトウェア管理アプリケーション | A-7 |

| | | |
|-----|------------------------------|-----|
| A-5 | システム管理のストレージ管理アプリケーション | A-7 |
| A-6 | システム管理ツール | A-8 |

まえがき

本書では、AlphaServer 上で稼働する Tru64™ UNIX オペレーティング・システムの管理作業について説明します。

本書の対象読者

本書は、システム管理者だけを対象としています。本書は、読者がシステム管理者として、UNIX オペレーティング・システム概念、コマンド、およびサポートされているハードウェア構成とソフトウェア構成に関する知識があることを前提に説明しています。管理者は、UNIX のシステム管理操作に慣れており、UNIX システムの可用性を高く保つために必要な手順をすべて知っていなければなりません。本書は、システム管理者の教育や UNIX システムのインストールを目的としたマニュアルではありません。

新しい機能および変更された機能

本書の改訂内容は、以下のとおりです。

- 第 5 章は、この章の以前のバージョンのサブセットです。削除した項目は、『ハードウェア管理ガイド』に移しました。
- 第 8 章は、TCP/IP を使用したプリンタ構成に焦点を当てるように改訂されました。この章では、アップデートされた `lprsetup` 手順と、`lpc` コマンドも説明しています。
- 9.7.5 項が改訂されました。
- 11.3 節は、HP Insight Manager の一部としてコマンド行で使用する環境モニタリングの説明と、センサ・モニタリング(一部の最新のハードウェア・プラットフォームのみで使用可能)の使用方法が改訂されました。
- 第 13 章は、Event Manager の起動と停止、イベントの紛失を防ぐための構成、リモート認証、リモート・ロギング、およびイベント接続でのメッセージ容量の増加について改訂されました。
- 第 14 章では、必要に応じてクラッシュ・ダンプを構成するためのグラフィカル・ユーザ・インタフェースの説明と、システム・メモリのスナップショットをダンプ・ファイルに保存する方法が追加されました。

古いバージョンの UNIX オペレーティング・システムからアップデートする場合は、中間のリリースで実装されたすべての変更についての情報を参照することもできます。この情報は、ソフトウェア・ドキュメント CD-ROM 上の HTML ファイル (特に、『*New and Changed Features from Previous Releases*』) にあります。また、以下のオンライン・リソースもご利用になれます。

- 本書の旧バージョンは、下記の URL の Tru64 UNIX の Web サイトにあります。

<http://tru64unix.compaq.co.jp/document/>

本書の改訂履歴は、これらの旧バージョンの「新しい機能および変更された機能」の項を参照してください。

前述の Web サイトでは、メディアで配布されるドキュメントには含まれていない、技術的な更新情報を入手できます。

オペレーティング・システムの管理に関する多くのコマンドとユーティリティに、新機能が追加されています。本書のコマンド例と手順については、正しいことが確認されています。場合によっては、関連のリファレンス・ページが完全に改訂されています。また、一部の情報をリファレンス・ページに移して、重複した内容をなくし、本書のサイズを小さくしました。

本書の構成

本書は以下の章から構成されています。

| | |
|-------|--|
| 第 1 章 | システム管理作業を実行するために使用するメソッドやツールについて説明します。 |
| 第 2 章 | オペレーティング・システムの起動方法とシャットダウン方法について説明します。また、不意のシステム・シャットダウンから回復する方法についても説明します。 |
| 第 3 章 | オペレーティング・システム環境を最適化するために、オペレーティング・システムのファイルやオペレーティング・システムのコンポーネントをカスタマイズする方法について説明します。 |
| 第 4 章 | オペレーティング・システムのカーネルを動的および静的に構成する方法について説明します。 |

| | |
|--------|--|
| 第 5 章 | ディスクのパーティション分割，ディスクのコピー，およびディスクのモニタリングを含む，ディスク管理に関連するシステム管理タスクについて説明します。 |
| 第 6 章 | UNIX ファイル・システム (UFS) を管理する方法について説明します。 |
| 第 7 章 | オペレーティング・システムのユーザ・アカウントおよびユーザ・グループの管理方法について説明します。 |
| 第 8 章 | プリント・サービス・システムの管理とプリンタの構成方法について説明します。 |
| 第 9 章 | オペレーティング・システムのアーカイブ・サービスの管理を行って，大容量ストレージ・デバイスのバックアップとリストアを行う方法について説明します。 |
| 第 10 章 | オペレーティング・システムのリソース課金サービスの管理について説明します。 |
| 第 11 章 | モニタリング・ユーティリティおよびテスト・ユーティリティについて説明します。 |
| 第 12 章 | オペレーティング・システムの基本的なイベント・ロギング・サービスのセットアップ方法および管理方法について説明します。 |
| 第 13 章 | EVM，拡張イベント管理，およびログ・メカニズムのセットアップ方法および管理方法について説明します。 |
| 第 14 章 | クラッシュ・ダンプのセットアップ方法および管理方法について説明します。 |
| 付録 A | 管理ユーティリティの一覧を示します。 |

関連資料

以下のドキュメントには，本書の各章の説明を補足する重要な情報が含まれています。

- 『インストレーション・ガイド』および『インストレーション・ガイド — 上級ユーザ編』

オペレーティング・システムのインストールについて説明しています。これらのマニュアルでは、ソフトウェアのインストールやインストールのクローニングなどの重要な管理作業について、詳細に説明します。

- 『ハードウェア管理ガイド』

本書の姉妹編です。『ハードウェア管理ガイド』では、Tru64 UNIX オペレーティング・システムで制御されるシステム・ハードウェアを保守するために必要な作業を説明しています。

- ストレージ構成についての重要な情報 (ストレージ・アレイの構成と保守など) は、StorageWorks のドキュメントを参照してください。オペレーティング・システムで提供されるユーティリティの他に、SWCC (StorageWorks Command Console) のような StorageWorks ソフトウェア・アプリケーションを使用します。Web 上にあるリソースについては、関連資料を参照してください。

- 『ネットワーク管理ガイド：サービス編』および『ネットワーク管理ガイド：接続編』

ネットワークの設定、構成、トラブルシューティングについて説明しています。

- Advanced Server for UNIX (ASU) の『*Advanced Server for UNIX* コンセプトとプランニング・ガイド』および『*Advanced Server for UNIX* インストール/管理ガイド』

Windows ドメイン・アカウントの管理、PC ユーザとのプリンタの共用について説明しています。これらのマニュアルは、ASU ソフトウェアの「*Associated Products, Volume 2*」CD-ROM に入っています。

- 『セキュリティ管理ガイド』

アカウント管理およびファイル・システムの共有に影響のあるセキュリティについて説明しています。

- 『*AdvFS* 管理ガイド』および『*Logical Storage Manager*』

拡張ファイル・システムおよびストレージ管理について説明しています。

- 『システムの構成とチューニング』

システム性能のチューニングおよびカーネルの高度な構成について説明しています。

- 『リリース・ノート』

オペレーティング・システムの機能を使用する上での制限事項などの、重要な情報について説明しています。

本書で説明する手順の大半は、システム・ハードウェアや、ストレージ・デバイスなどの周辺機器の管理に関連します。特に、付属のソフトウェアの使用方法や、デバイスの管理に必要な情報については、ハードウェア・デバイスのオーナーズ・マニュアルを参照してください。

オーナーズ・マニュアルに説明されている、プロセッサのコンソール・コマンドを使用することがあります。デバイス特有の制限についての情報は、『リリース・ノート』で説明されています。次のオンライン・リソースが利用できます。

- Alpha Systems Technology の Web サイトのハードウェア・ドキュメント URL は下記のとおりです。

<http://www.compaq.com/alphaserver/technology/index.html>

- Alpha ファームウェアのダウンロードを含む、ソフトウェアおよびドライバ

URL は下記のとおりです。

<http://www.compaq.com/support/files>

- AlphaServers の一般的リソース

URL は下記のとおりです。

<http://www.compaq.co.jp/products/alphasystems/alphaserver/index.html>

本書の表記法

本書では、以下の表記法を使用します。

MB1, MB2, MB3

MB_N は、項目を選択する際、あるいはアクションを開始するために押す、マウス・ボタンを意味します。

%

\$

パーセント記号は、C シェルのシステム・プロンプトを表します。ドル記号は、Bourne シェル、Korn

シェル，および POSIX シェルの場合の システム・プロンプトを表します。

番号記号は root としてログインした場合のシステム・プロンプトを表します。

file イタリック体 (斜体) は，変数値，プレースホルダ，および関数の引数名を示します。

[|]
{ | }

構文定義では，大カッコはオプションの項目を示し，中カッコは必須項目を示します。大カッコまたは中カッコの中の項目を縦線で区切っている場合は，そこに併記されている項目の中から 1 つの項目を選択することを示します。

...

構文定義では，水平の反復記号は，前の項目を 1 回以上繰り返して使用できることを示します。

:

垂直の反復記号は，実際には存在する例の一部が省略されていることを示します。

cat(1)

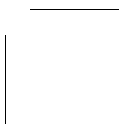
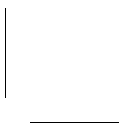
リファレンス・ページの参照には，該当するセクション番号をカッコ内に示します。たとえば，cat(1) は，cat コマンドについての情報が，リファレンス・ページのセクション 1 に記載されていることを示します。

Ctrl/x

この記号は，スラッシュの前に指定されているキーを押しながら，スラッシュの後のキーまたはマウス・ボタンを押すことを示します。例中では，このようなキーの組み合わせは，四角あるいは大カッコで囲まれて示されます(たとえば，`Ctrl/C`)。

`Return`

四角で囲まれたキー名はユーザがそのキーを押すことを示します。





システム管理メソッドおよびユーティリティ

オペレーティング・システムには、さまざまなメソッドおよびユーティリティがあります。これらを使用して、初期設定 (セットアップ) からシステム環境の保守とカスタマイズまでを行うことができます。この章では、以下の説明を行います。

- 管理メソッドおよびユーティリティの概要について説明します (1.1 節)。
- オンライン・ヘルプや Web ベース・ヘルプなど、管理ユーティリティに関連するドキュメントの参照先について説明します (1.2 節)。
- フル・インストールを行った後、最初にシステムに root でログインしたときに自動的に表示される、システム・セットアップ・ユーティリティについて説明します (1.3 節)。
- さまざまな管理メソッドおよびユーティリティについて説明します (1.4 節)。
- CDE (共通デスクトップ環境) から実行する管理ツールについて説明します (1.5 節)。
- SysMan Menu の使用方法について説明します (1.6 節)。
- SysMan Menu のコマンド行インタフェースの使用方法について説明します (1.7 節)。
- SysMan Station の使用方法について説明します (1.8 節)。
- システム状態の表示や、Web ブラウザから SysMan Menu および SysMan Station を起動するのに使用できる、HP Insight Manager について紹介します (1.9 節)。
- SysMan Menu および SysMan Station クライアントを設定して、PC 上のウィンドウからそれらを直接起動できるようにする方法について説明します (1.10 節)。

- シリアル回線コンソールをセットアップして、モデム回線を使用してリモート・システムにアクセスする方法について説明します (1.11 節)。

1.1 SysMan Menu およびその他のユーティリティの概要

SysMan Menu ユーティリティは、ユーザ環境に関係なく使用できます。ユーザ環境には次のようなものがあります。

- CDE などの、X 準拠のユーザ環境
- Windows 98 や Windows NT[®] などの、PC で実行される Microsoft[®] Windows[®] のユーザ環境
- Internet Explorer などの Web ブラウザおよび HP Insight Manager を使用する Web ベース管理
- 端末、または以前のユーザ環境で実行される端末ウィンドウ。この場合は、SysMan ユーティリティを端末 curses モードで使用して表示します。

たとえば、Microsoft Windows NT を実行している PC から、リモート UNIX[®] システム上で管理タスクを実行できます。この管理タスクの実行には、Java アプリケーションとして実行されている SysMan Menu および SysMan Station クライアントを使用します。これらのユーティリティは、どのようなユーザ環境を使用しても外観が一貫しています。

さまざまなメソッドを使用して、同じタスクを実行することができます。ただし、使用する管理ユーティリティおよびその起動方法によって、使用できるオプションが多少異なることがあるので注意してください。たとえば、種々のユーザ環境で実行できるように設計されているため、多くの SysMan Menu ユーティリティには、アイコンなどのグラフィカル要素がありません。また、CDE などのウィンドウ環境で実行するように設計されているため、X11 ベースのユーティリティにはグラフィカル要素を含むものが多く、ドラッグ&ドロップなどのウィンドウ機能がサポートされています。次にメソッドの例を示します。

- ユーザ・アカウントおよびグループを管理するためのアカウント・マネージャ (dxaccounts)
- UNIX カーネルをカスタマイズするためのカーネル・チューナ (dxkerneltuner)

- ローカル・ディレクトリを共有してリモート・シェアをマウントするためのファイル共有 (dxfilesahre)

従来のユーティリティは、下位互換性のために残されていますが、文字セル端末でのみ使用するように設計されています。ただし、これらのユーティリティを SysMan Menu から起動する場合は、サポートされている他のユーザ環境でも動作します。例としては、SysMan Menu に「NIS (Network Information Service) の設定」として表示される、NIS の設定ユーティリティ `nissetup` があります。

X11 準拠のユーティリティと比べて SysMan Menu ユーティリティは、それほど機能的ではなく、グラフィカルでもありませんが、ユーザ環境に依存することなく基本的な管理タスクを実行することができます。またさらに、多岐にわたる管理機能を備えています。SysMan Menu ユーティリティには、次のような制約があります。

- SysMan Menu ユーティリティの表示とレイアウトは、使用しているユーザ環境によって多少異なる場合があります。たとえば、X11 CDE のユーザ環境でシステムのシャットダウンを実行すると、シャットダウンの遅延の選択がスライダ・バーとして表示されます。マウス・ボタンを使ってこのバーを選択してドラッグすることで、遅延時間を長くすることができます。同じユーティリティが文字セル端末で起動された場合は、スライダ・バーの代わりにフィールドが表示され、シャットダウンの遅延時間は数値で入力します。
- さらに、管理ユーティリティは、機能的に違いがあります。SysMan Menu ユーティリティでは、対応するコマンド行 (または X11 準拠) ユーティリティのオプションで使えないものがある場合があります。たとえば、ユーザ・アカウントを管理する場合、`useradd` コマンドを使用すると、新たに作成されるすべてのアカウントが継承する、省略時設定の属性を設定できます。SysMan Menu の「アカウント」ユーティリティでは、これらの属性は設定できません。一般に、SysMan Menu ユーティリティでは最も頻繁に使用されるオプションが提供され、コマンド行インタフェース (CLI) ではすべてのオプションが提供されています。

システム管理者は、システム管理者がいる場所や使用可能なユーザ環境に関係なく、SysMan Menu および SysMan Station で一貫したプレゼンテーション形式の管理ユーティリティを使用することができます。たとえば、ローカル PC からリモートの UNIX システムにログオンして、使い慣れた

同じユーティリティを使用して管理タスクを行うことができます。また、HP Insight Manager を使用すると、Web インタフェースですべてのシステムに接続することができ、そのシステムの状態を表示することができます。また、SysMan Menu および SysMan Station を起動して、リモート・システムでタスクを実行することができます。

1.2 関連ドキュメント

本書では、管理タスクを実行するために管理ユーティリティを使用する方法について説明していますが、管理ユーティリティの起動方法とその全フィールドについては説明していません。その説明には使用例がありますが、すべてのユーザ環境やオプションについては示していません。以降の項では、管理ユーティリティおよびメソッドの起動と使用についてのさらに詳しい説明の参照先を示します。ドキュメントの提供形態は、リファレンス・ページ、オンライン・ヘルプ、および Web ベースのヘルプです。

1.2.1 リファレンス・ページ

各ユーティリティには、その起動方法とオプションを説明しているリファレンス・ページがあります。たとえば、`sysman_cli(8)` では、SysMan Menu データのコマンド行バージョンの起動方法について説明します。

また、リファレンス・ページでは、特定のユーティリティのユーザ環境オプションについても説明します。管理ユーティリティには、さまざまなユーザ環境で起動できるものと、1 つのユーザ環境でのみ起動できるものがあります。

1.2.2 オンライン・ヘルプ

各ユーティリティには、その使用方法とオプションを詳しく説明しているオンライン・ヘルプがあります。また、オンライン・ヘルプは、有効な必須データ、参照情報、および用語の定義について説明します。オンライン・ヘルプへのアクセスは、ユーティリティの最初のウィンドウでボタンをクリックするか、あるいは CDE フロント・パネルのライブラリ・アイコンを起動して CDE ヘルプ・ライブラリから行います。最初に表示されるヘルプ・ボリュームは、システム管理のオンライン・ヘルプです。

グラフィカル・ユーザ環境の中には、オプションとフィールドのコンテキスト依存ヘルプを提供するものがあります。画面上でポインタを移動すると、そのフィールドやオプション・ボタンの簡単な説明がメッセージ・フィールドに表示されます。`curses` ユーザ環境では、矢印キーや Tab キーでフィー

ルドおよびオプションに移動すると、ヘルプ・メッセージが表示されます。詳細は、`curses(3)` リファレンス・ページを参照してください。

コマンド行ユーティリティには、そのコマンドの構文を説明するヘルプがあります。コマンドの構文のヘルプは、`-h` または `-help` フラグを指定するか、引数やパラメータを省略してコマンドを入力して Return キーを押したときに表示されます。

1.2.3 Web ベースのヘルプ

Netscape ビューアを『インストール・ガイド』に説明されているように設定して起動した場合、ホームページの省略時設定は、次のとおりです。

```
file:/user/doc/netcape/Tru64_UNIX.html
```

このページからリンクされている情報は、次のとおりです。

| | |
|--------|----------------------------|
| ドキュメント | オペレーティング・システムのオンライン・ドキュメント |
|--------|----------------------------|

System Management

HP Insight Manager の Web ベース管理 (WBEM) ページである `file:/user/doc/netcape/SYS-MAN/index.html` へのリンク。オペレーティング・システムの管理に関するこのページの情報は、次のとおりです。

- SysMan Menu と SysMan Station の使用
- CDE などの、X 機能のユーザ環境の使用
- Microsoft Windows を実行している PC の使用。
このセクションには、使用している PC にダウンロードしなければならないクライアント・ソフトウェアへのリンクがあります。

SysMan Menu は、Web ブラウザまたは SysMan Station から起動された場合、Web/Java モードで実行されます。SysMan タスクは、Web ブラウザまたは SysMan Station から起動されたか、Web/Java

モードで実行されている SysMan Menu から起動された場合、Web/Java モードで実行されます。

Web/Java モードで実行しているとき (PC からなど) に SysMan Menu またはメニュー・タスクのオンライン・ヘルプを参照するには、接続先のサーバ上で HP Insight Manager デーモンが稼働していなければなりません。このデーモンを起動するには、サーバ上で次のコマンドを root として実行します。

```
# /sbin/init.d/insightd start
```

どのシステムがサーバになっているかは、help コマンドを起動したウィンドウのタイトル・バーを見ると分かります。

SysMan Station でも、オンライン・ヘルプの表示には insightd デーモンが必要です。

World Wide Web

World Wide Web 上の製品情報へのリンク

HP Insight Manager が設定されている場合は、ローカル・ネットワーク・ドメイン内の、HP Insight Manager エージェントを実行しているすべてのシステムの Insight Manager Web エージェントにも接続できます。たとえば、UNIX システムのローカル・ホストに接続するには、Netscape を起動して、その「場所」フィールドに次の URL を入力します。

```
http://<host>:2301
```

ここで、<host> は、bender.fut.ram.ma などのネットワーク上の完全修飾名か、111.22.255.11 などの TCP/IP アドレスです。ポートは常に :2301 です。HP Insight Manager の設定についての詳細は、1.9 節を参照してください。

「HP Insight Manager Agents」を選択した後に「Summary?」をクリックして、HP Insight Manager の Web ベースのユーザ・ガイドにアクセスします。

使用しているユーザ環境によって、HP Insight Manager の使用に制限があります。詳細については、1.9 節を参照してください。

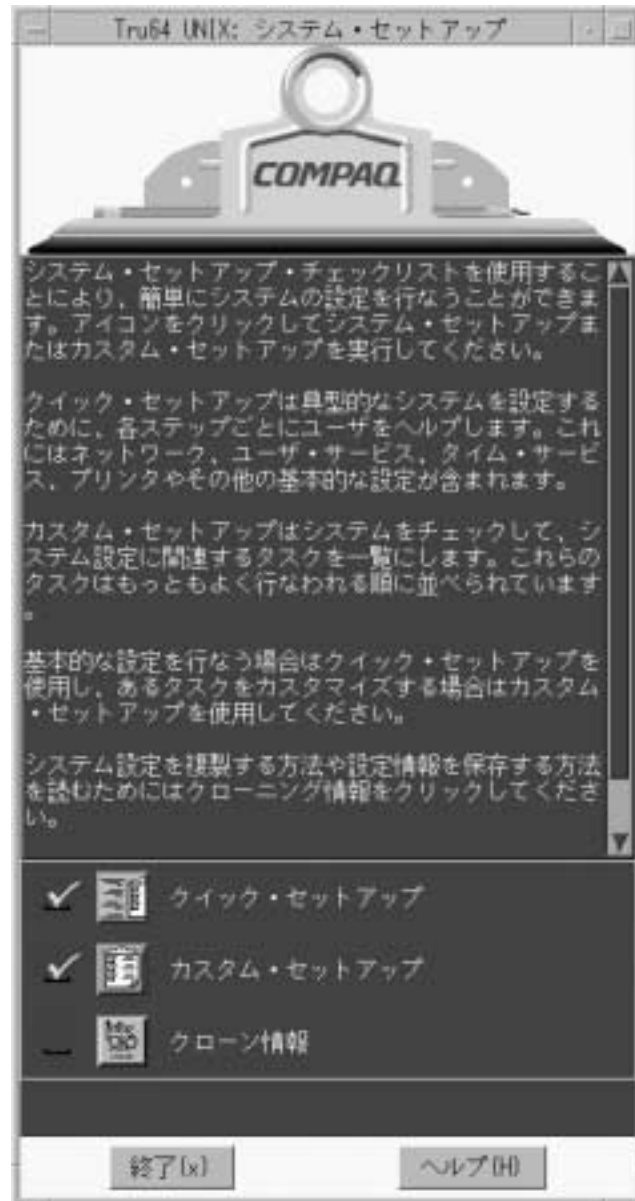
1.3 システムのセットアップ

システムの初期設定 (セットアップ) は通常、インストールの後処理として実行されます。「システム・セットアップ」は、インストール後に初めてスーパーユーザ (root) でログインしたときに自動的に起動されます。インストール時には、この章で説明したユーティリティを使用しているはずですが、システムのメンテナンスおよびカスタム設定で使用するユーティリティと同じユーティリティを初期設定で使います。

システムにグラフィック・ボードが搭載され、省略時設定の CDE などの X11 ユーザ環境で実行している場合は、「システム・セットアップ」ユーティリティ (クリップボードとも呼ばれます) はグラフィカル・ユーザ・インタフェースで動作します。初めてのログインが文字セル端末で行われた場合は、「システム・セットアップ」はテキスト・インタフェースで動作します。

図 1-1 では、グラフィック形式の「システム・セットアップ」を示します。

図 1-1: 「システム・セットアップ」のグラフィカル・ユーザ・インタフェース



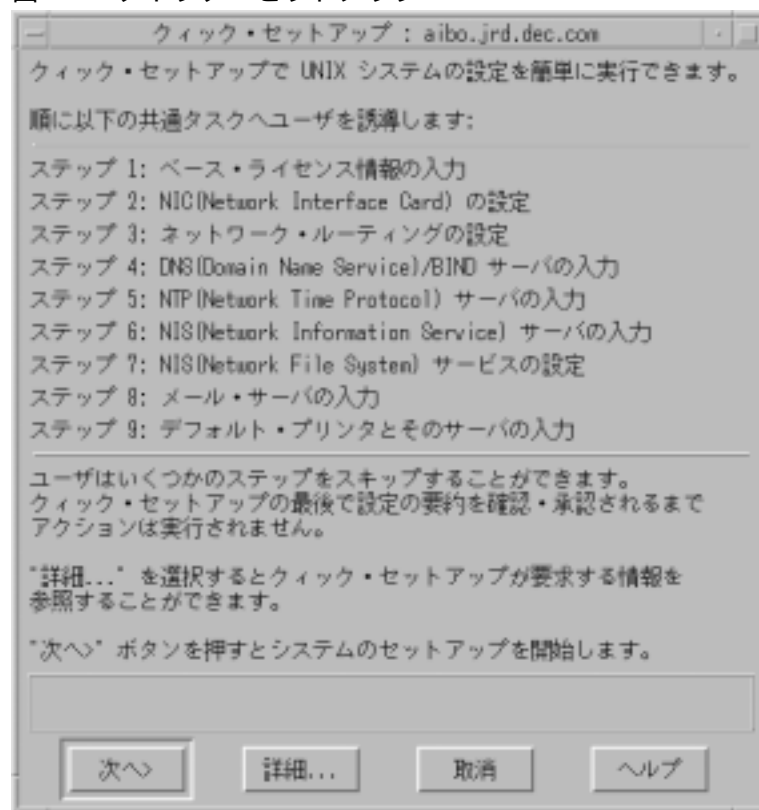
コマンド行で単に `setup` と入力するか、CDE の「アプリケーション・マネージャ - システム管理」フォルダで「システム・セットアップ」アイコンをクリックすると、いつでも「システム・セットアップ」を起動して既存のシステム構成を変更することができます。オプションは、次のとおりです。

1-8 システム管理メソッドおよびユーティリティ

クイック・セットアップ

ネットワーク、メールおよびプリンタなどの、システム・サービスの基本構成を設定することができます。このオプションは、拡張設定オプションを後回しにして、システムをすばやく設定して稼働させたいときに便利です。図 1-2 では、「クイック・セットアップ」の初期ウィンドウを示します。

図 1-2: クイック・セットアップ

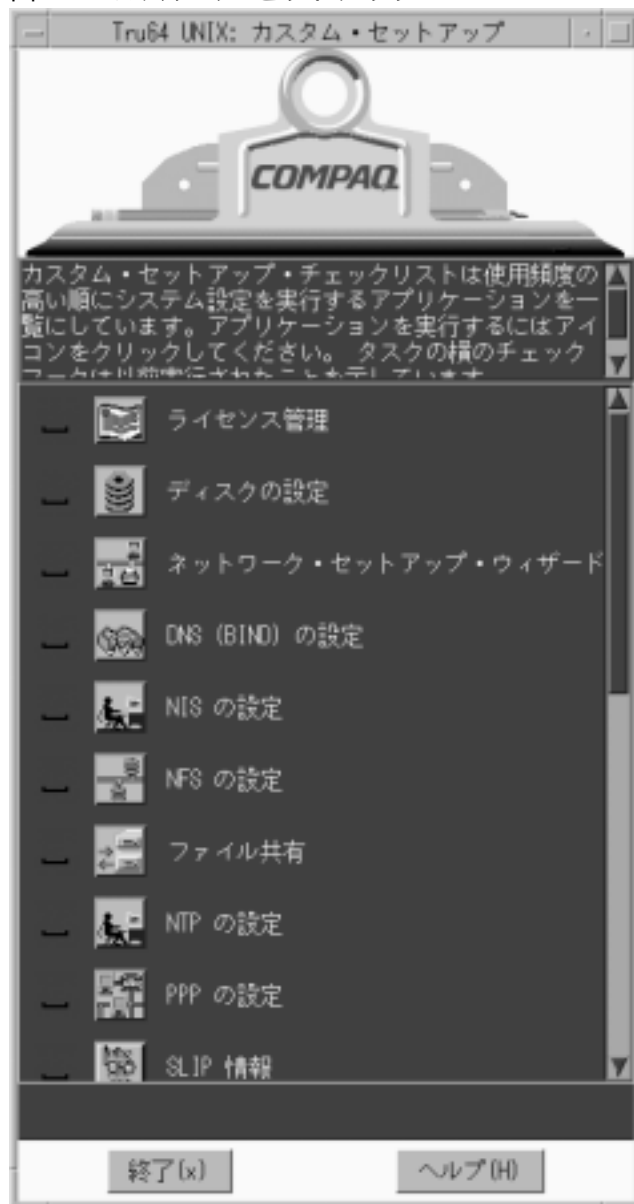


カスタム・セットアップ

さまざまなシステム設定ユーティリティを実行して、「クイック・セットアップ」のタスクをすべて実行することができます。また、ディスクのカスタム設定または二地点間 (point-to-point) プロトコルの設定などの追加のセットアップ・オプションを実行します。

図 1-3 では、「カスタム・セットアップ」のグラフィカル・インタフェースの一部を示します。

図 1-3: カスタム・セットアップ



クローニング・セッ
トアップ情報

このオプションは、システム設定のクローニングを
行う SysMan Menu オプションについての情報へ
のリンクを提供します。これにより、システム設
定を他のシステムに適用することができます。詳
しい説明は、『インストレーション・ガイド』上

級ユーザ編』および `sysman_clone(8)` リファレンス・ページを参照してください。

詳細については、1.5.2 項を参照してください。

1.4 管理メソッド

本書で説明しているタスクのほとんどは、以下の 1 つまたは複数のメソッドを使用して行います。種々のユーザ環境で利用できるように、SysMan はシステム管理タスクの推奨メソッドになっています。

- SysMan Menu

SysMan Menu は、利用できるシステム管理ユーティリティの大半を 1 つのメニューにまとめたものです。SysMan Menu を使用すると、次のような環境からユーティリティを実行できます。

- ローカルまたはリモートの文字セル端末
- CDE などの、X11 準拠ウィンドウ環境
- PC 上の Microsoft Windows
- Web ブラウザと HP Insight Manager

詳細は 1.6 節を参照してください。

- SysMan Station

SysMan Station は、システムまたはクラスタのグラフィック表示で、CPU ダウンからディスクなどのシステムの個々の構成要素までを監視することができます。さらに、ファイル・システムや AdvFS ドメインなどの論理的なグループを表示または監視したり、表示内容をカスタマイズすることができます。システムの構成要素を表示する場合は、その構成要素のプロパティについての詳細情報を取得したり、その構成要素に対して管理タスクを行うユーティリティを起動したりできます。SysMan Menu とは異なり、SysMan Station にはグラフィック機能が必須で、文字セル端末や `curses` ユーザ環境では実行できません。

詳細は、1.8 節を参照してください。

- CDE の「アプリケーション・マネージャ - システム管理」のグラフィカル・ユーザ・インタフェース

X11 準拠のグラフィカル・ユーザ・インタフェース (GUI) のセットであり、CDE または他の X11 準拠のウィンドウ環境で実行されます。GUI

を使用するには、グラフィックス(ウィンドウ機能)を備えた端末またはワークステーションに、ウィンドウ機能ソフトウェア・サブセットがインストールされている必要があります。これらのグラフィカル・ユーティリティでは、ウィンドウ環境の機能がサポートされています。たとえば、`dxaccounts` でユーザ・アカウントの複製バージョンを作成するのにカット・アンド・ペースト機能を使用できます。

詳細は、1.5 節を参照してください。

- コマンド行スクリプト

古い管理ユーティリティはほとんどの場合、互換性維持のために残されています。いくつかのコマンド行ユーティリティは、新しいコマンド行オプションに移行されています。たとえば、`adduser` スクリプトはまだ使用できますが、次のユーティリティがこのスクリプトに取って代わります。

- SysMan Menu の「アカウント」ユーティリティ。このユーティリティを使用すると、ローカル環境および NIS 環境でユーザとグループを管理できます。
- `useradd` コマンド行ユーティリティ。このユーティリティは、文字セル端末から実行します。
- 「アカウント・マネージャ」グラフィカル・ユーザ・インタフェース。CDE 環境の「アプリケーション・マネージャ - 日常管理」から利用可能です。または端末ウィンドウから `dxaccounts` を起動します。このインタフェースは、他の X 準拠のウィンドウ環境で実行します。
- SysMan Menu の「アカウント」オプション。CDE 環境の「アプリケーション・マネージャ - システム管理」から利用可能です。または端末ウィンドウから `sysman` を起動します。

システム管理プロセスを、古いコマンド行スクリプトから適切な SysMan Menu メソッドに移行する必要があります。これらのコマンド行ユーティリティは、`OSFRETIREDxxx` サブセット(オプション)に移行されています。使用されなくなったコマンド・サブセットのインストールについての詳しい説明は、『インストール・ガイド』を参照してください。

- シリアル回線コンソール

ネットワークを利用した管理メソッドに加えて、シリアル回線コンソールにはダイアルアップ機能があります。これにより、モデム経由でのリモート・システムとの接続が可能です。詳細は、1.11 節を参照してください。

- システム・ファイルの編集によるファイルの手動変更 (推奨していません)

従来、経験を積んだ UNIX のシステム管理者は、個々のシェル・コマンド、スクリプト、およびユーティリティの組合わせを使用したり、簡単にシステム・ファイルを編集していました。本書のほとんどの節では、管理タスクを実行するときに更新または変更するさまざまなシステム・ファイルについて説明します。このような場合に、編集によってシステム・ファイルを変更したいかもしれませんが、システム・ユーティリティを使用することで、`/etc/sysconfigtab` などのシステム・ファイルの完全性と整合性を維持します。システム・ファイルの変更には、適切なユーティリティを使用することをお勧めします。適切なユーティリティを使用することで、システム・ファイルの構造を保護します。

重要な考慮事項は、次のとおりです。

CDSL (コンテキスト
依存シンボリック・
リンク)

多くのシステム・ファイルが、特別なシンボリック・リンクになっています。これはクラスタ処理を容易にするために作成されたものです。これらのリンクが破壊された場合は、そのリンクを再度作成しない限り、システムはクラスタに結合できません。詳細は、第 6 章および `hier(5)` リファレンス・ページを参照してください。

バイナリ・データ
ベース、設定定義

多くのシステム・コンポーネントは、データをテキスト・ファイルとバイナリ・ファイルの両方に書き込み、その管理ユーティリティが、バイナリを再作成することがよくあります。他のシステム情報は、多くの場合保存され、システムを更新するときに、保存されたシステム情報を復元して再使用することで、システム管理の時間と労力を節約します。

クラスタの内部サ
ポート

個々のシステムはクラスタとして結合することが可能です。多くの UNIX システム・ファイル

が最近変更され、クラスタの内部サポートが提供されています。たとえば、`rc.config` ファイルには、実行時構成変数を格納する 2 つのファイル、`rc.config.common` と `rc.config.site` が関連しています。`rcmgr` ユーティリティを使用することで、これらのファイルの完全性と整合性が確保できます。

アップデート・インストール・カスタマイズ・ファイルの保存

アップデート・インストール中、インストール処理は変更された情報を既存のシステム・ファイルにマージします。`.new..*` および `.proto..*` ファイルは、この処理において重要になります。詳しい説明は、『インストール・ガイド — 上級ユーザ編』を参照してください。

1.5 CDE での管理ユーティリティ

CDE (共通デスクトップ環境) は、省略時設定の X11 ウィンドウのユーザ環境です。ただし、以降の項で説明しているユーティリティは、他の X11 準拠ユーザ環境でも実行できます。フル・インストールが完了すると、「システム・セットアップ」のグラフィカル・ユーザ・インタフェースが表示され、初めて使用する場合のシステムの設定手順を示します。システムの管理およびカスタマイズで今後使用するグラフィカル・ユーザ・インタフェース (GUI) と同じインタフェースを「システム・セットアップ」から起動します。「システム・セットアップ」の詳細は、1.5.2 項で説明しています。

CDE から起動する管理ユーティリティの多くが、SysMan Menu のタスク・オプションを開始します。ただし、ユーティリティの中には、SysMan Menu に類似したオプションを持たないものや、CDE でのみ使用できる機能を備えたグラフィカル・ユーティリティがあります。これらのユーティリティの例を次に示します。

- CDE Setup は、CDE 環境の設定に使用します。
- Disk Configuration (`diskconfig`) は、ディスク・パーティションの構成に使用するアプリケーションです。

- アーカイバ (dxarchiver) は、tar、pax、またはcpio アーカイブの作成に使用するアプリケーションです。ドラッグ・アンド・ドロップ操作を使用して、フォルダを簡単にアーカイブ・ファイルに追加できます。

CDE では、これらの GUI は「アプリケーション・マネージャ」の中に置かれています。アプリケーション・マネージャは、図 1-4 に示したような、CDE フロント・パネルのツール・ドロワのオプションです。ツール・ドロワの隣のアイコンは、root でログインした場合にのみ CDE フロント・パネルに表示され、SysMan Station を起動するために使用します。このアイコンについては、1.8 節 で説明しています。

図 1-4: CDE のツール・ドロワおよび SysMan Station アイコン



CDE 以外の、X11 準拠のユーザ環境を使用する場合は、次の例に示すようにコマンド行で個々に GUI を起動してください。

```
# /usr/sbin/X11/dxaccounts
```

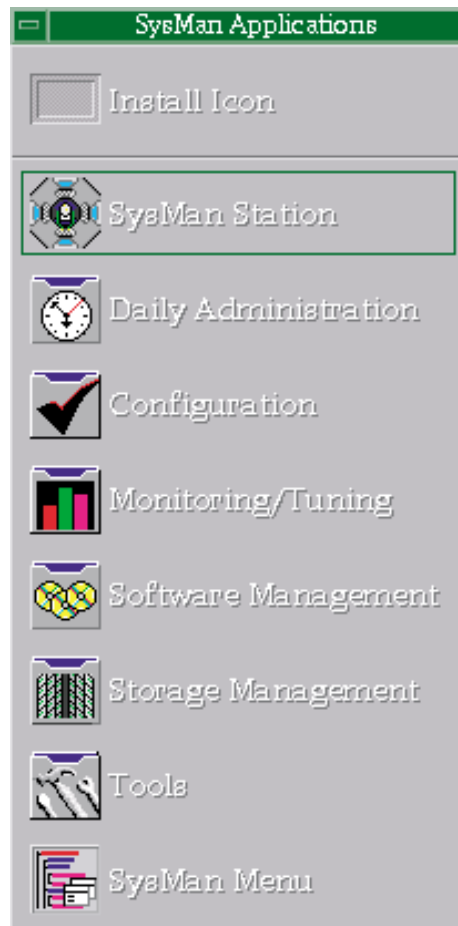
```
# /usr/sbin/X11/dxarchiver
```

1.5.1 CDE での SysMan へのアクセス

CDE では、SysMan Menu のユーティリティが「アプリケーション・マネージャ」フォルダ内で使用できます。このフォルダには、次のようにアクセスします。

1. CDE フロント・パネルで、「SysMan Applications」パネルの矢印を選択します。この矢印は、図 1-4 で示した SysMan Station のアイコンの上に表示されています。この矢印を選択すると、図 1-5 のようなパネルが表示されます。

図 1-5: 「SysMan Applications」パネル



このパネルから次のアイコンのいずれかを選択して、ユーティリティを起動するか、または複数の管理ユーティリティが収められているフォルダを開きます。

- 「SysMan Station」を起動します。これについては 1.8 節 に説明されています。root で CDE にログインすると、このアイコンはフロント・パネル上に 図 1-4 のように表示されます。
- 「Configuration」などのフォルダ・アイコンを選択して、アプリケーション・フォルダを開きます。このフォルダには、付録 A で説明するユーティリティが入っています。
- 「SysMan Menu」を起動します。

2. CDE フロント・パネルから、図 1-4 で示したツール・ドロワのアイコンを選択します。オープンされたトップレベルのフォルダで、「システム管理」グループをダブルクリックすることにより、「システム・セットアップ」、「SysMan の紹介」のオンライン・ヘルプ・ボリューム、および 5 つのユーティリティ・グループにアクセスすることができます。詳細は、1.5.2 項を参照してください。

ユーティリティを実行せずに、SysMan Menu ユーティリティのオンライン・ヘルプへアクセスできます。CDE フロント・パネルから「ヘルプ・マネージャ」アイコンを選択して、オンライン・ヘルプのブラウザを表示します。このブラウザには、CDE、CDE デスクトップ、システム管理のヘルプ・ボリュームが含まれています。「デスクトップアプリケーション」フォルダにある「アクション作成」ユーティリティで、CDE ワークスペースをカスタマイズすることもできます。カスタマイズされたアイコンを使用して、ワークスペースから直接 SysMan アプリケーションを起動することもできます。詳細については、『CDE ガイドブック』を参照してください。

別の X-Window 環境では、コマンド行から SysMan ユーティリティを起動できます。SysMan ユーティリティの一覧については、`sysman_intro(8)` を参照してください。このリファレンス・ページでは、CDE 以外のグラフィカル環境でオンライン・ヘルプのブラウザを起動する方法についても説明しています。SysMan Station アイコンは、root ユーザでログインした場合は CDE フロント・パネルにも表示されます。

詳細は、以下のリファレンス・ページを参照してください。

| | |
|--------------------------------|---|
| <code>sysman(8)</code> | SysMan Menu についてと、さまざまな環境でそれを起動する方法について説明します。1.6 節を参照してください。 |
| <code>sysman_station(8)</code> | SysMan Station についてと、その起動方法について説明します。1.8 節を参照してください。 |
| <code>sysman_cli(8)</code> | SysMan Menu のコマンド行オプションについて説明し、そのコマンド・オプションを定義します。1.7 節を参照してください。 |

1.5.2 システム・セットアップ

「システム・セットアップ」は、初めて使用する場合のシステムの設定の手順を案内します。「システム・セットアップ」は、各構成アプリケーションのアイコンを含むクリップボードのグラフィカル表示です。フル・インストールを行った後、root ユーザで最初にログインしたときに「システム・セットアップ」が自動的に起動され、システム設定タスクの実行を促します。初期ウィンドウには、クイック・セットアップとカスタム・セットアップの2つのオプションがあります。

1.5.2.1 クイック・セットアップ

このオプションは、一般的なシステム設定の手順ガイド (ウィザード) を提供します。「クイック・セットアップ」を使用して、基本設定を行います。システムによっては、この基本設定だけの場合もあります。「カスタム・セットアップ」を使用して、拡張設定またはシステム固有の設定を後で行うことができます。

「クイック・セットアップ」のウィザードでは、次のようなタスクの手順を案内します。

- ソフトウェア・ライセンス (PAK) の入力
 - ネットワーク・インタフェース・カード (NIC) の設定
 - 静的ネットワーク・ルーティングの設定
 - 次のようなネットワーク・サービスおよびネーム・サーバの指定
 - ドメイン・ネーム・サービス (DNS, 以前は BIND)
 - ネットワーク・タイム・プロトコル (NTP)
 - ネットワーク情報サービス (NIS, 以前は YP すなわちイエローページ)
 - ネットワーク・ファイル・システム (NFS)
 - 電子メール・サーバ
 - 省略時設定のローカルまたはリモートの、プリンタとサーバの設定
- 必要としないオプションはスキップできます。詳細は、この節の後半で説明しています。

1.5.2.2 カスタム・セットアップ

このオプションは、個々の構成アプリケーションのアイコンを含む、「システム・セットアップ」の1つのバージョンを起動します。システムをサーバとして設定するなど、システム固有の設定またはカスタム設定に必要なオプションだけを選択できます。すべての構成アプリケーションがすべてのシステムで利用できるわけではありません。 `/etc/checklist.desc` ファイルには、構成アプリケーションの一覧が含まれています。

アイコンを選択すると、適切な SysMan Menu ユーティリティ、X11 ベースの GUI、または文字セル・スクリプトが起動されます。使用可能なユーティリティは、以下のとおりです。

License Manager

License Manager (`dxlicenses`) を起動します。License Manager を使用すると、オペレーティング・システムやレイヤード・ソフトウェア・アプリケーションの製品登録キー (PAK またはライセンス) を登録できます。ソフトウェア・ライセンスのハードコピーは、製品の配布メディアに添付されています。詳細は、`dxlicenses(8)`、`lmf(8)`、および『*Software License Management*』を参照してください。

Disk Configuration

Disk Configuration (`diskconfig`) を起動します。Disk Configuration を使用すると、システム上のディスク・デバイスの設定と管理ができます。詳細は、`diskconfig(8)`、`disklabel(8)`、および『*ハードウェア管理ガイド*』を参照してください。

Network Configuration Step By Step

SysMan Menu Network Setup ウィザードを起動します。このウィザードを使用すると、システムのネットワーク・コンポーネントを設定し管理する処理を、手順を追って実行することができます。詳細は、`sysman(8)`、`network_manual_setup(7)`、および『*ネットワーク管理ガイド：接続編*』を参照してください。表示される設定オプションは、以下のとおりです。

- ネットワーク・インタフェース・カード (NIC) の設定
- 静的ルートの設定と `/etc/routes` ファイルの設定

- ルーティング・サービス (gated , routed , あるいは IP ルータ) の設定
- リモート who サービス (rwhod) のセットアップ
- DHCP サーバ (joind) のセットアップ
- /etc/hosts.equiv ファイルの内容の指定
- /etc/networks ファイルの内容の指定

Network Setup ウィザードで表示されるオプションの他に、サイト固有のネットワーク要件に応じて、NTP のようなその他のオプションをセットアップする必要があります。詳細は、『ネットワーク管理ガイド：接続編』と『ネットワーク管理ガイド：サービス編』を参照してください。

DNS (BIND) Configuration

「Configure system as a DNS client」を選択すると、ドメイン・ネーム・サーバ (DNS) を設定する DNS Client Configuration ユーティリティを起動できます。詳細は、bindconfig(8)、network_manual_setup(7)、『ネットワーク管理ガイド：接続編』、および『ネットワーク管理ガイド：サービス編』を参照してください。

NIS Configuration

ネットワーク情報サービス NIS を設定する nissetup スクリプトを起動できます。これは、ypsetup と呼ばれます。詳細は、nissetup(8)、network_manual_setup(7)、『ネットワーク管理ガイド：接続編』、および『ネットワーク管理ガイド：サービス編』を参照してください。

NFS Configuration

SysMan Menu が起動され、システムの NFS コンポーネントの設定と管理を行う Network File Systems (NFS) ユーティリティが表示されます。詳細は、sysman(8)、nfs_intro(4)、『ネットワーク管理ガイド：接続編』、および『ネットワーク管理ガイド：サービス編』を参照してください。

File Sharing

ファイル・システムへのアクセスと共用を可能にする dxfileshare オプションを起動します。詳細は、dxfileshare(8)、『ネットワー

ク管理ガイド：接続編』，および『ネットワーク管理ガイド：サービス編』を参照してください。ファイル・システムについての詳細は，第 6 章を参照してください。

NTP Configuration

ネットワーク時刻を設定する，SysMan Menu の Network Time Protocol Configuration オプションを起動できます。詳細は，sysman(8)，ntp(1)，ntp_intro(7)，『ネットワーク管理ガイド：接続編』，および『ネットワーク管理ガイド：サービス編』を参照してください。

PPP Configuration

SysMan Menu が起動され，PPP (ポイント・ツー・ポイント・プロトコル) のオプションとシークレット・ファイルの設定を行う Serial Line Networking オプションが表示されます。詳細は，sysman(8)，ppp_manual_setup(7)，pppd(8)，『ネットワーク管理ガイド：接続編』，および『ネットワーク管理ガイド：サービス編』を参照してください。

SLIP Configuration

詳細は，PPP のエントリと startslip(8) を参照してください。

Account Manager

クライアント PC の UNIX および Windows NT ドメインのユーザ・アカウントの作成と，ユーザ・グループの管理を行う，Account Manager (dxaccounts) の GUI を起動できます。詳細は，dxaccounts(8)，adduser(8)，および 第 7 章を参照してください。

Mail Configuration

電子メールを送受信するシステムを設定する Mail Configuration ユーティリティを起動できます。詳細は，sysman(8)，mail_intro(7)，mailconfig(8)，『ネットワーク管理ガイド：接続編』，および『ネットワーク管理ガイド：サービス編』を参照してください。

LAT Configuration

LAT (ローカル・エリア・トランスポート) サービスを設定する latsetup スクリプトを起動できます。詳細は，latsetup(8)，

lat_intro(7), 『ネットワーク管理ガイド：接続編』, および『ネットワーク管理ガイド：サービス編』を参照してください。

UUCP Configuration

UNIX 間接続とモデムを設定する接続設定スクリプト `uucpsetup` を起動できます。詳細は, `uucpsetup(8)`, `uucp_intro(7)`, 『ネットワーク管理ガイド：接続編』, および『ネットワーク管理ガイド：サービス編』を参照してください。

Printer Configuration

ローカル・プリンタとリモート・プリンタを設定する SysMan Menu の `Configure line printers` オプションを起動できます。詳細は, `sysman(8)`, `printconfig(8)`, `lprsetup(8)`, および第 8 章を参照してください。

Security Configuration

ベース・セキュリティと拡張セキュリティを設定する, SysMan Menu の `Security ユーティリティ` を起動できます。詳細は, `secconfig(8)` と『セキュリティ管理ガイド』を参照してください。

Audit Configuration

監査サブシステムを設定する, SysMan Menu の `Security ユーティリティ` を起動できます。詳細は, `auditconfig(8)` と『セキュリティ管理ガイド』を参照してください。

DOP (Division of Privileges)

通常は `root` ユーザだけが実行できるユーティリティを, 特権のないユーザに特権を割り当てて実行できるようにさせるための, SysMan Menu のオプション `DOP (Division of Privileges)` を起動できます。詳細は, `dop(8)` と `sysman(8)` を参照してください。

Prestoserve I/O Acceleration Configuration

`Prestoserve` を設定する `prestosetup` スクリプトを起動できます。詳細は, `presto(8)`, `presto_setup(8)`, および『*Guide to Prestoserve*』を参照してください。

GUI Selection

CDE のウィンドウ・マネージャ `xwm` を設定できます。

ATM

ATM (非同期転送モード) を設定するスクリプトを起動できます。

HP Insight Manager

HP Insight Manager を有効にして設定するユーティリティを起動できます。

「システム・セットアップ」が提供しているオプションをすべて使用する必要はありません。オプションを後で指定することができます。オプションの設定の延期を選択して「システム・セットアップ」を終了した場合は、「システム・セットアップ」を「アプリケーション・マネージャ－システム管理」フォルダから、または SysMan Menu から、あるいは次のコマンド行で、手動で起動する必要があります。

```
# /usr/sbin/sysman
# /usr/sbin/checklist
# /usr/sbin/setup
```

1.6 SysMan Menu

SysMan では、ほとんどのすべてのシステム管理ユーティリティを統合し、それらを種々のユーザ環境で実行できるようにしています。これらのユーティリティには、SysMan Menu という、階層構造のタスク指向型メニュー・インタフェースでアクセスできます。

SysMan Menu のすべてのタスクは、X11 機能を持つ画面、Windows NT Version 4.0 などの Microsoft Windows を実行している PC、または文字セル端末から実行できます。SysMan Menu を開始する方法には、次のようないくつかの方法があります。

CDE デスクトップから SysMan Menu を開始する場合

- root でログインし、CDE フロント・パネルの「SysMan Applications」パネルで SysMan Menu アイコンを選択します。
- 「アプリケーション・マネージャ」の「システム管理」グループから SysMan Menu アイコンを選択します。

- 端末ウィンドウのコマンド・プロンプトから SysMan Menu を開始するには、次のコマンドを入力します。

```
# /usr/sbin/sysman
```

- SysMan Station から SysMan Menu を開始するには、ビュー・ウィンドウでシステム・アイコンを選択した後、SysMan Station の [Tools] メニューで [SysMan_Menu] を選択します。

メニューでの名前またはそのアクセラレータの名前を使用して、コマンド行から直接特定のタスクを実行することができます。アクセラレータは、SysMan Menu の各オプションに対する一意のキーワードです。たとえば、[Configure Division of Privileges (DOP)] メニュー・オプションを起動するタスクを実行するには、そのアクセラレータである `dopconfig` を使用して、システム・プロンプトで次のようにコマンドを入力します。

```
# /usr/sbin/sysman dopconfig
```

次のコマンドを使用すると、使用可能なタスクとそのアクセラレータの完全な一覧が表示されます。

```
# /usr/sbin/sysman -list
```

SysMan Menu では、オプションの一覧が階層構造 (ツリー) で表示されます。各オプションは、ツリーのブランチとして表示され、さらにブランチ、またはタスクで終了するサブオプションが続きます。サブオプションがある場合、各オプションを縮小したり展開したりできます。サブオプションがあるかどうかは、各メニュー項目の前に表示される文字で識別できます。+ 記号は、さらにメニュー項目があることを示し、- 記号は、ブランチが完全に展開されていることを示します。

ブランチの終点のタスクの前には縦線 (|) が表示されます。この記号は、そのブランチがそれ以上展開できないことを示しています。この場合、そのタスクを選択するだけで、管理ユーティリティを起動することができます。

図 1-6 は、CDE ユーザ環境で起動された SysMan Menu を示しています。ただし、このメニューの内容は、使用するシステムによって異なります。

図 1-6: SysMan Menu



図 1-6 では、「ソフトウェア」ブランチ(ラベル 1) が完全に展開され、「インストール」ブランチと「ライセンス・データの登録」タスクが表示されています。「インストール」ブランチ(ラベル 2) には、「ソフトウェアのインストール」や「インストールされているソフトウェアの一覧」など、いくつかのタスクが含まれます。タスクを選択すると、関連するユーティリティが起動されます。

メニュー項目への移動方法や選択方法、あるいはタスクの起動方法は、`curses` 端末とウィンドウ環境のどちらかなど、使用しているユーザ環境によって異なります。端末を使用している場合は、矢印キーまたは Tab キーを使ってメニュー項目の中を移動します。移動した場所のオプションやボタンは強調表示されます。Enter キーを使用して項目を選択し、ブランチを拡張したり、関連するユーティリティを起動するタスクを選択します。ウィンドウ環境でマウスを使用している場合は、ブランチまたはタスクにポインタを移動して MB1 をダブルクリックすることで、ブランチを拡張したり、タスクを選択して関連するユーティリティを起動します。ユーティリティのナビ

ゲーシヨンの手順についての詳細は、オンライン・ヘルプを参照してください。SysMan Menu に表示されるオプションは、次のとおりです。

| | |
|----------|---|
| 選択 | 強調表示されている項目を選択します。 ブランチを選択すると、そのブランチが展開または縮小されます。 タスクを選択すると、関連するユーティリティが起動されます。 |
| 検索... | 検索ウィンドウを起動します。 このウィンドウを使用して、キーワードを検索して関連するタスクを見つけることができます。 |
| 項目別ヘルプ | ブランチまたはタスクにコンテキスト依存ヘルプを表示します。 |
| 終了 | SysMan Menu ウィンドウをクローズします。 |
| オプション... | アクセラレータの表示など、SysMan Menu の画面構成のためのオプションを示します。 |
| ヘルプ | SysMan Menu の一般的なヘルプを表示します。 |

コンテキスト依存ヘルプは、2 行あるボタンの行間にあるペインにも表示されます。 このオンライン・ヘルプでは、マウス・ポインタまたは Tab キーを使用してウィンドウ内の項目に移動したときに、その項目の説明が表示されます。 タスクを選択すると、現在のユーザ環境に最も適したフォーマットで、関連するユーティリティが起動されます。 たとえば、X11 準拠ウィンドウ環境、または文字セル端末の `curses` フォーマットなどです。

詳細については、`sysman(8)` およびオンライン・ヘルプを参照してください。 関連ユーティリティについての詳しい説明は、付録 A を参照してください。

1.7 SysMan コマンド行の使用

`sysman -cli` ユーティリティは、SysMan Menu のコマンド行インタフェース版であり、SysMan Menu タスクの実行、SysMan データの表示、あるいはスクリプトの作成による設定タスクのカスタマイズを行うことができます。 SysMan Menu タスクを使用してネットワークを設定する場合など、シ

システムの別の部分をセットアップするときには、`/etc/rc.config.common` や `/etc/hosts` などのシステム構成ファイルを操作します。 `sysman -cli` ユーティリティを使用すると、コマンド行またはシェル・スクリプトで直接これらのファイルのエントリを表示し操作することができます。

`sysman -cli` オプションを使用するには、`root` の特権が必要です。ただし、特権を持たないユーザでも、このオプションを使用してシステムのセットアップ・データを表示することはできます。特権を持たないユーザに SysMan タスク実行の権限を与えることができる「DOP (division of privileges)」ユーティリティの使用についての詳しい説明は、`dop(8)` を参照してください。

この節では、`sysman -cli` ユーティリティの多様な機能について簡単に説明します。オプションとフラグの完全なリストについては、`sysman_cli(8)` を参照してください。一連のシェル・スクリプトの例が `/usr/examples/systems_management/sysman_cli` にあります。コマンド行のいくつかの例を次に示します。

`sysman -cli` コマンドを使用して、SysMan Menu で管理可能なコンポーネントをすべて表示できます。たとえば、次のコマンドを使用すると、SysMan Menu の階層のメインのコンポーネントの一覧を表示できます。

```
# sysman -cli -list components
```

```
Component(s):
  account_management
  atm
  auditconfig
  bindconfig
  bttape
  ciconfig
  clsschl
  doprc
.
.
.
networkedSystems
.
.
.
```

次のコマンドは、`networkedSystems` コンポーネントに含まれるグループを表示します。

```
# sysman -cli -list group -comp networkedSystems
Component: networkedSystems
Group(s):
```

```
hostEquivalencies
hostEquivFileText
hostFileText
hostMappings
joinMappingService
componentid
digitalmanagementmodes
```

次のコマンドは、networkedSystems コンポーネントの hostMappings グループ内のデータの現在の値を表示します。このデータは、/etc/hosts ファイルの内容です。

```
# sysman -cli -list values -group hostMappings /
-comp networkedSystems
Component: networkedSystems
Group: hostMappings
{} {} 127.0.0.1 localhost
argnot {local host} 16.140.112.139 argnot.xxx.yyy.com
jason server 16.140.112.3 jason.xxx.yyy.com
fleece {backup server} 16.140.112.28 fleece.xxx.yyy.com
{} {} 150.2.3.4 newshst.pubs.com
```

SysMan Menu のすべてのオプションに対して、sysman -cli コマンドは、ユーティリティを起動することなく、システム・データを表示したり、操作することができます。たとえば、次のコマンドは /etc/hosts ファイルからホストのエントリを削除する方法を示しています。

```
# sysman -cli -delete row -group hostMappings /
-comp networkedSystems

Please enter key 1 [systemName]: newshst.pubs.com
Please enter key 2 [networkAddress]: 150.2.3.4
```

キー・データの入力を促すプロンプトが表示されています。これによってユーティリティは /etc/hosts ファイル内のエントリを正確に識別します。SysMan Menu オプションはテーブルに格納されているデータを扱うこともあるので、テーブル内の削除または変更する行を正しく識別する必要があります。各行には、一意の識別子があります。この識別子はキーと呼ばれ、sysman -cli のコマンド・オプションで指定する必要があります。キーを指定しなかった場合、キーの入力を促すプロンプトが表示されます。次のコマンドは、特定のテーブルのキーを調べる方法を示しています。

```
# sysman -cli -list keys -group hostMappings -comp /
networkedSystems
```

```
Component: networkedSystems
```

Group: hostMappings

Keys: systemName,networkAddress

sysman -cli コマンドを使用して、SysMan Menu で更新されるシステム・データ・ファイルのユーザ・データ・エントリを、追加したり削除することもできます。たとえば、次のコマンドでは、対話式にメール・ユーザを追加します。

```
# sysman -cli -add row -comp mailusradm -group mailusers
```

```
Attribute Name: user_name (key attribute)
Attribute Description: user name
Attribute Type: STRING(8), Default Value:
Enter Attribute Value: davisB
```

```
Attribute Name: nis
Attribute Description: NIS User
Attribute Type: INTEGER, Default Value: 0
Enter Attribute Value ( to use default): 1
```

```
Attribute Name: mail_type (key attribute)
Attribute Description: mail user type
Attribute Type: INTEGER ENUM /
{ 0=Local/pop, 1=Secure Pop, 2=IMAP, 3=Secure IMAP }, /
Default Value: 0
Enter Attribute Value ( to use default): 2
```

```
Attribute Name: acl
Attribute Description: acl list
Attribute Type: INTEGER ENUM /
{ 0=all, 1=read, 2=post, 3=append }, Default Value: 0
Enter Attribute Value ( to use default): 0
```

```
Attribute Name: quota
Attribute Description: user name
Attribute Type: STRING(8), Default Value:
Enter Attribute Value:
```

```
Attribute Name: passwd
Attribute Description: password
Attribute Type: STRING(20), Default Value:
Enter Attribute Value: change_me
```

```
Attribute Name: orig_mailtype
Attribute Description: original mail user type
Attribute Type: INTEGER ENUM /
{ 0=Local/pop, 1=Secure Pop, 2=IMAP, 3=Secure IMAP }, /
Default Value: 0
Enter Attribute Value ( to use default):
```

#:

次のようにすべての属性値を指定して、1 行のコマンドとして入力することもできます。

```
# sysman -cli -add row -comp mailusradm -group mailusers /  
-data "{davisB} {1} {2} {0} {0} {pls_chg} {1}"
```

1.8 SysMan Station

SysMan Station を使用すると、システム、システムのグループ、またはクラスタ全体を監視し、システム・リソースを管理することができます。また、SysMan Menu を起動したり、[Tools] メニューからユーティリティを直接起動したりすることもできます。さらに、システム・コンポーネントを示すアイコンを選択した後に MB3 を押して、選択したデバイスに適用するオプションのメニューを表示できます。SysMan Menu とは異なり、SysMan Station は高度なグラフィカル・インタフェースであり、CDE または Microsoft Windows などのウィンドウ・ユーザ環境でだけ実行できます。

注意

SysMan Station のサーバとクライアントの互換性のあるバージョン間でのみ接続できます。互換性のないサーバに接続しようとする、次のようなエラー・メッセージまたはダイアログが表示されます。

```
System Management Server on host host name running version N,  
This client running incompatible version N
```

サーバからクライアントのソフトウェアをダウンロードして、適切なバージョンにアップグレードしてください。

この節では、カスタマイズされたビューを含む、SysMan Station の主要な機能について簡単に説明します。詳細については、SysMan Station のオンライン・ヘルプを参照してください。

CDE で SysMan Station を開始するには、次の手順に従ってください。

1. root でログインし、CDE フロント・パネルまたは「SysMan Applications」サブパネルで SysMan Station アイコンを選択します。

ただしこれは、SysMan Station アイコンが「SysMan Applications」サブパネルの下フロント・パネルに表示される、省略時設定の CDE 設定を想定しています。

2. 「アプリケーション・マネージャ」の「システム管理」グループから SysMan Station アイコンを選択します。

コマンド・プロンプトで SysMan Station を開始するには、次のように入力します。

```
#sysman -station
```

SysMan Station を起動すると、ローカル・ホストに接続されます。図 1-7 と同じような SysMan Station のメイン・ウィンドウが表示されます。ただし、省略時設定の表示では、監視することが可能なものとして「Filesystems...」、「Network...」、「Storage」、および「System...」オプションが表示されます。これらのオプションは、対象グループと呼ばれます。

図 1-7: SysMan Station のメイン・ウィンドウ



ポインタを対象グループに移動して MB1 をダブルクリックすることで、グループのイベント・データを取得できます。イベントの一覧を表示したウィンドウが表示されます。

SysMan Station は、階層 (ツリー) 構造をした、システムのグラフィカル表示です。たとえば、「Storage」オプションを使用すると、システムの全プロセッサ用の、すべてのバスのディスクをすべて表示できます。監視対象の特定のデバイスを選択したり、そのデバイスを管理するユーティリ

ティを起動することができます。また、個々のデバイスの詳細 (プロパティ) を表示することもできます。SysMan Station を使用して、システム、またはストレージ・デバイスなどの対象グループの表示 (ビュー) をカスタマイズすることもできます。カスタム・ビューを素早く表示して、デバイス状態を確認することができます。

SysMan Station のメイン・ウィンドウで利用できる機能は、次のとおりです。

| | |
|--------|---|
| Status | 「Status」ペインでは、対象グループを監視します。「Status」オプションについての詳細は 1.8.1 項 で説明しています。 |
| Views | 「Views」ペインでは、システムコンポーネントの特定のビューを選択します。「views」オプションについての詳細は、1.8.2 項 で説明しています。このペインでは、SysMan Station で作成するカスタマイズ・ビューも表示します。 |
| Menu | メニュー・オプションでは、ビューを変更したり、タスクを選択します。これらのオプションについての詳細は、1.8.3 項 で説明しています。その節では、カスタマイズ・ビューの保存方法についても簡単に説明しています。 |

1.8.1 SysMan Station の「Status」オプションの使用

SysMan Station を起動したとき、対象グループの状態が正常の場合には、「Status」ペインに大きなチェックマークのアイコンが表示されます。状態が悪くなると、アイコンの色が変化し、背景色が赤のバツ記号 (X) になります。これは、重大な問題が発生したことを示しています。これらのアイコンを使用して、対象グループのコンポーネントが通知したシステム・イベントを即座に表示することもできます。

監視することが可能な省略時設定の対象グループは、次のとおりです。

| | |
|--------------|------------------------------|
| File systems | UFS ファイル・システムまたは AdvFS ドメイン。 |
|--------------|------------------------------|

| | |
|---------|---|
| Network | tu0 など、ローカル・ホストに接続されたネットワークおよびデバイス。 |
| Storage | floppy など、バスやデバイス・インタフェースに接続されたストレージ・デバイス。floppy は、fdi0 などの fdi インタフェースに接続されたフロッピー・ドライブ・ユニットです。 |
| System | システム・コンポーネントに対応するイベントです。 |

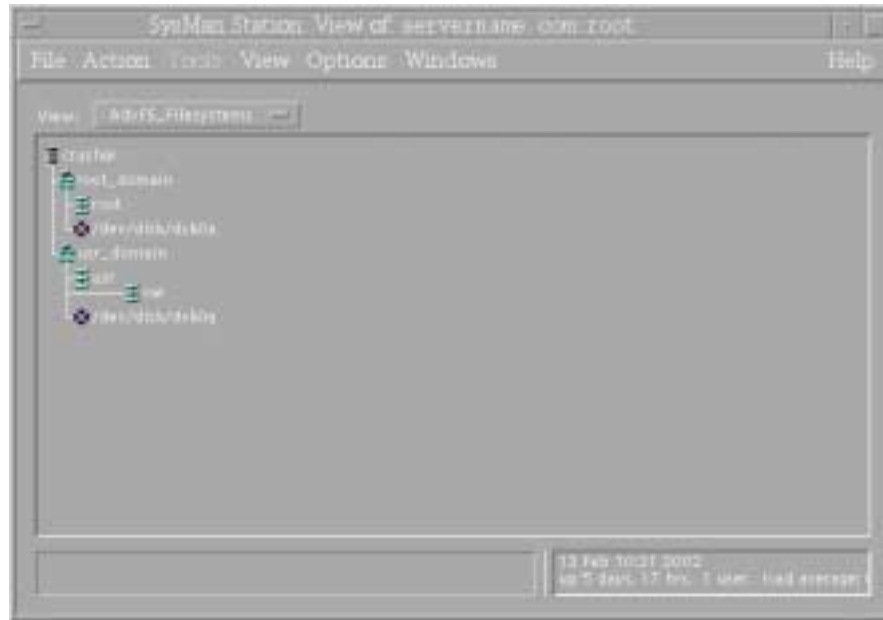
1.8.2 SysMan Station の「views」の使用

「views」オプション・メニューには、対象グループの一覧が表示されます。そのメニュー・オプションを選択して、ステータス・モニタ、または「views」ペインのグループの階層構造を示すウィンドウを表示することができます。グループは、次のとおりです。

AdvFS_Fileystems すべての AdvFS ドメインのビュー。

図 1-8 では、小規模の単一ディスク・システムにおける一般的な AdvFS ドメインのビューを示しています。

☒ 1-8: AdvFS_Fileystems View



Hardware

CPU から個々のディスクまで、すべてのデバイスを表示します。

図 1-9 では、小規模の単一プロセッサ・システムにおける一般的な Hardware View を示しています。

前の例には、システム・バス、および CD-ROM リーダである `cdrom0` などバスに接続されたさまざまなデバイスが表示されています。

図 1-9: Hardware View



Mounted_Filesys-

AdvFS_FileSystems のビューと同じフォーマットで、現在マウントされているファイル・システムを表示します。

Physical_Filesystems

AdvFS_Fileystems のビューと同じフォーマットで、使用可能なすべてのファイル・システム (UFS , AdvFS) を表示します。

システム内の最も重要な箇所だけを監視したり、ビューに表示されているコンポーネントを管理するアプリケーションを実行するように、ビューをカスタマイズして、保存することができます。ビューをカスタマイズした場合、そのビューを保存して、アイコンに割り当てることができます。これについては、1.8.3 項で説明しています。

システム・コンポーネントの画面では、コンポーネントを MB1 でクリックして、個々のシステムコンポーネントを選択し、階層構造の部分を展開したり縮小することができます。コンポーネントを選択するとき、MB3 を使用すると次の 1 つまたは複数のオプションを含むメニューが表示されます。メ

メニューに含まれるオプションは、選択したオブジェクトに対してオプションが適用されるかどうかによって異なります。

Display hierarchy functions:

[Expand] と [Contract]

これらのオプションは、コンポーネントの下のサブ・コンポーネントを表示したり見えなくしたりします。たとえば、SCSI バスを選択するときに [Expand] オプションを選択すると、そのバスに接続されているすべてのデバイスが表示されます。

[Contract] オプションを選択すると、表示されていたデバイスが見えなくなります。

[Hide] と [Unhide Children]

コンポーネントとそのサブ・コンポーネントを表示しないようにしたり、非表示されているコンポーネントを表示するようにしたりできます。たとえば、pci0 などの PCI バスを選択しているときに [Hide] オプションを選択すると、そのバスに接続されているすべてのデバイスが非表示になります。ここで、非表示になったデバイスは、そのバスを MB1 でダブルクリックしても、あるいは [Expand] メニュー・オプションを選択しても表示できません。

[Unhide Children] を選択すると、PCI バスに接続されたデバイスを表示できます。

Available SysMan Menu Utilities

このオプションは、コンポーネントに対して起動できる管理ユーティリティまたは設定ユーティリティを表示します。たとえば、ディスク・デバイスを選択して、ディスク構成ユーティリティを起動できます。

Properties

選択しているデバイスの特性と現在の構成設定の詳細を表示します。

オプションは、使用不可能な場合には淡色表示となります。

1.8.3 SysMan Station のメニュー・オプションの使用

SysMan Station のメイン・ウィンドウには、次のプルダウン・メニューとオプションがあります。これらは、マウスによる選択以外に、キーボードによる選択も可能です。

| | |
|------------|--|
| File | このメニューには、SysMan Station をクローズして終了するオプション、または他のシステムに接続するオプションが含まれています。 |
| Monitoring | このメニューを使用すると、対象グループ「Filesystems...」など、対象グループ全体を削除することにより「Status」ビューをカスタマイズすることができます。 |
| Options | このオプションを使用すると、初期ウィンドウの選択により、さらなる SysMan Station のカスタマイズが可能になります。 |
| Windows | このメニューを使用すると、種々の表示ビューを選択することができます。 |

SysMan Station を終了する前に、カスタム・ビューを保存するように促されます。そのとき、名前とアイコンをカスタム・ビューに割り当てることができます。SysMan Station を次に起動したときに、保存したカスタム・ビューが「Views」ペインに追加されます。

コンポーネントのビューのプルダウン・メニューのオプションは、次のとおりです。

| | |
|--------|--|
| File | 現在の画面をプリントするオプション、新たに接続を作成するオプション、および現在のウィンドウをクローズして SysMan Station を終了するオプションがあります。 |
| Action | コンポーネントのグループを変更するオプション、「Expand」および「Hide」オプションなど省略時設定の表示内容を変更するオプションがあります。 |

| | |
|---------|---|
| Tools | 選択されているコンポーネントに適用できる SysMan Menu ユーティリティの起動を行います。このウィンドウの内容は、選択されているコンポーネントまたはデバイスのタイプによって異なります。そのコンポーネントに適用できるユーティリティがない場合や、コンポーネントが何も選択されていない場合は、メニューには何も表示されません。 |
| View | 現在のシステム・ビューを制御し、ビューを切り替えることができます。 |
| Options | アイコン・サイズなど、ビューの表示内容を制御できます。 |
| Windows | メイン・ウィンドウなど、他のウィンドウを呼び出すことができます。 |

SysMan Station を終了する前に、カスタム・ビューを保存するように促されます。そのとき、名前とアイコンをカスタム・ビューに割り当てることができます。SysMan Station を次に起動すると、保存したカスタム・ビューがビュー・ペインに追加されます。

SysMan Station の Microsoft Windows でのインストールについての詳細は、1.10 節を参照してください。

1.9 HP Insight Manager

Web ベースの管理ユーティリティである HP Insight Manager は、異機種コンピューティング環境で使用でき、ネットワークに接続された任意のデバイスの情報にアクセスできます。デバイスとは、コンピュータ・システム、ネットワーク・プリンタ、あるいはネットワーク・コンポーネント (ルータなど) です。システムとそのコンポーネントあるいは周辺装置などの設定情報を取得でき、場合によっては、資産管理、資産の保全、負荷管理、またはイベント管理のような管理タスクを実行できます。

現在の実装では、HP Insight Manager は、DNOSysManGUI; や他の UNIX ユーティリティに対して、一貫性のあるラッパーを提供します。このラッパーにより、サポートされているシステムを Web ブラウザで管理することが

できます。Windows NT を実行している PC またはサーバでは、デバイスの詳細を表示したり、管理タスクを実行することができます。UNIX システムでは、HP Insight Manager を使用してデバイスの詳細を表示することはできますが、管理タスクを実行するには UNIX の SysMan Menu または SysMan Station を起動しなければなりません。

HP Insight Manager の機能は、いくつかのオペレーティング環境では、完全に実装されていますが、その他の環境ではまだ完全には実装されていません。つまり、Windows NT システムでは多くの機能を使用できますが、Tru64 UNIX では使用できない機能がいくつかあります。

HP Insight Manager の主要なサーバ構成要素は HP Insight Manager で、Windows NT に対して完全な管理サービスを提供するソフトウェア・コンソールです。このコンソールによって、ローカル・エリア・ネットワーク内のデバイス、またはエージェントを実行しているドメインと通信することができます。ここでは、デバイスとはネットワークに接続されているエンティティのことです。エンティティとは、すべての周辺装置を含むコンピュータ・システム、ネットワーク・プリンタ、またはルータなどです。エージェントの実行が可能な、アドレスを持つネットワーク・エンティティは、XE サーバとの通信が可能です。ただし、デバイスの中には、ハードウェアの追加が必要になるものもあります。

Web ブラウザによる管理を可能にするには、デバイスは HP Insight Manager により認識されるオペレーティング環境を備えていなければなりません。このようなオペレーティング環境は、WBEM ネットワークとデバイス情報を通信したり、WBEM ネットワーク内の他の (権限のある) デバイスから送信された命令を受信して実行できなければなりません。

オペレーティング環境では、標準プロトコルを使用してお互いに通信する HP 管理エージェントを実行できなければなりません。デバイスおよびそれらのオペレーティング環境は、データ・モデルを使用して、ハードウェアおよびソフトウェアの状態の情報を提供します。データ・モデルには、管理情報ベース (MIB)、簡易ネットワーク管理プロトコル (SNMP) などがあります。これらは、デバイスの管理可能な構成要素を表す、属性と値を持つオブジェクトのデータベースと考えられます。

HP Insight Manager は標準のプロトコルを使用してそのデータのためにデバイスをポーリングし、データベースの形式に依存しない、一貫した形式でその

データをユーザに渡します。この標準プロトコルを使用して、取得 (または操作) できるデバイス・データに対して一貫したラッパー処理を行います。

クライアント PC と、UNIX サーバまたは Windows NT サーバで構成された環境では、HP Insight Manager を管理タスクへの共通インタフェースとして使用できます。たとえば、特権 (root) ユーザとして PC で作業している場合、HP Insight Manager を起動して UNIX オペレーティング・システムを実行している AlphaServer の一般的なシステム状態を表示できます。また、SysMan Station を起動して、ディスク上のファイル・システムの状態など、周辺装置の特定の状態を検査できます。さらに、SysMan Menu タスクを起動して、そのファイル・システムに対して操作を行うこともできます。

ローカル・エリア・ネットワーク内の、エージェントを実行しているシステムのポートに Web ブラウザを接続して HP Insight Manager を使用します。たとえば、UNIX システムのホスト名とアドレスが `trout.cu.da.com` である場合には、ブラウザの「場所」(アドレス) フィールドに次の URL を入力します。

`HTTP://trout.cu.da.com:2301`

また、ホスト名とアドレスの代わりに `20.111.255.10` などの TCP/IP アドレスを指定することもできます。システムに接続した後は、ローカル・システムや状態を表示したり、ローカル・ネットワーク内の他のシステムを選択することができます。また、ローカル・デバイスの一覧からアドレスを選択することで、他のホストに接続することができます。

管理エージェントの設定と使用方法についての詳細は、`insight_manager(5)` と、`http://tru64unix.compaq.com/cma` のホーム・ページ HP Management Agents for Tru64 UNIX にある『*HP Management Agents for AlphaServers for Tru64 UNIX Reference Guide*』を参照してください。

1.10 パーソナル・コンピュータでの SysMan の使用

Java アプレットを使用する方法 (`insight_manager(5)` を参照) の他に、PC 上に SysMan クライアントをインストールし、このクライアントをデスクトップのアイコンから起動することも、[スタート] メニューから起動することもできます。

この機能では、Microsoft Windows、MacOS および Linux がサポートされています。この機能についての詳細情報と、必須ソフトウェアをダウンロードできるアドレスは、UNIX システムから利用可能な Web ページに説明してあり

ます。このページの URL は、`http://<host>:2301/SysMan_Home_Page` です。ここで、`<host>` には、ホスト名およびアドレス、または TCP/IP アドレスを指定します。次の手順に従ってください。

1. PC 上で、Netscape Web ブラウザを使用して `.../SysMan_Home_Page` ページを表示します。このページへは、省略時設定の UNIX ホーム・ページの、Tru64 UNIX SysMan アイコンからリンクされています。
2. このページの「*PC SysMan Client Software*」セクションまで下方向にスクロールしてください。

必要条件と制限を確認して、自身のクライアント・システムに関する必要条件をメモします。たとえば、クライアント・システム上では Internet Explorer Web ブラウザを使用しなければなりません。
3. 必須ソフトウェアをダウンロードします。
4. ソフトウェア・キットをクライアント・システム上に保存するか、または直接実行するかを尋ねてきます。後者のオプションでは、一般的な Windows のインストール手順でソフトウェアのインストールと設定を開始します。たとえば、インストール先の場所の入力が求められます。デスクトップの [スタート] メニューの既存のプログラム・グループまたは新規のプログラム・グループ、または指定したフォルダに、ショートカットを作成できます。
5. インストール処理が完了したら、「SysMan Station」および「SysMan Menu」が、Java アプリケーションとして、選択した場所に表示されます。必要なアプリケーションを起動します。
6. どちらかのアプリケーションを起動すると、次の接続オプションを示すダイアログ・ボックスがオープンされます。

| | |
|-----------|--|
| Host name | 接続するホストの名前およびアドレス、または TCP/IP 番号を入力します。省略時設定では、ローカル・ホストが表示されます。 |
|-----------|--|

| | |
|-------------|---|
| login as... | 現在のログイン名を使用するか、または新しいログイン名を使用するかを選択します。たとえば、すでにクライアント・システムに通常のユーザでログインしている場合、そのホスト上 |
|-------------|---|

で特権タスクを実行するためには、新たに root ユーザで接続する必要があります。

Set X/Motif® display 出力するディスプレイをリダイレクトする場合は、このボックスにチェック・マークを付けて出力先ディスプレイのアドレスを指定します。

[OK] を選択すると、アプリケーションのウィンドウがオープンします。アプリケーションの起動にかかる時間は、ネットワークの転送速度と通信量によって異なります。前述の節で説明したように、SysMan Station または SysMan Menu が使用できるようになります。

1.11 シリアル回線コンソールのセットアップ

モデム接続を介してリモート・システムを管理できます。シリアル回線コンソールを使用すると、ローカル・システムに接続されているモデムを介して、ローカル端末を、リモート・システム・コンソールおよびリモート・システムの通信ポート COMM1 に接続することができます。ローカル・システムは、モデム接続が可能であれば、どのような端末または端末エミュレーション・デバイスでも構いません。たとえば、ダム端末、X 端末ウィンドウ、パーソナル・コンピュータ (PC) などが使用できます。管理タスクを実行するには、root (または管理特権を持つアカウント) としてログインできなければなりません。

この接続はコンソール・ポートと呼ばれます。端末接続は、プロセッサがサポートするコンソール・ファームウェアに応じて、57,600 までの通信レートをサポートします。現在のところこの機能は、AlphaServer 1000A のような、コンソール・デバイスとしてモデムをサポートするシステムでのみ使用可能です。使用しているシステムでこのような機能が使用できるかどうかについては、システム・ハードウェアのマニュアルを参照してください。

コンソール・ポートを使用すると、次の操作を実行することができます。

- `tip`, `telnet`, または PC 端末エミュレーション・ユーティリティなどのユーティリティを使用してリモート・システムに接続する。
- システムをリモートでブートしたりシャットダウンして、すべてのブート・メッセージを監視する。
- カーネル・デバッガを開始して、デバッガ・メッセージを監視する。

- コマンドおよびユーティリティを使用して、システム管理タスクを実行する。

リモート・システムで環境構成ユーティリティ (ECU: Environment Configuration Utility) を実行すると、モデム接続が切断されます。このため、コンソール・デバイスとしてモデムを設定して使用する前に、ECU を使用して環境構成を完了しておく必要があります。

1.11.1 コンソール・ポートの設定

以降の各項で、シリアル回線コンソール・ポートを設定し、ダイアル・イン用にリモート・モデムを設定する手順の概要を説明します。ローカル(ダイアル・アウト) モデムはすでにインストールされ、使用できるように構成されているものとします。

1.11.1.1 モデムの **COMM1** への接続

リモート・システムの `CONSOLE` 環境変数を `serial` に設定しておく必要があります。

モデムに付属のハードウェア・マニュアルを参照して、モデムをシステムに接続してください。正しいモデム設定値を取得し、適切なシステム・ファイル・エントリを作成する方法については、`modem(7)` を参照してください。特に、`/etc/inittab` ファイルの `cons` エントリを変更して、`getty` または `uugetty` プロセスが `COMM` ポートを正しく設定できるようにする必要があります。この行は、次のような行です。

```
cons:1234:respawn:/usr/sbin/getty console console vt100
```

コンソール・デバイスとしてボー・レート 38,400 で動作するように設定されたモデムを使用する場合は、この行を次のように変更します。

```
cons:1234:respawn:/usr/sbin/getty console M38400 vt100
```

1.11.1.2 構成可能な **DCD** タイマ値の設定

シリアル・ドライバが変更されて、Carrier Detect (DCD) タイムアウト値が構成できるようになっています。このタイマの省略時の値は2秒ですが、これは DEC STD-052 規格に従うものであり、ほとんどのモデムに適しています。このタイマは、DCD 信号がドロップした場合に、回線の切断を宣言して DTR および RTS 信号をドロップするまでにドライバが待機しなければならない時間を決定するために使用されます。モデムの中には DTR が短時間

の間隔でドロップすると想定するものがあるので、モデムのマニュアルを参照して、間隔を確認してください。

タイマの変更は、`/etc/sysconfigtab` ファイルまたは `sysconfig` コマンドで、0 (タイムアウト期間なし)、1、あるいは2 秒に設定することができます。`/etc/sysconfigtab` ファイルでタイマを設定するには、このファイルを編集して次の行を追加します。

```
ace:
    dcd_timer=n
```

この場合、`n` には 0、1、または 2 を指定します。

`sysconfig` コマンドでタイマを変更する場合には、次の構文を使用します。

```
# sysconfig -r ace dcd_timer=n
```

この場合、`n` には 0、1、または 2 を指定します。

`sysconfig` コマンドで値を変更した場合には、システムのリブート時に設定が失われることに注意してください。リブートしても設定が失われないようにするには、`/etc/sysconfigtab` ファイルを編集します。

1.11.1.3 コンソール環境変数の設定

`COM1_MODEM`、`COM1_FLOW`、および `COM1_BAUD` コンソール環境変数の設定は、モデム用にシステム・ファイルのエントリを作成した際に使用した `getty` または `ugetty` 設定と同じ値でなければなりません。

コンソール環境変数の設定方法についての説明は、ハードウェアのマニュアルを参照してください。通常、変数の設定は、次の例に示すように、システムのシャットダウン時にコンソール・モードで行います。

```
>>> set COM1_MODEM ON
>>> set COM1_FLOW SOFTWARE
>>> set COM1_BAUD 9600
```

有効な設定は次のとおりです。

- `COM1_MODEM`: ON または OFF
- `COM1_FLOW`: NONE、HARDWARE、SOFTWARE、BOTH
- `COM1_BAUD`: ご使用のシステムのハードウェア・マニュアルを参照

ボー・レートやフロー制御，またはモデム設定を変更すると (たとえば `getty` コマンドを使用して)，その変更はコンソール・レベルまで通知されて，環境変数が自動的に変更されます。

1.11.1.4 モデム設定の確認

リモート・システムがブートされない場合は，そのシステムにダイヤルして，ログイン・プロンプトまたはコンソール・プロンプトを表示させます。ログアウトまたは接続を切断して，回線が正しくハングアップすることを確認してください。再度ダイヤル・インして，再接続できることを確認します。

1.11.2 コンソール・ポート接続の開始

別の方法で，ローカル・システムとリモート・システム間の接続を開始することができます。 `tip`，`kermi`，`cu` 接続は，端末または X 端末ウィンドウから開始することも，あるいは PC ベースの端末エミュレータを使用して開始することもできます。

たとえば，次のように `tip` コマンドを使用することができます。

```
# tip [telephone number]
# tip cons
```

この例で，`telephone_number` には，外線用の番号や市外局番を含めて，リモート・システムの電話番号を指定します。2 行目は `/etc/remote` ファイルのエントリ例であり，このファイルでリモート・システムおよび `tip` 設定を詳細に指定することができます。

ダイヤル・アウト・コマンドを開始して，2 つのモデムの接続が確立されると，`connect` という文字がローカル端末のウィンドウに表示されます。Return キーを押すと，コンソール・プロンプト (`>>>`) または `login:` プロンプトが表示されます。

詳細については，`tip(1)` を参照してください。

1.11.2.1 コンソール・ポートの使用

システムにアクセスして特権アカウントにログインすると，グラフィカル・ユーザ・インタフェースにアクセスする必要のない管理タスクであれば，コマンドの使用やユーティリティの実行など，本書で説明しているすべての管理タスクを実行できます。次の機能は，リモート管理に有効です。

uucp

UNIX システム間コピー・ユーティリティはスクリプトやファイルをリモート・システムにコピーします。詳細については、`uucp(1)` を参照してください。

ikdebug

カーネル・デバッグ・ツール `ikdebug` はリモートで起動して使用することができます。詳細については `ikdebug(8)` を参照してください。場合によっては、`/etc/remote` ファイルのエントリを変更して、正しいボー・レートに修正する必要があります。たとえば、次の行のボー・レートを 9600 から変更しなければならない場合があります。

```
# access line for kernel debugger
kdebug:dv=/dev/tty00:br#9600:pa=none:
```

詳しい説明については、『`Kernel Debugging`』を参照してください。

1.11.2.1.1 コンソール・ログ・メッセージのオフ

現在、`syslogd` デーモンには内部スイッチがあり、コンソールへのメッセージ出力を無効にしたり有効にしたりすることができます。この機能は、`syslogd` コマンド行に `-s` フラグを指定するか、または次のコマンドを実行することにより起動します。

```
# /usr/sbin/syslog
```

詳細は、`syslog(1)` を参照してください。

1.11.2.1.2 リモート・システムのシャットダウン

リモート・システムをシャットダウンすると、モデム接続はドロップします。これを回避するには、システムをシャットダウンする前に、次のコマンドを実行してください。

```
# stty -hupcl
```

詳細については、`stty(1)` を参照してください。

シャットダウンが完了すると、コンソール・プロンプトにアクセスできます。

1.11.2.1.3 リモート・セッションの終了

オペレーティング・システムのシェル・プロンプトからリモート・セッションを終了するには、Ctrl/d を押してログアウトしてリモート・セッションを終了します。別の方法では、+++ をタイプしてモデムをローカル・コマンド・レベルにし、ATH をタイプした後に Return キーを押して、接続をハング・アップします。

1.11.3 トラブルシューティング

システムの設定および接続に問題がある場合には、設定に関して次の点を調べてください。

- ローカル・モデムがダイヤル・アウトしない。
ケーブルと接続を調べて、電話線のプラグが正しいソケットに差し込まれ、ダイヤル・トーンが聞こえることを確認してください。
- リモート・モデムが応答しない。
リモート・モデムが自動応答 ATSO=n に設定されていることを確認してください。この場合 n には、モデムが応答するまでに鳴るベルの回数を指定します。
modem(7) を参照して、ダイヤル・イン・アクセスの設定を確認してください。
- リモート・モデムが応答した後に接続が切断される。
これは、誤ったダイヤル・イン・アクセスの設定が原因である場合がほとんどです。modem(7) を参照して、ダイヤル・イン・アクセスの設定を確認してください。
- リモート・モデムが応答するが、文字がランダムに出力されるだけである。
この問題は通常、COMM ポートのボー・レートとモデムのボー・レートの不一致によって起こります。modem(7) を参照して、ダイヤル・イン・アクセスの設定を確認してください。
- shutdown コマンドでリモート・システムをシャットダウンすると接続がドロップされる。
stty 属性 hupcl が省略時の設定です。シャットダウン中に回線が切断されるのを回避するには、次のコマンドを使用します。

```
# stty -hupcl
```

システムの起動とシャットダウン

この章では、次の情報について説明します。

- システムの起動およびシャットダウンの概要 (2.1 節)
- ブート操作 (2.2 節)
- システムをブートするための準備 (2.3 節)
- システムのブート方法 (2.4 節)
- 各種のシステム実行レベル (2.5 節)
- システムの実行レベルの変更方法 (2.6 節)
- マルチプロセッサ・システムでのブート上の考慮事項 (2.7 節)
- システムの日付および時刻の設定方法 (2.8 節)
- ブートの問題に対するトラブルシューティング方法 (2.9 節)
- システムのシャットダウン用のオプション (2.10 節)
- マルチユーザ・モードからシステムをシャットダウンする方法 (2.11 節)
- シングルユーザ (root) モードからシステムをシャットダウンする方法 (2.12 節)

2.1 シャットダウンとブート操作の概要

システムのシャットダウンと再起動は、システム管理者が定期的に行う日常的な業務です。コンピュータ環境によっては、システムが常に動作していて、いつでも利用可能な状態でなければなりません。このようなシステムでは、定期的な保守作業やソフトウェアのアップグレード時にだけシャットダウンされます。

Tru64 UNIX には、シャットダウンの回数を極力減らし、システムの動作時間を最大化するために、以下の機能があります。

- AdvFS (Advanced File System) には、システムをオフラインにすることなく、バックアップを可能にする機能があります。

- Event Manager , sys_check , およびメモリ巡視機能のような診断ツールには、システムの問題を初期段階で検出し、それがシャットダウンの原因になる前に訂正する機能があります。Event Manager は、第 13 章で説明されています。sys_check についての詳細は、第 12 章を参照してください。メモリ巡視機能は、『*Managing Online Addition and Removal*』で説明されています。
- CPU , およびディスクやテープのようなストレージ構成要素のホット・スワップ機能は、構成要素の故障が原因となる予期しないシャットダウンを回避するのを助ける機能です。CPU のホット・スワップ機能は、最近のハードウェア・プラットフォームだけでサポートされています。詳細は、『*Managing Online Addition and Removal*』を参照してください。ストレージ構成要素のホット・スワップ機能は、大部分の最近のプラットフォームでサポートされています。詳細は、ご使用中のプロセッサのハードウェア・ドキュメント、『ハードウェア管理ガイド』 , および hwmgr(8) を参照してください。
- クラスタ・システムの場合は、クラスタを動作させたままでも、1 つのクラスタ・メンバ上で、ソフトウェアのアップグレードを行うことができます。この機能は、「ローリング・アップグレード」と呼ばれます。

通常は、システム・ユーザへの影響が最小になるように簡単にシステムをシャットダウンすることができます。場合によっては、システムを迅速にシャットダウンしなければならないために、ユーザにある程度の混乱をもたらすことがあります。またシステム管理者が制御できないような状況でシステムが突然シャットダウンして、ユーザに多大な影響を及ぼすことがあります。それぞれのサイトの運用マニュアルでは、次の事項を規定してください。

- 計画的シャットダウンの手順とスケジュール。
- シャットダウン原因の調査手順と、次の事項の手順。
 - エラーまたは問題の訂正手順。トラブルシューティングの情報については、第 11 章、第 12 章、および第 14 章を参照してください。
 - できるだけ早くシステムをオンラインに戻す手順。
 - 必要に応じて、紛失したデータを回復する手順。システムのバックアップについては、第 9 章を参照してください。

2-2 システムの起動とシャットダウン

システムをシャットダウンするには、root (スーパーユーザ) の権限が必要です。システム構成に従って、システムのシャットダウンとリブートで利用できるオプションがいくつかあります。

2.1.1 シャットダウンの方法

システムは、自動でシャットダウンすることも、手動でシャットダウンすることもできます。

自動シャットダウン

あるシステム・イベントの発生でシステムを自動的にシャットダウンするように、環境モニタリングのようなシステム監視ツールを構成することができます。イベント管理については、第 13 章を参照してください。

手動シャットダウン

- SysMan Menu と SysMan Station を使用すると、ローカルまたはリモートのシステムまたはクラスタをシャットダウンできます。SysMan Menu の「一般的なタスク」ブランチには、SysMan Menu をどのようににアクセスしたかに従って適切なユーザ・インタフェースを呼び出す、「システムのシャットダウン」タスクがあります。また、次のコマンドを使用して、コマンド行から上記のタスクを実行することもできます。

```
# sysman shutdown
```

詳細については、第 1 章を参照してください。

- /usr/sbin/shutdown コマンド行インタフェースを、文字セル端末で実行します。利用可能なコマンド・オプションについては、shutdown(8) を参照してください。
- CDE の「アプリケーション・マネージャ - 日常管理」フォルダの「シャットダウン」アイコンは、SysMan Menu のタスク「システムのシャットダウン」を呼び出します。

2.1.2 ブート方式

オペレーティング・システムのブートは、システム・コンソールを使って実行します。システムに電源が投入されると、コンソール・プロンプトの記号 >>> が表示されます。このプロンプトで、コマンドを入力したり、システムのブート時の動作を制御する変数などの、システム構成変数を設定できま

す。この章では、記号 >>> はシステム・プロンプトを表します。コンソールは、System Reference Manual (SRM) コンソールまたはファームウェア・コンソールとも呼ばれます。コンソール・プロンプトに入力できるコマンドについては、システムに付属のオーナーズ・マニュアルを参照してください。

次のようにシステムをブートすることができます。

- コンソールから手動でブートできます。
- リモート・コンソール方式などのネットワークまたはモデムの接続を使ってブートすることができます。リモート・コンソール方式については、第 1 章を参照してください。
- シャットダウン後に実行されるブート動作を指定します。たとえば、SysMan Menu または SysMan Station を使用してシャットダウンを開始した場合は、シャットダウンが完了してからシステムをシングルユーザ・モードで自動的にリブートするように設定できます。
- `auto_action` コンソール変数を設定すると、システムを自動的にブートできます。これにより、たとえば電源の異常などのシステムの予期しないシャットダウンの後に、自動的にブートが行われます。この動作は、自動ブートとも呼ばれます。

2.1.3 関連ドキュメント

システムのシャットダウンとリブートに関する追加ドキュメントは、マニュアル、リファレンス・ページ、およびオンライン・ヘルプにあります。

2.1.3.1 マニュアル

Tru64 UNIX オペレーティング・システムのドキュメント・セットの中で、システムのシャットダウンとリブートについて説明しているものを以下にリストします。

- コンソール・コマンドとコンソール変数の詳細については、システムに付属のオーナーズ・マニュアルを参照してください。 `consvar(8)` を参照してください。 `consvar` は、ファームウェアのリビジョンに応じて、オペレーティング・システム内でコンソール環境変数を操作するためのコマンドです。
- 損傷のあるファイル・システムのチェックと修復をリブート前に行う場合は、『*AdvFS 管理ガイド*』と『*Logical Storage Manager*』のファイル・システムに関する章を参照してください。

- システムのインストレーションと初期ブート操作については、『インストレーション・ガイド』を参照してください。この章では、インストール済みのオペレーティング・システムのブートまたはリブートについて説明します。
- 『*Kernel Debugging*』には、クラッシュ・ダンプ・ファイルの解析方法が示されています。
- 『ハードウェア管理ガイド』には、ハードウェアの問題を診断する方法が示されています。

このマニュアルには、シャットダウンの計画や管理、およびエラー回復に関する次の情報についても説明があります。

- システムによっては、冷却ファンの故障などの問題が発生した際に自動的にシステムをシャットダウンするために使用できる、環境モニタリングをサポートしています。この機能の構成方法については、第 12 章を参照してください。
- また、エラー状況、ログ・ファイル、およびクラッシュ・ダンプについては、第 12 章を参照してください。
- Event Manager および SysMan Station には、統合化されたモニタ機能およびイベント報告機能があります。この機能によって、ローカル・システム、リモート・システム、およびクラスタがモニタできます。これらの機能呼び出す方法については、第 1 章を参照してください。
- リモートからシステム管理作業を行う場合や、ネットワークに障害がありダイヤルアップ通信が必要な場合は、第 1 章の 1.11 節でリモート・シリアル・コンソールについて参照してください。
- 必要であれば失われたデータを回復できるようにバックアップをスケジュールする方法については、第 9 章を参照してください。

2.1.3.2 リファレンス・ページ

次のリファレンス・ページには、各コマンドのオプションとインタフェースについての詳細情報が記載されています。

| | |
|-------------|---|
| shutdown(8) | shutdown コマンド行インタフェースの呼び出し方法と使用方法について説明しています。 |
|-------------|---|

| | |
|--|---|
| <code>sysman(8)</code> および <code>sysman_station(8)</code> | SysMan オプションの使用方法について説明し、これらのユーティリティを呼び出して「システムのシャットダウン」タスクを実行する方法を説明しています。 |
| <code>wall(1)</code> および <code>rwall(1)</code> | ユーザに対して、システムのシャットダウンを警告するユーティリティです。 |
| <code>halt(8)</code> および <code>fasthalt(8)</code> | システムを停止するユーティリティです。 |
| <code>reboot(8)</code> および <code>fastboot(8)</code> | システムをブートするユーティリティです。 |
| <code>fsck(8)</code> | UFS ファイル・システムを検査し、修復するユーティリティです。 |
| <code>init(8)</code> | システムを初期化するユーティリティです。 |
| <code>rc0(8)</code> , <code>rc2(8)</code> , および <code>rc3(8)</code> | システムを停止するとき、実行レベル 2 に入るとき、および実行レベル 3 に入るときに実行するコマンド・スクリプトです。 |
| <code>consvar(8)</code> | システム・ファームウェアのコンソール環境変数を操作するコマンドです。 |

2.1.3.3 オンライン・ヘルプ

次のオンライン・ヘルプを利用することができます。

- `shutdown -h` コマンドにより、コマンド行オプションのヘルプが表示されます。
- SysMan Menu で提供されるシャットダウン・ユーティリティは、強化されたオンライン・ヘルプ機能を備えています。
- SysMan Menu および SysMan Station の各タスクでは、それぞれのオンライン・ヘルプを表示できます。ヘルプの概要は、
`/usr/doc/netscape/sysman/index.html` を参照してください。

オンライン・ヘルプを呼び出す方法については、第 1 章を参照してください。

2.1.4 システム・ファイル

ブート操作とシャットダウン操作では、次のシステム・ファイルを使用します。

| | |
|--|---|
| <code>/etc/inittab</code> | <code>init</code> プログラムに、初期化プロセスの作成および実行手順を指定します。 |
| <code>/vmunix</code> | カスタム・カーネルの省略時の名前です。カスタム・カーネルを構築する際には、正当なものであれば任意のファイル名を選択できます。 |
| <code>/genvmunix</code> | 汎用カーネルの省略時の名前です。カスタム・カーネルを構築する場合、またはカスタム・カーネルが破損してブートできない場合は、汎用カーネルをブートします。 |
| <code>/sbin/rc0</code> , <code>/sbin/rc2</code> , および <code>/sbin/rc3</code> | <p>実行レベルのコマンド・スクリプトです。</p> <p><code>rc0</code> スクリプトには、シャットダウンを速やかに行い、システムをシングルユーザ状態にする実行コマンドが入っています。この実行コマンドは、<code>/sbin/rc0.d</code> ディレクトリに置かれます。</p> <p><code>rc2</code> スクリプトには、マルチユーザ状態、実行レベル 2 に初期化する実行コマンドが入っています。この実行コマンドは、<code>/sbin/rc2.d</code> ディレクトリに置かれます。</p> <p><code>rc3</code> スクリプトには、マルチユーザ状態、実行レベル 3 に初期化する実行コマンドが入っています。この実行コマンドは、<code>/sbin/rc3.d</code> ディレクトリに置かれます。</p> |

2.1.5 関連ユーティリティ

次のユーティリティも、ブート操作の際に使用する可能性があります。

| | |
|-------------------|--|
| <code>fsck</code> | <code>fsck</code> コマンドは、 <code>ufs_fsck</code> プログラムのラッパー・プログラムです。このコマンドは、UFS ファイル・システムを検査し、修復します。AdvFS ファ |
|-------------------|--|

イル・システムの検査については、advfs(4) および『AdvFS 管理ガイド』を参照してください。

consvar

consvar コマンドを使用すると、オペレーティング・システムの動作中に、コンソール環境変数の取得、設定、リスト表示、保存ができます。

システムが consvar をサポートしているかどうかは、次のコマンドで調べることができます。

```
# /sbin/consvar -l
auto_action = HALT
boot_dev = dsk0
bootdef_dev = dsk0
booted_dev = dsk0
boot_file =
booted_file =
boot_osflags = A
.
.
.
```

consvar がサポートされている場合、コンソール変数の現在の設定が表示されます。

2.2 ブート操作の概要

オペレーティング・システムをブートすると、正常に動作するために必要な一連のタスクが開始されます。ブート中は、カーネルをメモリにロードし、動作に必要なルーチンを初期化するため、システムは障害が発生しやすい状態にあります。したがって、システムのブート処理中はシステムの状態を把握し、問題が発生した場合には迅速に対応できるようにしておいてください。

2.2.1 自動ブートと手動ブート

システムのブート方法には自動ブートと手動ブートがあります。自動ブートの場合、システムが初期化プロセスを開始し、ブートが完了するか障害が発生すると、自動ブート操作を終了します。なんらかの理由で自動ブートに失敗したときのみ、手動でブートする必要があります。たとえば fsck コマンドでファイル・システムを検証できない場合です。

手動ブートの場合、システムが初期動作を実行した後で制御をユーザに渡し、ユーザの操作が終了するとシステムに制御が戻りブート操作を終了しま

す。システムをシングルユーザ・モードでブートする場合は、手動ブートを使用します。自動または手動ブートの結果、正常終了するか異常終了するかによって次のような状態になります。

- ブート操作が正常終了した場合

システムは初期化されます。シングルユーザ・モードでは、コンソールまたは端末画面にスーパーユーザ・プロンプト `#` が表示されます。マルチユーザ・モードでは、`login` プロンプトまたはスタートアップ表示が表示されます。プロンプトまたはスタートアップ表示は、ハードウェアあるいは起動ソフトウェアによって異なります。

- ブート操作が異常終了した場合

エラー・メッセージとその後にコンソール・プロンプト (`>>>`) が表示されます。エラーの状態によっては、コンソール・プロンプトを表示しない状態でシステムがハングします。

2.2.2 シングルユーザ・モードまたはマルチユーザ・モードへのブート

システムはブートの後、シングルユーザ・モードかマルチユーザ・モードになります。

`init` 操作は、制御をユーザに渡すに前に起動スクリプトを呼び出さないため、ルート・ファイル・システムは読み取り専用モードでマウントされます。ネットワークやその他のデーモンのスタートアップは行われず、ファイルの検査や修正も行われません。また、システムの完全な使用に必要なその他の操作も、自動的に実行されません。

他のユーザが並行して処理を行う可能性がなく、特定の管理者用のタスクのみを実行する場合は、シングルユーザ・モードにブートします。必要なタスクを手動で実行した後、`Bourne` シェルを終了します。たとえば、新しいハードウェアの検査、異常のあるファイル・システムのマウントと検査、ディスク・パーティションの変更、またはシステム・クロックの設定などを行います。作業が終了したらシステムに制御を戻します。`init` 操作は起動タスクを続行し、マルチユーザ・モードでブートします。

マルチユーザ・モードへのブートでは、システムがカーネルをロードし、ハードウェアおよび仮想メモリの初期化、資源割り当て、スケジューリング、構成、モジュールのロードなどの処理を実行します。

主要な初期化タスク (プロセス 0) が終了すると、`init` プロセス (プロセス 1) は `/etc/inittab` ファイルを読み取り、そのファイル内の命令に従って動作し、コマンド実行スクリプトの実行などの一連の補助タスクを開始します。コマンド実行スクリプトには、ファイル・システムのマウントと検査、一時的ファイルの削除、クロック・デーモンの初期化、ネットワーク・デーモンの初期化、プリント・スプーリング・ディレクトリおよびデーモンのセットアップ、エラー・ロギング機能の設定、スクリプト内または関連ディレクトリで指定されたその他のタスクの実行などの処理を開始するためのエントリが含まれています。

これらの処理が終了するとシステムは使用可能な状態になります。

Tru64 UNIX システムでは、カスタム・カーネルがブートできない場合、ブートするカーネルを選択することができます。システムで発生している問題を解決するために汎用カーネル (`/genvmunix`) でブートすることができます。また、新しいドライバのテストや既存カーネルへのオプションの追加のために、別のカスタム・カーネルでブートすることもできます。

2.3 インストール済みシステムのブート

システム管理者は、さまざまな状態からシステムをブートする必要があります。以降の項では、次に示すシステム状態からのリブートの準備と開始のための手順を説明します。

- 電源が切られているシステム (2.3.1 項)
- 電源は入っているが停止しているシステム (2.3.2 項)
- シングルユーザ・モードでブートしているシステム (2.3.3 項)
- クラッシュしたシステム (2.3.4 項)
- ネットワークから切り離されたネットワーク・システム (2.3.5 項)

注意

システムがシングルユーザ・モードで実行されている場合、`ed` エディタを使用するためにはルート・ファイル・システムの保護を読み取り/書き込み可能に変更する必要があります。スーパーユーザ・プロンプトに対して、次のコマンドを入力してください。

```
# mount -u /
```

2.3.1 電源が切られているシステムのブート

電源を投入し、システムをブートする場合は、次の手順に従ってください。

1. ハードウェアとすべての周辺デバイスが接続されていることを確認します。

診断出力の情報とそれに対する対処方法については、ハードウェアのオペレータ・マニュアルを参照してください。

2. 周辺デバイスの電源をいれます。周辺デバイスの起動については、オペレータ・マニュアルまたはハードウェアのユーザ・マニュアルを参照してください。

3. プロセッサの電源をいれます。

4. ハードウェアの起動オペレーションと診断オペレーションが終了したことを確認します。

ほとんどのハードウェアは、起動オペレーションで診断検査を行います。ハードウェアの再起動および診断オペレーションについては、ハードウェアのオペレータ・マニュアルを参照してください。

5. コンソール・プロンプト (>>>) が表示されるのを待ちます。電源投入時にシステムが自動的にブートされるように設定してある場合は、停止ボタンを押してコンソール・プロンプトを表示させてください。システムの停止ボタンの位置は、ハードウェアのオペレータ・マニュアルを参照してください。システムの省略時のブートについての詳細は、2.4 節を参照してください。

6. 起動モードを決めます。

| | |
|-------------|---|
| シングルユーザ・モード | システム管理者が実行するタスクがあり、root 以外のユーザがシステムへアクセスするのを制限したい場合は、このモードを使用します。 |
|-------------|---|

| | |
|------------|--|
| マルチユーザ・モード | シングルユーザ・アクセスの制限を必要とせず、システムのすべての機能を初期化する場合は、マルチユーザ・モードの 1 つ (ネットワーク |
|------------|--|

ングなしまたはネットワーキングありのマルチ
ユーザ・モード) でブートします。

7. 希望する起動モードに対応するブート・コマンドを入力します。システムのブートに必要なコマンドと手順については、2.4 節を参照してください。

2.3.2 電源は入っているが停止しているシステムのブート

ハードウェアに電源が入っていて使用可能な状態ではあるがプロセッサが停止している場合、システムはコンソール・モードになっています。たとえば、`shutdown -h` コマンドでプロセッサをシャットダウンした後や、`halt` コマンドを実行した場合には、コンソール・プロンプト (`>>>`) が表示されます。

システムがコンソール・プロンプトを表示している場合、次の手順に従ってシステムをブートしてください。

1. 起動モードを決めます。

| | |
|-------------|--|
| シングルユーザ・モード | システム管理者が実行するタスクがあり、 <code>root</code> 以外のユーザがシステムへアクセスするのを制限したい場合は、このモードを使用します。 |
| マルチユーザ・モード | シングルユーザ・アクセスの制限を必要とせず、システムのすべての機能を初期化する場合は、マルチユーザ・モードの1つ (ネットワーキングなしまたはネットワーキングありのマルチユーザ・モード) でブートします。 |

2. 希望する起動モードに対応するブート・コマンドを入力します。システムのブートに必要なコマンドと手順については、2.4 節を参照してください。

2.3.3 シングルユーザ・モードからのブート

システムの電源が投入されて使用可能になっており、プロセッサが動作していて、かつアクセスが `root` に制限されている場合は、システムはシングルユーザ・モードになっています。

システムがスーパーユーザ・プロンプト (#) を表示している場合、次の手順に従って、マルチユーザ・モードへ移行する準備を行ってください。

1. シングルユーザ・モードで処理を続行するか、マルチユーザ・モードへ移行するかを決めます。

| | |
|-------------|---|
| シングルユーザ・モード | システム管理者がさらに実行するタスクがあり、root を除くすべてのユーザのアクセスを制限したい場合は、このモードで処理を続行します。 |
|-------------|---|

| | |
|------------|--|
| マルチユーザ・モード | シングルユーザ・アクセスの制限を必要とせず、システムのすべての機能を初期化する場合は、マルチユーザ・モードの1つ (ネットワーキングなしまたはネットワーキングありのマルチユーザ・モード) でブートします。 |
|------------|--|

2. マルチユーザ・モードへ移行する準備ができたなら Ctrl/d を押します。
システムのブートに必要なコマンドと手順については、2.4 節を参照してください。

2.3.4 クラッシュしたシステムのブート

システムがクラッシュし、自動的に回復してリポートすることができない場合は、次の手順に従ってシステムをブートしてください。

1. クラッシュ・ダンプ・ファイルの保存方法と、クラッシュの原因をシステム・ログ・ファイルで検査する方法については、第 12 章を参照してください。
2. ハードウェアとすべての周辺デバイスが接続されていることを確認します。
3. 必要なら、ハードウェアの電源をいれます。
必ず、プロセッサより先に周辺デバイスおよびその他のデバイスの電源をいれます。
4. ハードウェアの再起動オペレーションと診断オペレーションを監視します。

診断出力の情報と対処方法については、ハードウェアのオペレータ・マニュアルを参照してください。

- 診断テストでハードウェア障害が示された場合

弊社の各支店/営業所に連絡してください。ハードウェアの障害は重大な問題なので、処理を続行したり、欠陥のあるハードウェアを無視して使用しないでください。

- システムが自動的にブートするように設定してある場合

停止ボタンを押してコンソール・プロンプトを表示させます。システムの停止ボタンの位置については、ハードウェアのオペレータ・マニュアルを参照してください。

5. 起動モードを決めます。

| | |
|-------------|--|
| シングルユーザ・モード | root 以外のユーザがシステムへアクセスするのを制限したい場合は、このモードを使用します。クラッシュが発生したら、シングルユーザ・モードで作業を開始するのが賢明です。すべてのファイル・システムに矛盾がないかどうか完全に確認し、その他クラッシュ時に必要な措置を行った上で他のユーザのシステムへのアクセスを使用可能な状態にします。 |
| マルチユーザ・モード | ログイン・パーミッションを持つすべてのユーザにアクセスを許可する必要がある場合は、マルチユーザ・モードの1つ(ネットワークなしまたはネットワークありのマルチユーザ・モード)でブートします。 |

- #### 6. 必要なブート・コマンドを入力します。システムのブートに必要なコマンドと手順については、2.4 節を参照してください。

2.3.5 ネットワークから切り離されたシステムのブート準備

システムがネットワークをサポートするように構成されている場合、ブート操作では、構成されているすべてのネットワーク・サービスを起動しようとします。これにより、システムが停止したり、サービスが存在するかをテストするために非常に長い時間がかかります。ネットワークから動作中のシステムを削除して、システムをスタンドアロン・モードで使用したり、故障したネットワーク・デバイスなどのシステムの問題を訂正することもできます。

サービスを再構成することなくシステムがネットワークから切り離された場合や、システムがクラッシュしたためネットワークから切り離さなくてはならない場合は、追加の手順を実行してからシステムをリブートします。

また次の場合、動作中のシステムをネットワークから切り離さなければならないことがあります。

- システムをスタンドアロン・モードで使用する場合
- 故障したネットワーク装置などの、システム障害を修復する場合

次の手順では、システムがコンソール・プロンプトの状態では停止しているものとします。

1. コンソール・プロンプトで `boot_osflags` 環境変数に `s` を設定し、シングルユーザ・モードでブートが停止するようにします。

```
>>> set boot_osflags s
```

代替ディスクからブートしたい場合など、この時点で適切なコンソール変数を設定します。詳細については、2.4.1 項を参照してください。

2. システムをブートして、シングルユーザ・モード (スタンドアロン・モード) にします。

```
>>> boot
```

3. システムでスーパーユーザ (#) プロンプトが表示されたら、次のようなコマンドを使ってファイル・システムを書き込み可能でマウントします。

```
# mount -u /
```

ファイル・システムを書き込み可能でマウントすることにより、`ed` ライン・エディタを使用して、システム・ファイルを編集することができます。また、コマンドやユーティリティにアクセスすることができます。`vi` などの他のエディタは、ルート・ファイル・システム (/) には存在してないため、ここでは使用できません。

4. `/etc/rc.config`、`/etc/rc.config.common` および `rc.config.site` ファイルを、次のように、破損に備えてコピーしておきます。

```
# cp /etc/rc.config /etc/orig_rc.config
# cp /etc/rc.config.common /etc/orig_rc.config.common
# cp /etc/rc.config.site /etc/orig_rc.config.site
```

注意

スタートアップ操作とシステム構成では、ファイル `/etc/rc.config`、`/etc/rc.config.common` および `/etc/rc.config.site` の一貫性が確保されていなければなりません。これらのファイルは、`rcmgr` コマンド以外では修正しないでください。ファイルの形式が正しくない場合は、他のサブシステムやユーティリティでこれらのファイルが正しく解析されない可能性があります。詳細は、`rcmgr(8)` を参照してください。クラスタ・メンバでのブート操作の実行については、TruCluster のマニュアルを参照してください。

5. `rcmgr` ライン・エディタを使用して、ネットワーク・サービスを呼び出す構成ファイル・エントリを変更します。たとえば、Network Information Service (NIS) をテストしてオフにする場合は、次のようなコマンドを入力します。

```
# rcmgr get NIS_CONF
YES
# rcmgr set NIS_CONF NO
```

この操作を、NTP または NFS などの、起動済みのネットワーク・システムごとに繰り返します。

6. 変更し終わったら、システムを停止して、次のようにすべてのコンソール環境変数を再設定します。

```
>>> set boot_osflags a
>>> boot
```

システムは、ネットワーク・サービスを開始せずに、マルチユーザ・モードでリブートします。

システム・モデルとファームウェア・リビジョンによって、コンソール・コマンドには多少の違いがあります。ご使用中のプロセッサのコンソール・コマンドについては、ハードウェアのドキュメントを参照してください。

2.4 システムのブート

カーネルをブートするために使用するコマンドは次の要素により異なります。

- プロセッサ・タイプ
- 実行レベル

- ブートするカーネルの位置 (システム・ディスクかリモート・サーバか)
- すべてのプロセッサをブートするのか単一プロセッサのみをブートするのか (マルチプロセッサ・システムの場合)
- コンソール環境変数が定義されているかどうか
- 省略時のカーネルをブートするのか代替カーネルをブートするのか

2.4.1 コンソール環境変数の設定とブート・コマンドの使用

この項では、典型的なコンソール設定の例を示します。詳細情報については、システムに付属のハードウェア・ドキュメントを参照してください。システムのブートについては、『インストール・ガイド』および『インストール・ガイド — 上級ユーザ編』も参照してください。

RAID ストレージ・アレイ、または記憶領域ネットワーク内のファイバ・チャネル・コントローラを使う場合は、適切なストレージ管理ソフトウェアを使ってブート・デバイス情報を取得し、設定しなければなりません。ストレージ・アレイのドキュメントを参照してください。

システムをブートする場合、コンソール環境変数の使用方法と、ブート処理での役割について理解しておく必要があります。表 2-1 に、コンソール環境変数とその機能を示します。

表 2-1: コンソール環境変数

| 変数 | 機能 |
|---------------------------|-----------------------------------|
| <code>boot_reset</code> | on に設定するとハードウェアがリセットされる。 |
| <code>boot_osflags</code> | ブート・ローダとカーネルを制御するために使用されるフラグの組合せ。 |
| <code>bootdef_dev</code> | ブート・デバイス |
| <code>boot_file</code> | ブートするカーネル。 |
| <code>cpu_enable</code> | 特定のプロセッサをコンソールから選択的に有効にする。 |

ブート操作のためにハードウェアを準備するには、コンソール・プロンプトで次の操作を実行します。

1. `auto_action` 変数を `halt` に設定します。

```
>>> set auto_action halt
```

このコマンドを実行すると、システムがクラッシュした場合、あるいは停止ボタンを押した場合に、コンソール・プロンプトを表示してシステムは停止します。

- 2. ご使用のプロセッサで必要ならば、`boot_reset` 変数を `on` に設定して、ブートする前に強制的にハードウェアをリセットします。

例を次に示します。

```
>>> set boot_reset on
```

- 3. ご使用のプロセッサで必要ならば、SCSI デバイスをリセットするための待ち時間を設定します。

```
>>> set scsi_reset 4
```

- 4. 次の手順で、`boot_osflags` 変数とブート・デバイスを設定します。

- a. `boot_osflags` 変数に設定するオプションを決めます。

使用できるオプションを表 2-2 に示します。

表 2-2: `boot_osflags` 変数のオプション

| オプション | アクション |
|-------|---|
| a | マルチユーザ・モードでブートする。省略時の設定では、カーネルはシングルユーザでブートする。 |
| k | <code>kdebug</code> デバッガを使用して、カーネルをデバッグする。詳細は、『 <i>Kernel Debugging</i> 』を参照。 |
| d | フル・クラッシュ・ダンプを使用する。省略時の設定では、部分ダンプを使用する。クラッシュ・ダンプについての詳細は、第 12 章を参照。 |
| i | カーネルと特別な引数の入力を促すプロンプトを表示する。省略時の設定では、プロンプトは表示されない。対話式ブートの例は、2.4.3 項を参照。 |

これらのオプションは、`boot_osflags` 変数に続けて指定することで、効果を持ちます。たとえば、マルチユーザ・モードでブートしフル・クラッシュ・ダンプを使用する場合は、次のコマンドを入力します。

```
>>> set boot_osflags ad
```

省略時の値を設定する場合は、次のように入力して変数をクリアします。

```
>>> set boot_osflags ""
```

- b. 次のコマンドを入力して、システム・デバイスのユニット番号を調べます。

```
>>> show device
```

- c. 省略時のブート・デバイスを設定します。

システムをブートする際にはブート・デバイスを指定しなければなりません。常に同じデバイスからブートする場合は、次のコマンドで `bootdef_dev` 変数に省略時のブート・デバイスを設定することができます。たとえば、ディスク `dka000` からシステムをブートするためには、次のコマンドを入力してください。

```
>>> set bootdef_dev dka000
```

ハードウェア構成には、デュアル KZPBA-CB バスに接続され、マルチバス・フェイルオーバー用に構成される HSZ コントローラを含めることができます。この場合、`bootdef_dev` コンソール変数の設定では、ブート・ディスク装置へのバス・パスを両方指定します。デュアル・コントローラ・システムの構成時には、その片方のコントローラを優先バスとして指定します。このコントローラ上のブート・デバイスは、`bootdef_dev` コンソール変数の 1 番目の引数として指定します。

たとえば、システムに A と B の 2 つのコントローラがあり、4 つの論理ボリューム `dka0`、`dka1`、`dkb0`、および `dkb1` に接続されている場合、コントローラ B を優先コントローラとして指定するには、次のように、`**b*` デバイスを 1 番目の引数として `bootdef_dev` コンソール変数に指定します。

例:

```
>>> set bootdef_dev dkb0.0.0.0.6.0, dka0.0.0.5.0
```

各デバイス・パスは、コンマで区切ります。スペースやタブ文字は使用しないでください。1 番目のデバイスからコンソールをブートできない場合、次のデバイスからのブートが試みられます。

- d. 代替カーネルからブートする場合は次のコマンドを入力します。

```
>>> set boot_osflags i
```

システムは、ブート時にカーネルへのパスの入力を促します。次に例を示します。

```
Enter [kernel_name] [option_1 ... option_n]: genvmunix
```

システムは情報メッセージを表示します。

プロセッサによっては、`boot_file` 変数にブートしたいカーネル名を設定することによって、代替カーネルをブートすることができます。たとえば、汎用カーネル (`genvmunix`) をブートするには次のコマンドを入力します。

```
>>> set boot_file genvmunix
```

ご使用のプロセッサによっては、省略時カーネル (`/vmunix`) をブートしたい場合、`boot_file` 変数を次の例のようにクリアする必要があります。

```
>>> set boot_file ""
```

マルチプロセッサ・システムの場合、`set cpu_enable` コマンドを使用して、有効にするプロセッサをコンソールで設定することが可能です。コンソールから選択的に各プロセッサを有効にすることができます。各ビットがスロット番号を表すようなビット・フィールドをマスクと呼びます。CPU マスクを `ff` に設定することによって、すべてのプロセッサを有効にすることができます。マスクを設定した後、システムの電源を切り、再度電源をいれてください。

Tru64 UNIX オペレーティング・システムは、各プロセッサをブート時に有効/無効にするためのメカニズムも提供します。詳細については、『システムの構成とチューニング』の `cpu-enable-mask` 属性に関する説明を参照してください。

コンソール変数を設定したら次のコマンドを実行してシステムをブートします。

```
>>> b
```

2.4.2 ブート・コマンドの変更

次に、2.4.1 項に示したコマンドを変更する方法について説明します。

- `bootdef_dev` コンソール変数の変更

`bootdef_dev` コンソール変数を無視させるには、ブート・コマンドへの引数としてブート・デバイスを指定します。たとえば、ブート・デバイスとしてディスク `dka0` が設定されている場合に、ディスク `dkb0` からブートしたい場合には次のコマンドを入力します。

```
>>> b dkb0
```

- `boot_osflags` コンソール変数の変更

次のようにブート・コマンドに `-fl` オプションを指定すると、`boot_osflags` 変数は無視されます。

```
>>> b -fl
```

`boot_osflags` 変数無視させるには、選択内容を `-fl` オプションと一緒に指定します。たとえば、次のコマンドでは、代替カーネルを指定できるように対話モードでブートして、その後マルチユーザ・モードでブートします。

```
>>> b -fl ai
```

オプションの一覧は、表 2-2 を参照してください。対話式ブート・セッションの例は、2.4.3 項を参照してください。

- `boot_file` コンソール変数の変更

`boot_file` コンソール変数で指定したカーネル以外をブートするには、カーネル・ファイルへのパスを指定します。たとえば、汎用カーネル (`/genvmunix`) をブートするには、次のコマンドを入力します。

```
>>> b -fi genvmunix
```

2.4.3 対話式ブートを使うルート・ファイル・システムの検証

コンソールのブート・コマンドで `-flags i` オプションを使って、対話式ブート・セッションを起動します。システムで利用できるコンソール・コマンド・オプションによっては、他のブート・オプションとパラメータを `-i` オプションを使って入力することができます (対話式ブート・オプションの情報については、プロセッサのオーナーズ・マニュアルを参照してください)。

対話式ブート・セッションでは、ルート・ファイル・システム (`/`) にある `osf_boot` コマンドが実行されます。このコマンドを使用すると、システムを完全にブートせずにルート・ファイル・システムを検査することができます。次の手順を使ってこの作業を行います。この手順では、システムがシャットダウンされていて、コンソール・プロンプトの状態であることを前提としています。

1. コンソール・プロンプト (`>>>`) で次のコマンドを入力し、対話式モードでシステムをブートします。

```
>>> boot -flags i
```

2. 次のメッセージが表示されます。

```
UNIX Boot - date
```

```
Enter: <kernel_name> [option_1...option_n]  
or: ls [name]['help'] or quit to return to console  
Press return to boot 'vmunix'#  
#
```

この時点でのオプションは次のとおりです。

- a. 代替カーネルの名前を入力し、必要なブート・オプションを指定します。使用しているシステムのブート・オプションの一覧は、オーナーズ・マニュアルを参照してください。

- b. 次のコマンドを入力して、ls コマンドのヘルプを参照します。

```
# help
```

ls コマンドのオプションは、ステップ 3 で説明しています。

- c. quit コマンドを入力するとコンソール・プロンプトに戻ります。

- d. boot_file コンソール変数で他のカーネルが指定されていない場合は、リターンを押して省略時のカスタム・カーネル (/vmunix) をブートします。

- 3. ls コマンドを使い、ルート・ファイル・システムのディレクトリの内容をリストするか、特定のファイルをリストします。ファイル名を指定しない場合は、ディレクトリの内容全部が表示されます。次に、有効なコマンドの例を示します。

- このコマンドは、最上位レベルのルート・ディレクトリ (/) の内容をすべてリストします。

```
# ls /
```

- コンソールに表示を行っていて、他のコマンドが使えない場合は、表示出力を制御することができないため、出力がスクロールされて画面から消えることがあります。このような場合は、疑問符 (?) やアスタリスク (*) のワイルドカード文字による文字や文字列の照合を使って、表示される量を少なくしてください。たとえば次のようにします。

```
# ls /etc/*rc*
```

このコマンドは、/etc/rc.config のように、文字列 rc と一致する、/etc ディレクトリ内のファイルすべてを返します。

ワイルドカード文字は、ファイル名に対してはサポートされていますが、ディレクトリ名に対してはサポートされていません。

2.5 システム実行レベルの識別

実行レベルはシステムの状態を指定するものであり、それぞれの実行レベルでどのプロセスの実行が許されるかを定義します。一般的に使用される実行レベルは、次のとおりです。

| 実行レベル | システムの状態 |
|---------|--------------------------|
| 0 | 停止状態 |
| s または S | シングルユーザ・モード |
| 2 | ネットワーク・サービスなしのマルチユーザ・モード |
| 3 | ネットワーク・サービス付きのマルチユーザ・モード |
| null | コンソール・モード |

inittab ファイルには、特定の実行レベルを定義する行エントリと、その実行レベルに対応するコマンド実行スクリプトが入っています。init プロセスは実行を開始されると、inittab ファイルを読み取り、適切なコマンド起動スクリプトを実行します。このスクリプトが特定の実行レベルで起動するプロセス（および、システムがあるレベルから別のレベルに変わった場合に強制終了させるプロセス）を定義します。inittab ファイルの見方と変更方法についての詳細は、init(8) と inittab(4) および第 3 章を参照してください。

2.6.2 項では、init コマンドを使用して実行レベルを変更する方法について説明します。

2.6 システム実行レベルの変更

新しい実行レベルに変更する前に、inittab ファイルを見て、変更する実行レベルに必要なプロセスがサポートされていることを確認してください。特に重要なのは getty プロセスです。このプロセスはコンソールおよびその他のログイン・ユーザの端末回線アクセスを制御します。inittab ファイルの getty エントリで、すべての実行レベルでシステム・コンソール・アクセスが許されていることを確認してください。実行レベルの定義についての詳

細は、`inittab(4)` を参照してください。端末回線とアクセスの定義についての詳細は、`getty(8)` を参照してください。

実行レベルを変更するとユーザの `getty` プロセスが終了し、ログイン機能が使用不能になり、他のユーザ・プロセスが終了することがあります。新しい実行レベルに変更する前に、`wall` または `write` コマンドを使用して、実行レベルを変更することを各ユーザに警告してください。

ユーザ端末の `getty` エントリを検査して、新しい実行レベルがエントリに指定されていることを確認してください。指定されていない場合は、`init` プロセスからの `kill` シグナルに応答してプロセスが終了しないように、そのユーザにログオフするよう促します。

システムが初めて初期化される場合には、`inittab` ファイル内の `initdefault` 行エントリによって定義されている省略時の実行レベルに入ります。`init` プロセスが実行レベルを変更するためのシグナルを受信するまで、システムはその実行レベルのままです。次の項で、これらのシグナルについて説明し、実行レベルを変更するためのコマンドを示します。

2.6.1 シングルのユーザ・モードでの実行レベルの変更

シングルのユーザ・モードで作業している場合は、`Ctrl/d` を押して実行レベルを変更します。`Ctrl/d` を押すと、シェルが終了して次のメッセージが表示されます。

```
INIT: New run level: 3
```

このメッセージが一般的に見られるのは、ブート操作中にシングルのユーザ・モードからマルチユーザ・モードに変わるときです。その他の場合、実行レベルの入力を促すプロンプトが表示されます。実行レベルの変更についての詳細は、`init(8)` を参照してください。

`init` プロセスは `inittab` ファイルを検索して `boot` または `bootwait` キーワードのあるエントリを探し、これらのエントリに従って動作した後、新しい実行レベルでの実行が許可されているその他のエントリを実行します。

2.6.2 マルチユーザ・モードからの実行レベルの変更

ネットワークなし、ネットワーク付きの 2 つのマルチユーザ実行レベルのうちのいずれかで作業している場合には、`init` コマンドを使用して次のように実行レベルを変更することができます。

| 実行レベル | システムの状態 |
|-------------|---|
| 0 | 停止状態 |
| 2 | ローカル・プロセスとデーモンのあるマルチユーザ実行レベル |
| 3 | リモート・プロセスとデーモンのあるマルチユーザ実行レベル |
| 1, 4, 5 ~ 9 | /etc/inittab ファイル内の番号フラグで指定された実行レベルになる。このようなエントリが存在しない場合は、何も実行されず、メッセージも表示されない。 |
| M, m | 制御がコンソール・デバイスに移され、停止してシングルユーザ・モードになる。 |
| Q, q | init プロセスによる inittab ファイルの再調査が必要であることを指定。 |
| S, s | 実行レベルが、必須のカーネル・サービスのみのシングルユーザ状態に変更される。 |

2.6.2.1 別のマルチユーザ実行レベルへの変更

現在のマルチユーザ実行レベルから別のマルチユーザ実行レベルに変更するには、移りたい実行レベルに対応する引数を付けて `init` コマンドを入力してください。たとえば、実行レベル 2 から実行レベル 3 に変更するには、次のコマンドを入力します。

```
# init 3
```

`init` プロセスは `inittab` ファイルを読み取り、新しい実行レベルへの変化に対応する命令に従います。

2.6.2.2 シングルユーザ・モードへの変更

`init` コマンドで実行レベル引数 `s` を使用して現在のマルチユーザ・モードからシングルユーザ・モードに変更することができます。たとえば、現在の実行レベルからシングルユーザ・モードに変更するには、次のコマンドを入力してください。

```
# init s
```

ユーザに 10 分の警告期間を与えてマルチユーザ・モードからシングルユーザ・モードに変更するには、次のコマンドを入力します。

```
# /usr/sbin/shutdown +10 Bringing system down to single-user for testing
```

シングルユーザ・モードからマルチユーザ・モードに戻るには、`Ctrl/d` を入力するか、プロンプトで `exit` コマンドを入力してください。これにより、

`init` コマンドは、実行レベルの入力待ち状態になります。ネットワーク・デーモンを起動しないでマルチユーザ・モードに戻るには、プロンプトに対して `2` を入力します。ネットワーク・デーモンを起動してマルチユーザ・モードに戻るには、`3` を入力します。

別の方法として、次のコマンドのいずれかを使用して、システムをリブートすることができます。

```
# /usr/sbin/shutdown -r now
```

または

```
# /sbin/reboot
```

2.6.2.3 `inittab` ファイルの再読み取り

`inittab` ファイルを再度読み取るには、次のように、`q` 引数を付けて `init` コマンドを入力します。

```
# init q
```

`init` プロセスは `inittab` ファイルを読み取り、必要なら新しいプロセスを開始します。たとえば、最近新しい端末回線を追加した場合は、`init` プロセスは `init q` コマンドに応答してそれらの端末回線に対する `getty` プロセスを開始します。

端末回線と `init` コマンドの関係についての詳細は、`getty(8)` を参照してください。

2.7 シンメトリック・マルチプロセッシング

シンメトリック・マルチプロセッシング (SMP) は、オペレーティング・システムの同一のコピーを実行する複数のプロセッサとアドレス共通メモリによって実現され、命令を同時に実行することができます。マルチプロセッサ・システムでは、複数プロセッサの同時実行によって、複数のスレッドを同時に実行させることができます。

マルチプロセッサ・システムを使用し、そのシステムで Tru64 UNIX を実行していれば、そのシステムは SMP 環境で実行されています。SMP 環境のオペレーティング・システムでは、システムにプロセッサを追加することによって処理能力を増強することができます。この目的のためにオペレーティング・システムは、利用可能な複数のプロセッサで、複数のスレッドを同時に実行します。

2.7.1 既存システムへの CPU の追加

システムは、使用できる CPU の数をブート時に調べます。マルチプロセッサ・システムの計算能力を強化するには、システムにプロセッサ・ボードをインストールしてリブートします。カーネルを再構築する必要はありませんが、設定してあるプロセッサ数の制限を変更する必要がある場合があります。詳細は、『システムの構成とチューニング』を参照してください。また、PAK (製品登録キー) をインストールする必要がある場合は、『*Software License Management*』を参照してください。

2.7.2 マルチプロセッサ・システムでの自動リブート

マルチプロセッサ・システム内のシステムに障害が発生した場合、オペレーティング・システムは障害の発生したプロセッサを記録した後、自動的にそのシステムをリブートします。オペレーティング・システムは処理を続行しますが、システム管理者は障害の発生したシステムを手動で再起動しなければなりません。手順については『インストレーション・ガイド』を参照してください。

2.8 システム・クロックの設定および再設定

システムには、システムをインストールする際に設定する内部時計があります。システム・クロックは、電源がオンであるかどうかに関係なく、時刻と日付を常に保守します。しかし、時刻または日付の再設定が必要な場合があります。たとえば、バッテリーで電力を供給されるクロックの場合、バッテリー障害が発生すると時刻を再設定する必要がでてきます。または、システム時間と標準時の同期化が必要なこともあります。

日付と時刻を設定するには、スーパーユーザとしてログインし、date コマンドを使用します。日付パラメータと時刻パラメータのシーケンスは、どのコマンド・オプションを使用するかによって異なります。(詳細については、date(1) を参照してください。) 表 2-3 にパラメータの値を示します。

表 2-3: date コマンドのパラメータ

| パラメータ | 説明 |
|-------|-------------------------------|
| CC | 年の最初の 2 桁 (世紀) を 2 桁の整数で指定する。 |
| YY | 2 桁の整数で年を指定する。 |
| MM | 2 桁の整数で月を指定する。 |

表 2-3: `date` コマンドのパラメータ (続き)

| パラメータ | 説明 |
|-----------|------------------------------|
| <i>dd</i> | 2 桁の整数で日を指定する。 |
| <i>HH</i> | 2 桁の整数で時間を 24 時間単位で指定する。 |
| <i>mm</i> | 2 桁の整数で分を指定する。 |
| <i>.</i> | デリミタ。 |
| <i>ss</i> | 2 桁の整数で秒を指定する。このフィールドはオプション。 |

たとえば、`[[cc]yy]mmddHHMM[.ss]` のフォーマットを使用して、日付を 2002 年 9 月 7 日の午前 9 時 34 分 00 秒に設定するには、次のいずれかのコマンドを入力します。

```
# date 090709342002
# date 0907093402.00
# date 090709342002.00
```

年のデータを変更する場合、新しいデータでシステム・ディスクを更新する必要があります。新しい年のデータを含む日付データを入力した後、シングルユーザ・モードで `mount -u /` コマンドを入力します。このコマンドにより、新しい年のデータがシステム・ディスクのスーパーブロックに書き込まれます。ルート・ファイル・システムは、読み取り/書き込み可能でマウントされます。

2.9 ブート時の問題のトラブルシューティング

システムがブートできない場合に詳しく調査を行うための手順を、次に示します。

2.9.1 ハードウェア障害

ハードウェア・テストの手順については、システムに付属のハードウェア・マニュアルを参照してください。ハードウェアの問題が存在する場合は、ハードウェア・マニュアルの問題解決のための手順に従ってください。

2.9.2 ソフトウェア障害

ブートできない原因として次のようなことが考えられます。

- 間違ったブート・パスの指定

正しいブート・パスの指定については、2.4 節またはシステムのハードウェア・マニュアルを参照してください。

- カーネルの破壊

カーネルが破壊されている疑いがある場合は、汎用カーネル `/genvmunix` をブートしてください。このカーネルをブートすると、完全に機能するシステムになり、`kdbx` または `dbx` ユーティリティを使用してデバッグを開始することができます。`kdbx` および `dbx` ユーティリティについての詳細は、`kdbx(8)` または `dbx(1)` を参照してください。代替カーネルのブートについては、2.4.1 項を参照してください。

- ディスクまたはファイル・システムの破壊

ディスクまたはファイル・システムが破壊されている場合は、`fsck` コマンドを実行してください。`fsck` コマンドは UNIX ファイル・システム (UFS) を検査して、修復します。`fsck` プロセスは、問題を発見すると、リカバリ・オプションの選択を求めてきます。このような場合には、ファイルを誤って重ね書きしたり削除したりしないように十分に注意してください。このコマンドの使用方法的詳細は、`fsck(8)` を参照してください。AdvFS (Advanced File System) を使用している場合、ディスクの破壊は非常に起こりにくくなります。

AdvFS はマウント・プロシージャ実行中にディスク構造を訂正し、ディスクの回復を行います。また `fsck` 等のコマンドを実行する必要がないので AdvFS ファイルの回復は非常に高速です。詳細については、『*AdvFS 管理ガイド*』を参照してください。

2.10 システムのシャットダウン

以降の各節では、シャットダウン手続きと、システム管理者による正常なシャットダウンおよび予期しないシャットダウンの両方で使用する回復方法を説明します。最初の節では、制御下においてシャットダウンを行う手順について説明し、2 番目の節では、予期しないシャットダウンに対処して回復するための手順とガイドラインについて説明します。

システムをシャットダウンする一般的な理由は次のとおりです。

- ソフトウェアをアップグレードする場合、または新しいハードウェアをシステムに追加する場合

システムをシャットダウンして、構成ファイルに対して必要な変更を行い、新しいカーネルを構築します。

- ハードウェア・エラー・ログに警告メッセージが記録されている場合
ハードウェアに障害が発生するおそれがあるため、システムをシャットダウンして原因を調査します。
- システムの性能が急速に低下している場合
システム統計値を検査して、性能を改善するためにシステムをシャットダウンして調整します。
- ファイル・システム破壊の可能性がある場合
`fsck` プログラムを実行して問題を修正するか、または問題が存在しないことを確認するためにシステムをシャットダウンします。
- 環境モニタリング・ユーティリティ、イベント・マネージャ (EVM) が、パラメータ値が超過し、障害発生の可能性あることを示している場合

上記のような状況、またはこれに類似した状況では、解決方法としてさまざまな選択肢があります。解決方法に関係なく、最初の手順はシステムのシャットダウンを行うことです。シングルユーザ・モードまたはマルチユーザ・モードからシステムをシャットダウンするための、実用的で合理的な方法がいくつかあります。

パニックまたはクラッシュしたシステムでは、正常にシャットダウンしたシステムとは違う状況が発生します。この章では正常なシャットダウンについてのみ解説します。システム・クラッシュについては、第 12 章を参照してください。

2.11 マルチユーザ・モードにおけるシステムの停止

マルチユーザ・モードで実行しているときにシステムをシャットダウンするには、`shutdown` コマンドを使用するか、SysMan Menu のタスク「システムのシャットダウン」を起動します。`-h` または `-r` フラグを指定して `shutdown` コマンドを入力すると、このプログラムは通常、次の処理を順に実行します。

1. `wall` プログラムを実行して、シャットダウンすることをすべてのユーザに通知します。
2. 新規ログインを禁止します。
3. すべての課金プロセスおよびエラー・ログ・プロセスを停止します。
4. `killall` プログラムを実行して、すべてのプロセスを停止します。

5. `sync` プログラムを実行して、ディスクを同期化します。
6. ログ・ファイルにシャットダウンのログを記録します。
7. ファイル・システムをディスマウントします。
8. システムを停止します。

次の項では、一般的なシャットダウン操作を説明し、コマンド・フラグを使用した場合の例を示します。詳細については、`shutdown(8)` を参照してください。

2.11.1 SysMan シャットダウンの使用

`sysman shutdown` コマンドを使って SysMan Menu のシャットダウン・タスクを呼び出します。このインタフェースは、SysMan Station または SysMan Menu から呼び出すことができます。SysMan Menu の「一般的なタスク」ブランチから「システムのシャットダウン」オプションを選択するなど、別の SysMan インタフェースを呼び出す方法については、第 1 章を参照してください。

`sysman shutdown` を入力すると、「シャットダウン: *host name*」というタイトルのウィンドウが表示されます。*host name* には、ローカル・システム名が表示されます。クラスタ・メンバをシャットダウンする場合、シャットダウン・タスクには追加のオプションがあります。クラスタのメンバを 1 つ以上シャットダウンする場合は、TruCluster のドキュメントを参照してください。

次のオプションを使用できます。

| | | |
|---------------|---|--|
| 「シャットダウンのタイプ」 | このオプション・メニューを使用して、次のシャットダウン・オプションのいずれかを選択します。 | |
| 「Halt」 | オペレーティング・システムを停止し、コンソール・プロンプトを表示する。 | |
| 「リブート」 | システムをシャットダウンして停止してから、自動的にリブートする。 | |

| | |
|------------------------------|--|
| 「シングルユーザ」 | シングルユーザ・モードにシャットダウンして、スーパーユーザ・プロンプト(＃)を表示する。 |
| 「メッセージのみ」 | 現在のシステム・ユーザすべてに、メッセージをブロードキャストする。システムはシャットダウンしません。 |
| 「シャットダウンまでの時間(分)」 | マウス・ボタン 1 (MB1) を押したままスライダ・バーを移動して、シャットダウン動作が始まるまでの時間 (シャットダウン遅延) を分単位で指定します。時間はこのバーの横に表示されます。このスライダ・バーを使用すると、0 ~ 60 分の範囲から選択できます。文字セル端末を使用している場合など、ユーザ環境によってはスライダ・バーが使用できないことがあります。この場合は、数値を入力してシャットダウン遅延を指定します。このようなインタフェースでは、60 分を超える時間を指定できます。 |
| 「シャットダウン・メッセージ」 | <p>すぐにシャットダウンすることを警告してログアウトするように要求する、ユーザへのメッセージを入力します。このメッセージは、省略時のメッセージに追加して表示されます。</p> <p>すぐにシャットダウンしない場合は、シャットダウン要求時と、それ以降定期的に、メッセージが表示されます。たとえば、55 分後のシャットダウンを要求した場合、メッセージは、シャットダウンの 55 分、50 分、40 分、30 分、20 分、10 分、5 分、1 分、30 秒前と、シャットダウン時に表示されます。</p> |
| 「NFS クライアントへのブロードキャスト・メッセージ」 | このボックスをチェックすると、ローカル NFS ファイル・システムのリモート・ユーザにメッセージがブロードキャストされます。このローカル・システムがエクスポートしたファイル・システムに接 |

続しているすべてのリモート・ユーザは、すぐにシャットダウンするという警告を受け取ります。このようなメッセージを送信するには、リモート・ユーザのシステムで `rwalld` デーモンが動作していなければなりません。

「ラン・レベル移行
スクリプトの実行」

このボックスをチェックすると、
`/sbin/rc[N.d]/[Knn_name]` ファイルにある
既存の実行レベル移行スクリプト (たとえば、
`/sbin/rc0.d/K45.syslog`) を実行できます。詳細は、`shutdown(8)` で、`-s` オプションを参照してください。

「プレシャットダウン・スクリプト」

シャットダウンが完了する前に実行したいカスタム・スクリプトへのパスを指定します。このスクリプトはシャットダウン時に実行され、シャットダウンが完了するまでに、指定されたすべてのタスクが完了します。このスクリプト (または、このスクリプトが呼び出す中間スクリプト) が正常に終了しないと、システムのシャットダウンが正しく行われないことがあります。

「その他のオプション」

このオプションをチェックすると、システムのシャットダウンを高速化する次のオプションを指定することができます。

「高速」

ユーザおよび NFS クライアントへのメッセージは省略し、短時間でシャットダウンする。

「ディスクの Sync を
実行しない」

`sync` 操作でディスクの同期をとらずにシャットダウンする。

`SysMan Menu` を使用してシャットダウンを開始すると、システムは、
2.11.2 項の例 2-1 に示したようにシャットダウンします。ただし、「シャッ

トダウン: カウントダウン」ウィンドウにシャットダウンまでのカウントダウンが表示されます。シャットダウンは、いつでも取り消すことができます。

各種オプションについては、オンライン・ヘルプを参照してください。また、シャットダウン・コマンドの動作については、shutdown(8) を参照してください。

2.11.2 システムのシャットダウンとユーザへの警告

この作業は、shutdown コマンドを使用するか、SysMan Menu の「システムのシャットダウン」タスクを起動して行うことができます。

特定の時刻にマルチユーザ・モードからシングルユーザ・モードへシステムをシャットダウンし、ユーザにただちにシャットダウンする警告を行うには、次の手順に従ってください。

1. スーパユーザとしてログインしてルート・ディレクトリに移ります。

```
# cd /
```

2. shutdown コマンドを使ってシャットダウンを起動します。保守作業のためにシステムがシャットダウンされることをユーザに警告し、たとえば、10 分後にシステムをシャットダウンして停止する場合は、次のコマンドを入力してください。

```
# /usr/sbin/shutdown +10 "Planned shutdown, log off now"
```

例 2-1 は、典型的なシャットダウン・シーケンスです。

例 2-1: 典型的なシャットダウン・シーケンス

```
# /usr/sbin/shutdown +6
"Maintenance shutdown, please log off" [1]
System going down in 6 minutes
    ...Maintenance shutdown, please log off [2]
System going down in 5 minutes
    ...Maintenance shutdown, please log off [3]

No Logins, system going down @ <time>
    ...Maintenance shutdown, please log off [4]

System going down in 60 seconds
    ...Maintenance shutdown, please log off
System going down in 30 seconds
    ...Maintenance shutdown, please log off
System going down immediately
```

例 2-1: 典型的なシャットダウン・シーケンス (続き)

```
...Maintenance shutdown, please log off[5]
.
. process shutdown messages [6]
.
Halting processes ...
INIT: SINGLE USER MODE[7]
# halt
.
. <hardware reset messages> [8]
.
resetting all I/O buses
>>>[9]
```

- [1] このコマンドでは、6 分後にシャットダウンを開始します。すべてのユーザに、ログオフするように警告をブロードキャストします。
- [2] このメッセージは、コンソール端末と、shutdown コマンドを呼び出した端末ウィンドウにすぐにエコーされます。
- [3] これらのメッセージは、コンソール端末と、shutdown コマンドを呼び出した端末ウィンドウにすぐにエコーされます。メッセージは、当初のシャットダウン遅延の値に応じて繰り返されます。この繰り返しの間隔は、シャットダウン時間が近づくほど短くなります。
- [4] 5 分前になると、ログインできなくなります。誰かがログインしようすると、このメッセージがログイン端末に表示されます。他のユーザにはブロードキャストされません。
- [5] この最終メッセージは、システムが直ちにシャットダウンされ、ユーザ・プロセスが停止されることを警告します。システムは、課金やエラー・ログなどのプロセスを停止し、シャットダウンをログ・ファイルに記録します。次に、init プログラムにシグナルを送信し、システムをシングルユーザ・モードに移行させます。

シャットダウン遅延を指定しなかった場合 (shutdown now) は、システムのシャットダウン開始およびユーザ・プロセスの強制終了前に、このメッセージだけがブロードキャストされます。
- [6] プロセスが停止すると、通知メッセージがコンソールに表示され、記録されます。

- ⑦ システムが停止すると、すべてのログイン端末 (または CDE や XDM などのグラフィック・ディスプレイ) は停止し、出力はコンソールにリダイレクトされます。プロセスがシャットダウンされるたびにさまざまなメッセージがコンソールに表示されます。最終的にはシングルユーザ・モードになり、スーパーユーザ・プロンプト (#) が表示されます。この状態でシステムを使用できるのは root ユーザだけで、スタンドアロンでタスクを実行するか、halt コマンドを使用してシステムを完全にシャットダウンします。
- ⑧ システムの構成要素が初期化される間、さまざまなメッセージが表示されます。
- ⑨ コンソール・プロンプト (>>>) が表示されます。これで、システムの電源を切るか、システムをリブートするか、コンソール・コマンドを入力することができます。

2.11.3 システムのシャットダウンと停止

マルチユーザ・モードからシステムをシャットダウンして、すべてのユーザに警告し、すべてのシステムを停止するには、この手順を使用します。また、SysMan Menu の「システムのシャットダウン」タスクを呼び出して同じ操作を行うこともできます。

1. スーパーユーザとしてログインしルート・ディレクトリに移ります。

```
# cd /
```

2. shutdown コマンドを使ってシステムをシャットダウンし、停止させます。たとえば、保守のためにシステムを停止することをユーザに警告して、5 分後にシステムをシャットダウンして停止させるには、次のコマンドを入力してください。

```
# shutdown -h +5 /  
Maintenance shutdown in five minutes
```

システムは、例 2-1 に示したようにシャットダウンを開始します。ただし、システムも自動的に停止し、スーパーユーザ・プロンプト (#) では止まりません。コンソール・プロンプトが表示され、電源を切るか、リブートするか、システムのオーナーズ・マニュアルに示されているコンソール・コマンドを使用することができます。

2.11.4 システムのシャットダウンと自動リブート

マルチユーザ・モードからシステムをシャットダウンして、すべてのユーザに警告し、自動的にリブートしてマルチユーザ・モードに戻すには、この手順を使用します。また、SysMan Menu の「システムのシャットダウン」タスクを呼び出して同じ操作を行うこともできます。

1. スーパユーザとしてログインしルート・ディレクトリに移ります。

```
# cd /
```

2. `shutdown` コマンドを使ってシャットダウンを起動し、その後自動的にリブートします。たとえば、リブートのためにシステムがダウンすることをユーザに警告して、システムを 15 分後にシャットダウンし、その後自動的にリブートするには、次のコマンドを入力してください。

```
#shutdown -r +15 \  
Shutdown and reboot in 15 minutes
```

システムは、例 2-1 に示したようにシャットダウンを開始します。このとき、システムはすぐにシャットダウンすることをユーザに通知し、ログインを使用不可にして、通常のシャットダウン動作を実行します。これらの処理が終了すると、システムは自動的にリブートします。リブート時に、マウントされているすべてのファイル・システムについて整合性検査を行うための `fsck` コマンドが自動的に実行されます。問題がない場合は、システムはマルチユーザ・モードにリブートされます。

注意

`fsck` コマンドは、ファイル・システムの矛盾を発見すると、再度シングルユーザ・モードから `fsck` コマンドを実行してからマルチユーザ・モードでシステムを動作させることを勧める警告メッセージを表示します。

2.11.5 即時シャットダウンとシステムの停止

システムを即座にシャットダウンし停止するには、次の手順を実行します。また、SysMan Menu の「システムのシャットダウン」タスクを呼び出して同じ操作を行うこともできます。

1. スーパユーザとしてログインしルート・ディレクトリに移ります。たとえば、次のコマンドを入力します。

```
# cd /
```

2. 次の例のように、shutdown コマンドを入力します。

```
# shutdown -h now
```

システムは、例 2-1 に示したようにシャットダウンを開始します。ただし、シャットダウンはすぐに行われ、ユーザへの警告はありません。すべてのプロセスがシャットダウンされると、システムは停止し、コンソール・プロンプト (>>>) が表示されます。これで、システムの電源を切るか、システムをリブートするか、システムのオーナーズ・マニュアルに示されているコンソール・コマンドを入力することができます。

注意

この shutdown コマンドは、システムに他のユーザがログインしていない場合か、緊急にシャットダウンする必要がある場合にのみ使用してください。ユーザ・プロセスは警告なしに停止され、ユーザのデータが失われることがあります。

2.12 シングルユーザ・モードでのシステムの停止

システムをシャットダウンするには shutdown コマンドが最適ですが、halt, fasthalt, fastboot, および reboot コマンドを使用してシステムを停止することもできます (ただし、お勧めはしません)。これらのコマンドは、シングルユーザ・モードでのみ使用してください。

シングルユーザ・モードで作業している場合は、次の手順を実行すればシステムを停止することができます。

```
# /sbin/sync  
# /sbin/sync
```



```
# /usr/sbin/halt
```

halt コマンドに応答して次のイベントが発生します。

- ログ・ファイルにシャットダウンのログを取る
- 実行中のプロセスをすべて強制終了する
- sync システム・コールを実行する
- すべての情報がディスクに書き込まれる
- システムを停止する

最低 2 回, sync コマンドを入力することで, メモリ内のすべてのデータが安全にディスクに書き込まれます。halt コマンドとそのフラグについての詳細は, halt(8) を参照してください。

これ以外の方法については, fasthalt(8), fastboot(8), および reboot(8) を参照してください。

2.12.1 reboot コマンドでのシステムの停止とリブート

シングルユーザ・モードで作業している場合は, reboot コマンドを次のように使用して, システムを安全にシャットダウンしてマルチユーザ・モードでリブートすることができます。

```
# /usr/sbin/reboot
```

オプションを指定せずに reboot コマンドを実行すると, すべてのプロセスを停止し, ディスクを同期させ, リブートを開始して記録します。システムを緊急にシャットダウンしてリブートしたい場合は, 次のコマンドを入力します。

```
# reboot -q
```

このコマンドを入力すると, システムはプロセスを停止せず, 他のシャットダウン動作も行わないで, 緊急にシャットダウンします。このコマンドはイベントを記録せずに, リブートを開始します。このコマンドとフラグの詳細については, reboot(8) を参照してください。

2.12.2 fasthalt コマンドによるシステムの停止

シングルユーザ・モードで作業している場合は, fasthalt コマンドを次のように使用して, システムをただちに停止させることができます。

```
# /usr/sbin/fasthalt -n
```

オプションを指定せずに `fasthalt` コマンドを実行すると、システムが停止し、以降のリブートで `fsck` コマンドの実行が抑制されます。このプログラムは、`fastboot` ファイルを作成してから、`halt` プログラムを呼び出します。システム起動スクリプトには、`fastboot` ファイルを探す指示があります。このファイルが存在する場合、このスクリプトはこのファイルを削除し、`fsck` コマンドの実行が省略されます。このコマンドを、`-l`、`-n`、または `-q` フラグを指定せずに起動すると、`halt` プログラムは `syslogd` コマンドを使用してシャットダウンを記録し、このシャットダウンの記録をログイン課金ファイル `/var/adm/wtmp` に置きます。

詳細は、`fasthalt(8)` を参照してください。

2.12.3 `fastboot` コマンドによるシステムの停止

シングルユーザ・モードで作業し、ファイル・システムを検査する必要がある場合は、`fastboot` コマンドを次のように使用して、システムを停止しリブートすることができます。

```
# /usr/sbin/fastboot
```

オプションを指定せずに `fastboot` コマンドを実行すると、`/fastboot` というファイルが作成され、システムが停止された後、`fsck` コマンドを使ってファイル・システムを検査することなく、システムがすぐにリブートされます。詳細は、`fastboot(8)` を参照してください。

システム環境のカスタマイズ

この章では、システム環境をカスタマイズする方法を示します。これから説明する作業の中に、初期インストレーションとシステム構成の際にすでに行っているものもあるかもしれません。システムに変更が必要になった場合、新たな処理負荷の要求に対応させるため、この作業をいくつか行う必要があるかもしれません。たとえば、インストレーションの際に、初期スワップ領域(仮想メモリ)を作成しますが、物理メモリをシステムに追加する場合、それに対応して、スワップ領域を増やす必要が出てくる場合もあります。

この章では、以下の項目について説明します。

- システムの実行レベルを初期設定し、制御するために使用する、システム初期化ファイル (3.1 節)
- 各国語ディレクトリを使って言語および国に固有なプログラムをサポートする方法 (3.2 節)
- 多国語ユーザ対象のプログラムを開発および実行するプログラマとユーザをサポートする、国際化機能 (3.3 節)
- システムのローカルおよびワールドワイドの時間帯情報を管理する、システムの時間帯ディレクトリおよび環境変数 (3.4 節)
- Energy Star 規格に準拠した周辺機器およびプロセッサでの電力消費を制御するために設定して使用する、電力管理 (3.5 節)
- スワップ領域をカスタマイズする方法。『システムの構成とチューニング』も参照してください。性能をチューニングするために役立つ情報が得られます (3.6 節)。

セキュリティおよびネットワーク環境のカスタマイズについては、次のマニュアルを参照してください。

- 『Tru64 UNIX 概要』では、オペレーティング・システムのセキュリティ・コンポーネントについて簡単に説明しています。

- 『セキュリティ管理ガイド』と『セキュリティ・プログラミング・ガイド』は、セキュリティ・コンポーネントを扱うユーザ、管理者、およびプログラマにとって、中心となるセキュリティの情報源です。
- 『ネットワーク管理ガイド：接続編』および『ネットワーク管理ガイド：サービス編』は、システムのネットワーク・コンポーネントをカスタマイズする際に主要な情報源となります。

3.1 システム初期化ファイルの確認と変更

システム環境を定義したりカスタマイズする場合、プロセスおよび実行レベルの指定と制御を行う特定の初期化ファイルを変更します。オペレーティング・システムによって、有効な実行レベルと各実行レベルを指定するプロセスを定義した省略時のファイルがテンプレートとして提供されます。これらのファイルを使用することにより、システム環境を容易に変更またはカスタマイズすることができます。また、国際規格をサポートする場合は、それぞれの国際規格に対応するファイルの構造と条件に精通している必要があります。

この節では、システム環境の初期化および制御をするためのファイルについて紹介し、それらのファイルの使用方法および変更方法について説明します。利用できる機能を理解し、その機能を使用するためには、`init` プログラムおよびこのプログラムに関連するファイルおよびコマンドについて理解する必要があります。 `init` プログラムとその機能についての詳細は、`init(8)` を参照してください。

システム初期化ファイルを変更する前に、省略時の設定の確認、システムにおけるニーズの評価、および各テンプレート・ファイルのコピー作成などを行ってください。システム・ファイルまたは作業環境に影響するファイルを変更する場合には、問題が発生した場合の対応を考慮しておくことが必要です。初期化ファイルを変更しても意図した環境が設定されない場合は、カスタマイズに存在する問題を解決している間、省略時のファイルを元に戻しておくことができます。

以下に示すシステム・ファイルおよびディレクトリは、システムの初期化および操作に影響を与えます。

`/etc/inittab`

重要な初期化ファイルの1つで、実行レベルとそれに関連するプロセスを定義し、端末を管理します。このファイルについては、3.1.1 項で説明します。

3-2 システム環境のカスタマイズ

`/etc/securetty`

各端末 (tty) が root ログインできるかどうかを定義するテキスト・ファイル。このファイルについては、3.1.1.6 項で説明します。

`/sbin/bcheckrc`

ブート時のファイル・システムの検査とマウントを指定するシステム初期化コマンド実行スクリプト。このファイルについては、3.1.1.2 項で説明します。

`/sbin/init.d`

システムの初期化および有効な実行レベルに関連する実行可能ファイルを含んでいる初期化ディレクトリ。ディレクトリ構造と内容については、3.1.2.1 項で説明します。

`/sbin/rcn.d`

`/sbin` ディレクトリには、各実行レベルに対応する個々のサブディレクトリのセットが含まれます。個々のサブディレクトリには、特定の実行レベルを開始または変更するときにシステムが実行する、リンクされたファイルが含まれています。利用可能な `/sbin/rcn.d` ディレクトリには、`/sbin/rc0.d`、`/sbin/rc2.d`、および `/sbin/rc3.d` の 3 つがあります。`rc` ディレクトリ構造と内容については、3.1.2.2 項、3.1.2.3 項、および 3.1.2.4 項で説明します。

`/sbin/rcn`

特定の実行レベルに対応するコマンド実行スクリプトです。利用可能な `/sbin/rcn` スクリプトには、`/sbin/rc0`、`/sbin/rc2`、および `/sbin/rc3` の 3 つがあります。これらのスクリプトの内容と使用方法については、3.1.2.2 項、

3.1.2.3 項，および 3.1.2.4 項で説明します。

`/etc/rc.config` および
`/etc/rc.config.common`

実行時構成変数を含んでいるファイルです。これらの変数は，`/sbin/init.d` ディレクトリの中のスクリプトによって，さまざまなサブシステム（たとえば NFS や NTP）を構成するために使用されます。システム管理者またはプログラムは，`rcmgr` コマンドを使用して，`/etc/rc.config` ファイル内の変数を定義またはアクセスすることができます。詳細については，`rcmgr(8)` および『ネットワーク管理ガイド：接続編』と『ネットワーク管理ガイド：サービス編』を参照してください。

`/etc/sysconfigtab`

動的に構成できるサブシステムについての情報を含むデータベース・ファイルです。このファイルについては，第 4 章で説明します。

`/usr/sbin/getty`

端末回線の設定および管理のための実行可能ファイルです。このプログラムについては，3.1.1.3 項と 3.1.1.4 項で説明します。また，詳しい説明については，`getty(8)` を参照してください。

`/etc/gettydefs`

端末回線属性を設定および定義するエントリを含んでいるファイル。このファイルは `getty` によって使用されます。詳しい説明については，`gettydefs(4)` を参照してください。

`/var/spool/cron/crontabs/*`

特定プロセスの通常起動または周期的起動に関する設定を定義するエントリが含まれているファイルです。これらのファ

イルについての詳しい説明は、3.1.3 項を参照してください。

`/var/spool/cron/atjobs/*` 特定プロセスの一度限りの起動に関する設定を定義するファイルです。このファイルについての詳しい説明は、`at(1)` を参照してください。

次に示すファイルには、カーネル構成に関する情報が記述されています。

`/usr/sys/conf/NAME`

システムによってカーネル構成に組み込まれる構成要素を定義しているテキスト・ファイルです。通常、`NAME` 変数にはシステム名を指定します。このファイルについては、第 4 章で説明します。

`/usr/sys/conf/NAME.list`

レイヤード・プロダクトに関する情報の格納、および静的なサブシステムの自動的な構成のために使用されるオプションの構成ファイル。通常、`NAME` 変数にはシステム名を指定します。このファイルについては、第 4 章で説明します。

`/usr/sys/conf/param.c`

システムのカーネルの構築に使用される調整可能なシステム・パラメータの省略時の値を記述しているテキスト・ファイル。このファイルについては、第 4 章で説明します。

3.1.1 `/etc/inittab` ファイル

`init` プログラムが実行する最初の処理の 1 つは、`/etc/inittab` ファイルの読み取りです。`inittab` ファイルは、初期化プロセスの作成と実行のための命令を `init` プログラムに提供します。`init` プログラムは、実行されるたびに `inittab` ファイルを読み取ります。通常このファイルには、省略時の初期化命令、各実行レベルにおけるプロセスの作成および制御命令、端末回線の起動を制御する `getty` プロセスに対する命令が含まれています。

オペレーティング・システムは、一般的に初期化プロセスに必要な行エントリを含んだ `/etc/inittab` ファイルをテンプレートとして提供しています。

たとえば、次のような内容の `/etc/inittab` ファイルが提供されています。

```

is:3:initdefault: ss:Ss:wait:/sbin/rc0 shutdown </dev/console> \
/dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
kls:Ss:sysinit:/sbin/kloadsrv < /dev/console > /dev/console 2>&1
hsd:Ss:sysinit:/sbin/hotswapd < /dev/console > /dev/console 2>&1
sysconfig:23:wait:/sbin/init.d/autosysconfig start \
< /dev/console > /dev/console 2>&1
update:23:wait:/sbin/update > /dev/console 2>&1
smsync:23:wait:/sbin/sysconfig -r vfs smoothsync-age=30 > \
/dev/null 2>&1
smsyncS:Ss:wait:/sbin/sysconfig -r vfs smoothsync-age=0 > \
/dev/null 2>&1
it:23:wait:/sbin/it < /dev/console > /dev/console 2>&1
kmk:3:wait:/sbin/kmknod > /dev/console 2>&1
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
cons:1234:respawn:/usr/sbin/getty console console vt100

```

inittab ファイルは、任意の数の行で構成されます。inittab ファイルの各行は 4 つのフィールドで構成されています。各フィールドはコロン (:) で区切られています。inittab ファイルの各行の構文は次のとおりです。

Identifier: Runlevel: Action: Command

Identifier

このフィールドは、オブジェクト・エントリをユニークなものとして識別します。このフィールドは 14 文字です。

Runlevel

このフィールドは、オブジェクト・エントリが処理される実行レベルを定義します。このフィールドは 20 文字です。Runlevel 変数は、システム内のプロセスの構成に対応しています。init コマンドによって生成される個々のプロセスには、1 つまたは複数の実行レベルが割り当てられます。割り当てられる実行レベルは次のとおりです。

- 0 停止状態
- s または S シングルユーザ状態
- 2 ネットワーク・サービスなしのマルチユーザ状態
- 3 ネットワーク・サービス付きのマルチユーザ状態

`Runleve` フィールドに 2 つ以上の実行レベルを任意に組み合わせて指定することによって、1 つのプロセスに対して複数の実行レベルを定義することができます。

Action

このフィールドは、指定されたプロセスの処理方法を `init` に指示します。このフィールドは 20 文字です。このフィールドには次のようなキーワードを記述します。

| | |
|--------------------------|---|
| <code>respawn</code> | そのプロセスが存在しない場合または停止している場合、 <code>init</code> はそのプロセスを開始します。プロセスが存在している場合は、 <code>init</code> は何もせずに <code>inittab</code> ファイルの走査を続けます。 |
| <code>wait</code> | エントリの実行レベルと一致する実行レベルに入ると、 <code>init</code> はプロセスを開始して、終了するのを待ちます。この実行レベルに限り、 <code>init</code> は <code>inittab</code> ファイル内のエントリの読み取りを中断します。 |
| <code>initdefault</code> | <code>init</code> が最初に呼び出される際にこのキーワードのある行が処理されます。 <code>init</code> プログラムはこの行を調べて実行レベルを判定します。このプログラムは実行レベル・フィールドに指定された最上位の実行レベルを初期状態として使用します。実行レベル・フィールドが空の場合は <code>0s23</code> と解釈し、 <code>init</code> は実行レベル 3 に入ります。 <code>inittab</code> ファイルに <code>initdefault</code> キーワードがない場合、 <code>init</code> は初期の実行レベルをオペレータに要求します。 |

その他のキーワードも利用可能です。詳細については、`inittab(4)` を参照してください。

Command

これは、最大 1024 文字のデータ・フィールドです。このフィールドには、`sh` コマンドが含まれます。コマンド・フィールドのエントリには

前に `exec` を付けます。有効な `sh` 構文はすべてコマンド・フィールドに記述できます。

コメント文字 (`#`) を付けてコメントを挿入することができます。行継続文字 (`\`) を行の終わりに記述することもできます。

`/etc/inittab` のエントリを変更または追加する前に、関連するファイルとコマンド実行スクリプトの機能と内容を十分理解してください。

次の項では `/etc/inittab` の使用方法について詳しく説明します。

3.1.1.1 初期化時の省略時実行レベルの指定

`init` プログラムは、ブート時に `inittab` ファイルを調べて `initdefault` キーワードを探し、実行レベルの定義を見つけます。`inittab` に `initdefault` のエントリがない場合は、システムは実行レベルの入力を要求します。上記の `inittab` ファイルの例では、次に示す行で `initdefault` の実行レベルを 3 に設定しています。実行レベル 3 はネットワーク・サービス付きのマルチユーザ・モードを示します。

```
is:3:initdefault:
```

3.1.1.2 待ち実行レベルの指定

`init` プログラムは、`inittab` ファイルを調べて `wait` エントリを探します。上記の `inittab` ファイルの例では、次の行に `wait` エントリが記述されています。

```
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
```

この場合、`init` プログラムは `fs` エントリ用の `/sbin/bcheckrc` スクリプトを呼び出します。このエントリに関連するプロセスは、実行レベル 2 と 3 で実行されます。入力システム・コンソールから行います (`/dev/console`)。システムおよびプロセスのエラー・メッセージは、コンソールに送られます (`> /dev/console 2>&1`)。

`bcheckrc` コマンド実行スクリプトには、ファイル・システムの検査とマウントに関連する手順が含まれています。詳細については、`/sbin/bcheckrc` ファイルを参照してください。

3.1.1.3 コンソール実行レベルの指定

システム管理者またはその他のユーザがシステムにログインするためには、`getty` プログラムまたは `xm` プログラムのいずれかが実行中でなければなりません。これらのプログラムは、各端末またはワークステーションのログイン・プログラムおよびシェル・プログラムを実行するためのプロセスをセットアップします。初期の作業の大部分はシステム・コンソールで行われるため、`/etc/inittab` ファイルには、コンソール用の `getty` プロセスをセットアップするためのエントリが含まれています。`xm` プロセスは `/sbin/rc3.d` ディレクトリ内の実行レベル・スクリプトで開始されています。

3.1.1 項に示した `inittab` ファイルの例では、システム・コンソール用に次のエントリが含まれています。

```
cons:1234:respawn:/usr/sbin/getty console console vt100
```

`init` プログラムは、システム・コンソール (`/dev/console`) の端末回線属性を設定する `getty` プログラムを呼び出すように指示されています。実行レベル・フィールドには、`getty` プロセスが実行レベル 1, 2, 3, および 4 で実行されるように指定されています。`respawn` キーワードは、実行中のプロセスが終了した場合に `getty` プロセスを再生成するように、`init` に指示しています。プロセスが実行中である場合は、`init` はプロセスを再生成しません。終了した場合にプロセスを再生成します。

注意

システム・コンソールのアクセスを別の実行レベルに制限したい場合以外は、通常 `inittab` ファイル内のシステム・コンソール・エントリは変更しないでください。この端末回線の実行レベルの範囲に制限を設けると、コンソールの `getty` プロセスの実行を禁止する実行レベルにシステムが入った場合に、システム・コンソールが使用不能になる可能性があります。

3.1.1.4 端末と端末実行レベルの指定

システムがサポートしている各端末でユーザがログインできるように設定するには、システムで利用可能なターミナル・タイプを調べ、サポートされている個々のターミナル・タイプに対して実行レベルと `getty` プロセスを定義しなければなりません。次のデータベースとファイルを使用してください。

- /usr/lib/terminfo データベース

このデータベースは /usr/share/lib/terminfo ヘシンボリック・リンクされています。このデータベースはさまざまなターミナル・タイプを定義しています。

- /etc/inittab ファイル

このファイルのエントリは、サポートされているターミナル・タイプの実行レベルと `getty` プロセスを定義します。

Tru64 UNIX オペレーティング・システムは、広範囲のターミナル・タイプをサポートしています。terminfo データベースには、各ターミナル・タイプとその機能が記述されています。このデータベースは、ソース・ファイルをデータ・ファイルにコンパイルする `tic` プログラムによって作成されます。terminfo ソース・ファイルは通常、特定のフォーマットに準拠した最低 1 つのデバイス記述から構成されています。ソース・ファイルの作成とコンパイルの詳細については、`terminfo(4)` を参照してください。

/usr/lib/terminfo ディレクトリには、複数のソース・ファイルが含まれており、たとえば `name.ti` のようにそれぞれ `.ti` という接尾語が付いています。`tic` コマンドでソース・ファイルをコンパイルすると出力ファイルが /usr/lib/terminfo の下位のディレクトリに置かれます。

さまざまなコマンドとプログラムが、これらのディレクトリの中のファイルに依存しています。`TERMINFO` 環境変数を /usr/lib/terminfo ディレクトリに設定して、データベースに情報を依存しているプログラムに対して、関連する端末情報をこのディレクトリで探すように指示してください。

端末回線の定義と端末アクセスの管理については `getty(8)`、`gettydefs(4)`、および `inittab(4)` を参照してください。

3.1.1.5 プロセス実行レベルの指定

`inittab` ファイルの中の特定のエンタリで、システムが特定の実行レベルに入った場合または特定の実行レベルに変わった場合に実行されるコマンド実行スクリプトを定義します。たとえば、次の `inittab` ファイル・エンタリは、各実行レベルで `init` プログラムが実行する処理を指定しています。

```
ss:Ss:wait:/sbin/rc0 shutdown < /dev/console > /dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
```

3-10 システム環境のカスタマイズ

これらのエントリは `rc` ディレクトリ構造に関連しています。 `rc` ディレクトリ構造についての詳細は 3.1.2 項を参照してください。

3.1.1.6 端末回線の保護

`/etc/securettys` ファイルを使用して、端末回線または擬似端末回線が `root` ログインに使用できるかどうかを設定することができます。 端末回線で `root` ログインができるように設定するには、`/etc/securettys` ファイルにパス名を記述します。 擬似端末で `root` ログインができるように設定するには、`ptys` キーワードを記述します。 `X` 端末の場合は `:0` などの端末名を記述することによって `root` ログインが可能になります。 省略時の設定では、コンソールおよび `X` サーバ回線のみが `secure` に設定されています。

次の `/etc/securettys` ファイルの設定では、コンソール、`X` 端末、2 つの `LAT` 回線、およびすべての擬似端末で `root` ログインが可能です。

```
/dev/console
:0
/dev/tty00
/dev/tty01
ptys
```

3.1.2 `init` および `rc` ディレクトリの構造

オペレーティング・システムには、初期化および実行のためのコマンド・ディレクトリ構造が用意されています。 このディレクトリ構造には、次のような 4 つの主要な構成要素があります。

- `init.d` ディレクトリ
- `rc0.d` ディレクトリ
- `rc2.d` ディレクトリ
- `rc3.d` ディレクトリ

また、各 `rcn.d` ディレクトリは対応する `rcn` コマンド実行スクリプトを持っています。

3.1.2.1 `init.d` ディレクトリ

`/sbin/init.d` ディレクトリには、システムの初期化に関連する実行可能ファイルが含まれています。 たとえば、次のようなファイルが含まれています。

| | | |
|-----------------------|----------|------------|
| .mrg..autosysconfig | evm | recpasswd |
| .new..autosysconfig | gateway | rmtmpfiles |
| .new..rmtmpfiles | inet | route |
| .proto..autosysconfig | inetd | rwwho |
| .proto..rmtmpfiles | insightd | savecore |
| admincheck | kmod | security |
| advfsd | lat | sendmail |
| asudllink | lpd | settime |
| asudna | mfsmount | sia |
| asunbelink | motd | smauth |
| asutcp | ms_srv | smsd |
| audit | named | snmpd |
| autosysconfig | netrain | startlmf |
| bin | nfs | streams |
| binlog | nfsmount | syslog |
| crashdc | niffd | timed |
| cron | nis | uucp |
| dhcp | paging | write |
| dia_s_k | preserve | ws |
| enlogin | presto | xlogin |
| envmon | quota | xntpd |

3.1.2.2 rc0.d ディレクトリと rc0 コマンド実行スクリプト

rc0 スクリプトには、システムを円滑にシャットダウンし、停止またはシングルユーザ状態にするための実行コマンドが含まれています。すでに説明したように、システムをシングルユーザ・モードにシャットダウンしている場合 (レベル s) または停止している場合 (レベル 0) に、init プログラムが読み取って実行するエントリが inittab ファイルには含まれています。次に例を示します。

```
ss:Ss:wait:/sbin/rc0 shutdown < /dev/console > /dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
```

いずれの場合も、rc0 スクリプトが指定されています。スクリプト自体に記述されているコマンドの他に、rc0 には /sbin/rc0.d ディレクトリの中にあるコマンドを実行するための命令が含まれています。これらのコマンドは、init.d ディレクトリの中のファイルとリンクされています。このスクリプトには、コマンドが実行される条件も定義しています。システムが停止している場合に実行されるコマンドもあれば、システムをシャットダウンしてシングルユーザ・モードにリブートしている場合に実行されるコマンドもあります。

/sbin/rc0.d ディレクトリの中のファイルは英字 "K" または "S" で始まり、次に 2 桁の番号とファイル名が続きます。次に例を示します。

```

lrwxr-xr-x 1 root bin      17 May  8 16:35 K00enlogin -> ../init.d/enlogin
lrwxrwxrwx 1 root bin      16 May 10 10:05 K02.0ms_srv -> ../init.d/ms_srv
lrwxrwxrwx 1 root bin      16 May 10 10:03 K02.1asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root bin      20 May 10 10:03 K02.2asunbelink -> \
                                                ../init.d/asunbelink
lrwxrwxrwx 1 root bin      16 May 10 10:03 K02.3asudna -> ../init.d/asudna
lrwxrwxrwx 1 root bin      19 May 10 10:03 K02.4asudllink -> \
                                                ../init.d/asudllink
lrwxrwxrwx 1 root bin      13 May  8 16:39 K05lpd -> ../init.d/lpd
lrwxrwxrwx 1 root bin      13 May 10 11:06 K07lat -> ../init.d/lat
lrwxr-xr-x 1 root bin      15 May  8 16:35 K08audit -> ../init.d/audit
lrwxrwxrwx 1 root bin      14 May 10 11:06 K09dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root bin      15 May  8 16:37 K10inetd -> ../init.d/inetd
lrwxr-xr-x 1 root bin      15 May  8 16:37 K14snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root system  16 May 10 11:06 K16envmon -> ../init.d/envmon
lrwxr-xr-x 1 root bin      16 May  8 16:37 K19xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root bin      15 May  8 16:37 K20xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root bin      15 May  8 16:37 K21timed -> ../init.d/timed
lrwxr-xr-x 1 root bin      14 May  8 16:35 K22cron -> ../init.d/cron
lrwxr-xr-x 1 root bin      18 May  8 16:35 K25sendmail -> \
                                                ../init.d/sendmail
lrwxrwxrwx 1 root bin      13 May  8 16:37 K30nfs -> ../init.d/nfs
lrwxr-xr-x 1 root bin      16 May  8 16:35 K31presto -> ../init.d/presto
lrwxrwxrwx 1 root bin      18 May  8 16:37 K35nfsmount -> \
                                                ../init.d/nfsmount
lrwxr-xr-x 1 root bin      13 May  8 16:37 K38nis -> ../init.d/nis
lrwxrwxrwx 1 root bin      15 May 10 11:06 K40named -> ../init.d/named
lrwxr-xr-x 1 root bin      14 May  8 16:37 K42rwho -> ../init.d/rwho
lrwxr-xr-x 1 root bin      15 May  8 16:37 K43route -> ../init.d/route
lrwxr-xr-x 1 root bin      17 May  8 16:37 K44gateway -> \
                                                ../init.d/gateway
lrwxr-xr-x 1 root bin      16 May  8 16:35 K45syslog -> ../init.d/syslog
lrwxrwxrwx 1 root bin      14 May 10 11:07 K46uucp -> ../init.d/uucp
lrwxr-xr-x 1 root bin      15 May  8 16:35 K47write -> ../init.d/write
lrwxr-xr-x 1 root bin      16 May  8 16:35 K48binlog -> ../init.d/binlog
lrwxr-xr-x 1 root bin      14 May  8 16:37 K50inet -> ../init.d/inet
lrwxr-xr-x 1 root bin      17 May  8 16:37 K50netrain -> \
                                                ../init.d/netrain
lrwxr-xr-x 1 root bin      15 May  8 16:37 K51nifd -> ../init.d/nifd
lrwxr-xr-x 1 root bin      15 May  8 16:35 K52quota -> ../init.d/quota
lrwxr-xr-x 1 root bin      13 May  8 16:35 K95evm -> ../init.d/evm
lrwxr-xr-x 1 root bin      14 May  8 16:35 K96acct -> ../init.d/acct

```

一般に、システムは英字 "S" で始まるコマンドは起動し、英字 "K" で始まるコマンドは停止します。番号でソートされて昇順でコマンドが実行されるため、/sbin/rc0.d ディレクトリのコマンドの番号付けは重要です。

詳細については、rc0(8) を参照してください。

3.1.2.3 rc2.d ディレクトリと rc2 コマンド実行スクリプト

/sbin/rc2 スクリプトには、実行レベル 2 (ネットワーク・サービスなしのマルチユーザ・モード) にシステムを初期化するための実行コマンドが含まれています。inittab には、init プログラムが読み取るエントリが含まれています。init プログラムは、システムのブート時や、状態を実行レベル 2 に変更する際に、inittab ファイルのエントリを読み取って実行します。次に例を示します。

```
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
```

この例では rc2 スクリプトが指定されています。スクリプト自体に記述されているコマンドの他に、rc2 には /sbin/rc2.d ディレクトリの中にあるコマンドを実行するための命令が含まれています。これらのコマンドは通常、init.d ディレクトリの中のファイルとリンクされています。このスクリプトには、コマンドが実行される条件も定義しています。システムをブートしている場合に実行されるコマンドもあれば、システムの実行レベルを変更している場合に実行されるコマンドもあります。

/sbin/rc2.d ディレクトリの中のファイルは英字 "K" または英字 "S" で始まり、次に 2 桁の番号とファイル名が続きます。次に例を示します。

```
lrwxr-xr-x 1 root bin 17 May 8 16:35 K00enlogin -> ../init.d/enlogin
lrwxrwxrwx 1 root bin 16 May 10 10:05 K02.0ms_srv -> ../init.d/ms_srv
lrwxrwxrwx 1 root bin 16 May 10 10:03 K02.1asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root bin 20 May 10 10:03 K02.2asunbelink -> \
    ../init.d/asunbelink
lrwxrwxrwx 1 root bin 16 May 10 10:03 K02.3asudna -> ../init.d/asudna
lrwxrwxrwx 1 root bin 19 May 10 10:03 K02.4asudllink -> \
    ../init.d/asudllink
lrwxrwxrwx 1 root bin 13 May 8 16:39 K05lpd -> ../init.d/lpd
lrwxrwxrwx 1 root bin 13 May 10 11:06 K07lat -> ../init.d/lat
lrwxr-xr-x 1 root bin 15 May 8 16:35 K08audit -> ../init.d/audit
lrwxrwxrwx 1 root bin 14 May 10 11:06 K09dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root bin 15 May 8 16:37 K10inetd -> ../init.d/inetd
lrwxr-xr-x 1 root bin 15 May 8 16:37 K14snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root system 16 May 10 11:06 K16envmon -> \
    ../init.d/envmon
lrwxr-xr-x 1 root bin 16 May 8 16:37 K19xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root bin 15 May 8 16:37 K20xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root bin 15 May 8 16:37 K21timed -> ../init.d/timed
lrwxr-xr-x 1 root bin 14 May 8 16:35 K22cron -> ../init.d/cron
lrwxr-xr-x 1 root bin 18 May 8 16:35 K25sendmail -> \
    ../init.d/sendmail
lrwxrwxrwx 1 root bin 13 May 8 16:37 K30nfs -> ../init.d/nfs
lrwxr-xr-x 1 root bin 16 May 8 16:35 K31presto -> ../init.d/presto
lrwxrwxrwx 1 root bin 18 May 8 16:37 K35nfsmount -> \
    ../init.d/nfsmount
lrwxr-xr-x 1 root bin 13 May 8 16:37 K38nis -> ../init.d/nis
lrwxrwxrwx 1 root bin 15 May 10 11:06 K40named -> ../init.d/named
lrwxr-xr-x 1 root bin 14 May 8 16:37 K42rwho -> ../init.d/rwho
lrwxr-xr-x 1 root bin 15 May 8 16:37 K43route -> ../init.d/route
lrwxr-xr-x 1 root bin 17 May 8 16:37 K44gateway -> \
    ../init.d/gateway
lrwxr-xr-x 1 root bin 16 May 8 16:35 K45syslog -> ../init.d/syslog
```

通常、システムは英字 "S" で始まるコマンドは起動し、"K" で始まるコマンドは停止します。"K" で始まるコマンドは、システムが実行レベルを高いレベルから低いレベルへ変更している場合にのみ実行されます。"S" で始まるコマンドはどんな場合にも実行されます。番号でソートされて昇順でコマンドが実行されるため、/sbin/rc2.d ディレクトリのコマンドの番号付けは重要です。

詳細については、rc2(8) を参照してください。

3.1.2.4 rc3.d ディレクトリと rc3 コマンド実行スクリプト

/sbin/rc3 スクリプトには、ネットワークサービス付きのマルチユーザ状態 (実行レベル 3) にシステムを初期化するための実行コマンドが含まれています。すでに説明したように、システムをブートしている場合、または状態を実行レベル 3 に変更している場合に init プログラムが読み取って実行するエントリが inittab ファイルには含まれています。次に例を示します。

```
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
```

この例では rc3 スクリプトが指定されています。スクリプト自体に記述されているコマンドの他に、rc3 には /sbin/rc3.d ディレクトリの中にあるコマンドを実行するための命令が含まれています。これらのコマンドは通常、init.d ディレクトリの中のファイルとリンクされています。このスクリプトには、コマンドが実行される条件も定義しています。システムをブートしている場合に実行されるコマンドもあれば、システムの実行レベルを変更している場合に実行されるコマンドもあります。

/sbin/rc3.d ディレクトリの中のファイルは英字 "S" で始まり、その後には 2 桁の番号とファイル名が続きます。次に例を示します。

```
lrwxr-xr-x 1 root bin 15 May 8 16:37 S00cniffd -> ../init.d/niffd
lrwxr-xr-x 1 root bin 17 May 8 16:37 S00fnetrain -> ../init.d/netrain
lrwxr-xr-x 1 root bin 14 May 8 16:37 S00inet -> ../init.d/inet
lrwxr-xr-x 1 root bin 15 May 8 16:35 S01quota -> ../init.d/quota
lrwxrwxrwx 1 root bin 14 May 10 11:07 S04uucp -> ../init.d/uucp
lrwxr-xr-x 1 root bin 18 May 8 16:35 S08startlmf -> ../init.d/startlmf
lrwxr-xr-x 1 root bin 16 May 8 16:35 S09syslog -> ../init.d/syslog
lrwxr-xr-x 1 root bin 16 May 8 16:35 S10binlog -> ../init.d/binlog
lrwxr-xr-x 1 root bin 17 May 8 16:37 S11gateway -> ../init.d/gateway
lrwxr-xr-x 1 root bin 15 May 8 16:37 S12route -> ../init.d/route
lrwxr-xr-x 1 root bin 14 May 8 16:37 S13rwho -> ../init.d/rwho
lrwxr-xr-x 1 root bin 17 May 8 16:35 S14settime -> ../init.d/settime
lrwxrwxrwx 1 root bin 15 May 10 11:06 S15named -> ../init.d/named
lrwxr-xr-x 1 root bin 13 May 8 16:37 S18nis -> ../init.d/nis
lrwxrwxrwx 1 root bin 13 May 8 16:37 S19nfs -> ../init.d/nfs
lrwxrwxrwx 1 root bin 18 May 8 16:37 S20nfsmount -> ../init.d/nfsmount
lrwxr-xr-x 1 root bin 15 May 8 16:35 S21audit -> ../init.d/audit
lrwxr-xr-x 1 root bin 18 May 8 16:35 S25preserve -> ../init.d/preserve
lrwxr-xr-x 1 root bin 20 May 8 16:35 S30rmtmpfiles -> ../init.d/rmtmpfiles
lrwxr-xr-x 1 root bin 16 May 8 16:35 S36presto -> ../init.d/presto
lrwxr-xr-x 1 root bin 18 May 8 16:35 S40sendmail -> ../init.d/sendmail
lrwxr-xr-x 1 root bin 15 May 8 16:37 S45xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root bin 15 May 8 16:37 S46timed -> ../init.d/timed
lrwxr-xr-x 1 root bin 15 May 8 16:37 S49snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root bin 18 May 8 16:44 S50insightd -> ../init.d/insightd
lrwxrwxrwx 1 root system 16 May 10 11:06 S51envmon -> ../init.d/envmon
lrwxrwxrwx 1 root bin 16 May 8 16:41 S53advfsd -> ../init.d/advfsd
lrwxr-xr-x 1 root bin 15 May 8 16:37 S55inetd -> ../init.d/inetd
lrwxrwxrwx 1 root bin 14 May 10 11:06 S56dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root bin 14 May 8 16:35 S57cron -> ../init.d/cron
```

```

lrwxrwxrwx 1 root bin 13 May 10 11:06 S58lat -> ../init.d/lat
lrwxr-xr-x 1 root bin 14 May 8 16:35 S60motd -> ../init.d/motd
lrwxrwxrwx 1 root bin 19 May 10 10:03 S61.0asudllink -> \
    ../init.d/asudllink
lrwxrwxrwx 1 root bin 16 May 10 10:03 S61.1asudna -> ../init.d/asudna
lrwxrwxrwx 1 root bin 20 May 10 10:03 S61.2asunbelink -> \
    ../init.d/asunbelink
lrwxrwxrwx 1 root bin 16 May 10 10:03 S61.3asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root bin 16 May 10 10:05 S61.4ms_srv -> ../init.d/ms_srv
lrwxr-xr-x 1 root bin 15 May 8 16:35 S63write -> ../init.d/write
lrwxrwxrwx 1 root bin 13 May 8 16:39 S65lpd -> ../init.d/lpd
lrwxr-xr-x 1 root bin 17 May 8 16:35 S80crashdc -> ../init.d/crashdc
lrwxr-xr-x 1 root bin 12 May 8 16:45 S90ws -> ../init.d/ws
lrwxr-xr-x 1 root bin 16 May 8 16:37 S95xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root bin 13 May 8 16:35 S97evm -> ../init.d/evm
lrwxr-xr-x 1 root bin 16 May 8 16:35 S98smauth -> ../init.d/smauth
lrwxr-xr-x 1 root bin 20 May 8 16:35 S99admincheck -> \
    ../init.d/admincheck
lrwxr-xr-x 1 root bin 14 May 8 16:38 S99smsd -> ../init.d/smsd

```

通常、システムは英字 "S" で始まるコマンドを起動し、"K" で始まるコマンドを停止します。英字 "K" で始まるコマンドは、システムが実行レベルの高いレベルから低いレベルへ変更している場合にのみ実行されます。"S" で始まるコマンドはどんな場合にも実行されています。

通常、英字 "S" で始まるコマンドのみを `rc3.d` ディレクトリに置きます。省略時の設定では実行レベル 3 が最も高い実行レベルであるため、それより高い実行レベルから実行レベル 3 へ変更することはできません。したがって、"K" で始まるコマンドは実行できません。番号でソートされて昇順でコマンドが実行されるため、`/sbin/rc3.d` ディレクトリのコマンドの番号付けは重要です。

詳細については、`rc3(8)` を参照してください。

3.1.3 crontabs ディレクトリ

`crontab` コマンドは、コマンドのスケジュールを `cron` システム・クロック・デーモンに発行します。`cron` デーモンは、`/var/spool/cron/crontabs` ディレクトリの中のファイルに指定された日付と時刻に従ってシェル・コマンドを実行します。定期的に行うコマンドは、このファイルの中にあります。1 度だけ実行するコマンドは、`/var/spool/cron/atjobs/*` ファイルの中にあり、`at` コマンドで発行されます。

`/var/spool/cron/crontabs` ディレクトリの中の次に示すファイル・エントリの例は、月曜日から土曜日までの午前 2 時に `runacct` コマンドが実行され、出力が `/var/adm/acct/nite/fd2log` ファイルに送られることを指定しています。

```
0 2 * * 1-6 /usr/sbin/acct/runacct > /var/adm/acct/nite/fd2log&  
  ①      ②      ③
```

各エントリの構文は次のとおりです。

- ① 分，時，日，月，および曜日を指定します。曜日については，値 0 (ゼロ) は日曜日を示し，値 1 は月曜日を示します。それぞれの値は，1 つの値，コンマで区切って 2 つ以上の値，またはダッシュ (-) で値の範囲を指定することができます。また，アスタリスク (*) で，すべての値を示すこともできます。たとえば，時にアスタリスク (*) を指定すると，コマンドは 1 時間ごとに実行されます。
- ② 指定された時刻に実行するコマンドを指定します。
- ③ コマンドの引数を任意に指定します。

コメントをファイルに追加するには，行頭に番号記号 (#) を指定してください。

/var/spool/cron/crontabs ディレクトリの中のファイルにはシステム・ユーザにちなんだ名称が付けられ，ファイルの中のコマンドはそのユーザの権限のもとで実行されます。たとえば，adm ファイルのコマンドは adm の権限のもとで実行されます。

crontab コマンドを使用するためには，実行するファイル名と一致するユーザでなければなりません。たとえば，ユーザ adm が crontab コマンドを実行した場合，そのコマンドは /var/spool/cron/crontabs/adm ファイルに対して実行されます。

たとえば，adm 権限のもとで実行されるものとしてコマンドを cron デーモンに発行するには，次のようにします。

1. adm でログインします。
2. 次のように -l オプション付きで crontab コマンドを入力して，
/usr/spool/cron/crontabs/adm ファイルをホーム・ディレクトリ
の中の一時ファイルにコピーします。

% crontab -l > temp_adm
3. 一時ファイルを編集して，指定した時刻に実行したいコマンドを追加します。

4. `crontab` コマンドを入力して、`cron` デーモンにコマンドを発行するための一時ファイルを指定します。

たとえば、次のように入力します。

```
% crontab temp_adm
```

`/var/adm/cron/log` ファイルには、`cron` デーモンによって実行されたコマンドの履歴が入っています。

`/usr/spool/cron/crontabs/root` ファイルを使用して、システム・ログ・ファイルのバックアップおよびクリーンアップができます。

`root crontab` ファイル `/usr/var/spool/cron/crontabs/root` には、`/var/adm/wtmp` ログ・ファイルを毎週日曜日の午前 2 時にクリーンアップするモデル・エントリがあります。ログ・ファイルを圧縮したバックアップが 1 つ、次のクリーンアップまで保持されます。`crontab` エントリは、次のように、省略時の設定で有効になっています。

```
# To get the standard output by e-mail remove the output redirection.
#
0 2 * * 0 /usr/sbin/logclean /var/adm/wtmp > /dev/null
```

ローカル・システムの要件に合うように、タスクを追加したり、既存のタスクを変更してください。

上記の例では、出力が省略時の設定で `/dev/null` にリダイレクトされます。これを e-mail アドレスにリダイレクトして、タスクが終了したときに通知を受け取るようにすることもできます。この `cron` タスクは、ログイン・ログ・ファイルをバックアップし、新しい空のファイルを作成します (ログイン・ログには、システム上のすべてのユーザのログインが記録されます)。

ログ・ファイルをもっと長い期間保存した場合は、クリーンアップの頻度を変えるか、`./crontabs/root` の該当エントリをコメントアウトします。また、プリント・サービス関連のファイルなど、別のシステム・ログ・ファイルをクリーンアップする `cron` タスクを作成したい場合があるかもしれません。

`root crontab` ファイルを編集する場合は、`root` (スーパーユーザ) になり、必ず次のコマンドを使用してください。

```
# crontab -e
```

`/usr/bin/ed` 以外のエディタを使いたい場合は、あらかじめ環境変数 `EDITOR` を設定し、エクスポートしておく必要があります。

詳細については、`crontab(1)` を参照してください。

3.2 各国語サポート

オペレーティング・システムは、プログラムの言語固有および国固有の情報を提供し、またサポートを行います。

システム管理者に直接関係のあるソフトウェアの構成要素は、`/usr/lib/nls`にあるディレクトリとファイルです。

国際化されたシステムは、さまざまな方法で情報を提示します。ロケールとは、世界の特定の地域に対応する言語、テリトリ、およびコードセットの要件を指します。システムは、ロケール固有のデータを2種類のファイルに分けて保存します。

ロケール・ファイル これらのファイルには、月と日の名前、日付形式、通貨および数値の形式、有効な `yes/no` 文字列、文字分類データ、および照合順序に関する情報が含まれています。これらのファイルは、`/usr/lib/nls/loc` ディレクトリにあります。

メッセージ・カタログ これらのファイルには、プログラムが使用するメッセージ文字列が含まれています。これらのファイルは、`/usr/lib/nls/msg/locale-name` ディレクトリにあります。

表 3-1 は、シングル・バイトのヨーロッパ言語ロケールのサブセットをインストールしたときに `/usr/lib/nls/loc` に移動するロケールの例です。各国際化サブセットをインストールする場合にはそれぞれのライセンスを取得する必要があります。

表 3-1: ロケール・サポート・ファイル

| 言語/テリトリ | ロケール・ファイル名 |
|-------------------|-----------------|
| Danish-Denmark | da_DK.ISO8859-1 |
| Dutch-Netherlands | nl_NL.ISO8859-1 |
| Dutch_Belgium | nl_BE.ISO8859-1 |
| English_U.K | en_GB.ISO8859-1 |
| English_U.S.A. | en_US.ISO8859-1 |

表 3-1: ロケール・サポート・ファイル (続き)

| 言語/テリトリ | ロケール・ファイル名 |
|-----------------|-----------------|
| Finnish-Finland | fi_FI.ISO8859-1 |
| French_Belgium | fr_BE.ISO8859-1 |
| French_Canada | fr_CA.ISO8859-1 |
| French_France | fr_FR.ISO8859-1 |

注意

/usr/lib/nls/loc ディレクトリには表 3-1 にリストされているファイルと対応する環境テーブル (.en ファイル) と文字テーブル (.8859* ファイル) が含まれることがありますが、これらのテーブルは古いプログラムに対するシステムの互換性を保つためのものなので、新しいアプリケーションで使用しないでください。

国際化オプションと国際的なソフトウェアの開発をサポートする機能については、次のリファレンス・ページを参照してください。

| | |
|------------------|--|
| code_page(5) | Microsoft Windows と Windows NT システムで使用する文字コードセット |
| iconv_intro(5) | コードセット変換の概要 |
| iconv(1) | コード化された文字を別の文字セットに変換するコマンドの説明 |
| i18n_intro(5) | 国際化 (I18N) の概要 |
| i18n_printing(5) | プリンタ・サポートの国際化 (I18N) の概要 |
| l10n_intro(5) | ローカライズ (L10N) の概要 |
| locale(1) | ロケールについての情報 |

国際化に関するリファレンス・ページは、ここにリストしたものだけではありません。各リファレンス・ページの関連情報セクションまたは『国際化ソフトウェア・プログラミング・ガイド』を参照してください。日本語ロケールについては『日本語機能ガイドブック』を参照してください。

3.2.1 ロケールの設定

国際化のための省略時のシステム全体でのロケールは、C ロケールです。省略時のシステム全体でのロケールは、ユーザが LC_TIME, LC_COLLATE などの各国別環境変数の設定省略時に `setlocale` 機能が使用するロケールです。

システム全体で Bourne シェルおよび Korn シェルのユーザの省略時のロケールを変更するには、`/etc/profile` ファイルを編集し、設定するロケールの名前を書き込みます。 `setlocale` 機能は、このファイルで指定したロケールを使用します。C シェルのユーザの省略時のロケールを変更する場合は `/etc/csh.login` ファイルを編集し、設定するロケールの名前を書き込みます。

ネイティブ・ロケールは、`/usr/lib/nls/loc` ディレクトリ内のどのロケールにも設定することができます。

ロケールを設定するには、初期化ファイル内の 1 つまたは複数の環境変数にロケール名を割り当ててください。最も簡単な方法は、LANG 環境変数に値を割り当てることです。この環境変数はロケールのすべての構成要素に影響を与えます。

注意

C ロケールがシステムの省略時の設定です。C ロケールは、アメリカ英語を指定し、7 ビットの ASCII コード・セットを使用します。C ロケールとアメリカ英語ロケール (`en_US.ISO8859-1`) との主な違いは後者がエラー・メッセージを強化していることです。

次の例では、C シェルとそのシェルのすべての子プロセスのロケールをフランス語に設定します。

```
% setenv LANG fr_FR.ISO8859-1
```

別のシェルに異なるロケールを割り当てたい場合は、そのシェルの LANG 環境変数を再設定することができます。次の例では、Korn シェルおよび Bourn シェルのロケールがフランス語に設定されます。

```
$ LANG=fr_FR.ISO8859-1
$ export LANG
```

コマンド行で LANG 環境変数を設定すると、現在のプロセスに対してのみロケールが設定されます。

システム管理者および他のユーザがログインする際に必ずロケールを設定するよう設定するためには、省略時のシェルのログイン・スクリプトを編集してください。C シェルの場合は、`.login` ファイルの LANG 環境変数を設定してください。Bourne シェルまたは Korn シェルの場合は、`.profile` ファイルの LANG 環境変数を設定してください。

ほとんどの場合、LANG 環境変数に値を割り当てることによってロケールを設定することができます。これは、LANG 環境変数でロケールを設定すると、次の機能に対する省略時の値が自動的に設定されるためです。

- 照合
- 文字分類
- 日付と時刻の表記形式
- 数値および通貨の表記形式
- プログラム・メッセージ
- Yes/No プロンプト

ただし、ごくまれに、ロケール内の上記のいずれかのカテゴリの省略時の設定を変更する必要があります。この場合は、これらのカテゴリを指定する変数を再設定します。詳細については、次の項を参照してください。

3.2.2 ロケール・カテゴリの変更

LANG 環境変数でロケールを設定すると、ロケールに適した照合順序、文字分類機能、日付と時刻の表記形式、数値および通貨の表記形式、プログラム・メッセージ、および Yes/No プロンプトについて自動的に省略時の値が設定されます。ただし、省略時の値のいずれかを変更する必要がある場合には、これらのカテゴリに関連する環境変数を再設定します。

表 3-2 に、ロケール・カテゴリに影響する環境変数を示します。

表 3-2: ロケール環境変数

| 環境変数 | 説明 |
|-------------|---|
| LC_ALL | LANG を含む他のすべての国際化環境変数の設定を変更する。 |
| LC_COLLATE | 名前をソートする際、およびパターンに文字範囲が存在する場合に使用する照合順序を指定する。 |
| LC_CTYPE | 使用する文字分類情報を指定する。 |
| LC_NUMERIC | 数値のフォーマットを指定する。 |
| LC_MONETARY | 通貨フォーマットを指定する。 |
| LC_TIME | 日付と時刻の形式を指定する。 |
| LC_MESSAGES | システム・メッセージの表示に使用する言語を指定する。また yes/no プロンプトで "yes" と "no" を示すための文字列を指定する。 |

LANG 環境変数の場合と同様に、これらすべてのカテゴリ変数にはロケール名を割り当てることができます。たとえば、ある会社で一般的に使用する言語がスペイン語であるとします。LANG 環境変数でロケールをスペイン語に設定し、数値と通貨のフォーマットをアメリカ英語に設定することができます。この場合、次のような変数の割り当てを実行します。

```
% setenv LANG es_ES.ISO8859-1
% setenv LC_NUMERIC en_US.ISO8859-1
% setenv LC_MONETARY en_US.ISO8859-1
```

ロケール名に *@modifier* を含めて、別カテゴリの特別な要求に対応する特定のバージョンを示すことができます。

たとえば、辞書の順番と電話帳の順番という2つの方法でデータをソートすることができます。フランスにあるサイトで、省略時のフランス語ロケールを使用し、このロケールでは標準的なセットアップとして辞書の順番を使用するとします。しかし、この例では、電話帳の順序でデータを照合する、サイト定義のロケールもサイトで使用する必要があるとします。この場合は、次のように環境変数を設定することができます。

```
% setenv LANG fr_FR.ISO8859-1
% setenv LC_COLLATE fr_FR.ISO8859-1@phone
```

LC_COLLATE を明示的に設定することによって、LANG 変数の暗黙の設定が指定変更されます。

3.2.3 ロケール変数の制限事項

LANG および LC_* 環境変数によって、希望どおりにロケールを設定することができますが、誤りを防ぐことはできません。たとえば、LANG をスウェーデンのロケールに設定し、LC_CTYPE をポルトガルのロケールに設定してしまう可能性もあります。

同様に、ロケール情報とデータを結び付けることもできません。つまり、システムには、ユーザがあるファイルを作成した場合に設定したロケールを知る方法がなく、ユーザが後でそのデータを不適当な方法で処理することをシステムが防ぐことはできないということです。たとえば、foo ファイルを作成した場合に LANG がドイツ語のロケールに設定されていたとします。ここで、LANG をスペイン語のロケールに再設定し、foo に対して grep を実行したとします。この場合 grep コマンドは、ファイル内のドイツ語のデータに対してスペイン語の規則を使用します。

3.2.4 メッセージ・カタログとロケールについての環境変数の設定

メッセージ・カタログの位置を定義するには、NLSPATH 環境変数を設定してください。省略時のパスは次のとおりです。

```
NLSPATH=/usr/lib/nls/msg/%L/%N:
```

%L は現在のロケール名を指定し、%N はメッセージ・カタログの名前の値を示します。

また、LOCPATH 環境変数はロケールの探索パスを定義します。省略時のパスは次のとおりです。

```
LOCPATH=/usr/lib/nls/loc:
```

3.3 国際化機能のカスタマイズ

オペレーティング・システムには、多くの国際化機能があります。ユーザまたはローカル・サイトの責任者が、オペレーティング・システムのどの国際化機能(ワールドワイド・サポート機能とも呼ばれる)の構成要素が必要かを決定します。ワールドワイド・サポートはオプションのサブセットであり、インストール時に選択することができます。システム管理者の仕事は、次のようなユーザのために、これらの機能を設定して保守することです。

- 多国語対応アプリケーションを作成するソフトウェア開発者
- システムで多国語対応アプリケーションを実行するユーザ

ワールドワイド・サポートに関して 3 つの情報源が用意されています。

- 国際化をサポートするオプションのソフトウェア・サブセットの一覧については、『インストール・ガイド』を参照してください。
- 多国語対応ソフトウェアを作成するプログラマのために、オペレーティング・システム環境を設定して、保守する方法については、『国際化ソフトウェア・プログラミング・ガイド』を参照してください。
- 多国語対応アプリケーションのユーザのために、システムを設定して保守する方法については、システム設定グラフィカル・ユーザ・インタフェースを使用し、「システム設定」アイコンをクリックしてから「I18N」アイコンをクリックしてください。「I18N」ウィンドウからタスクを選択して、システム上のワールドワイド・サポート機能を構成したり変更することができます。ただしこのオプションを有効にする場合、少なくとも 1 つのワールドワイド言語サポート・ソフトウェア・サブセットをインストールしている必要があります。また、このオプションを CDE アプリケーション・マネージャから起動することもできます。CDE の使用については、第 1 章を参照してください。

3.4 時間帯のカスタマイズ

時間帯情報は、`/etc/zoneinfo` ディレクトリ内のファイルに格納されています。`/etc/zoneinfo/localtime` ファイルは、`/etc/zoneinfo` ディレクトリ中のファイルとリンクされており、ローカル時間帯を指定します。これらのファイルは、システムのインストール時にリンクされますが、スーパーユーザは、`/etc/zoneinfo/localtime` ファイルを再リンクしてローカル時間帯を変更することができます。たとえば、次のコマンドは、ローカル時間帯をアメリカ大陸のニューヨークと一致するように変更します。

```
# ln -sf /etc/zoneinfo/America/New_York /etc/zoneinfo/localtime
```

`/etc/zoneinfo/sources` ディレクトリには、全世界の時間帯情報および夏時間情報を指定するソース・ファイルが含まれています。これらの情報を使用して `/etc/zoneinfo` ディレクトリにファイルが生成されます。ソース・ファイルの情報を変更し、`zic` コマンドを使用して `/etc/zoneinfo` ディレクトリに新しいファイルを生成することができます。時間帯データベース・ファイルの形式の詳細は、`zic(8)` を参照してください。

`.login` ファイルまたはシェル環境ファイルで `TZ` 環境変数を設定すると、省略時の時間帯情報を変更することができます。`TZ` 環境変数を定義すると、

/etc/zoneinfo/localtime に指定された時間帯情報の省略時の設定がその値によって無効にされます。省略時の設定では、TZ 変数は未定義です。

TZ 環境変数の構文は、次のとおりです。

```
stdoffset [ dst[offset] [, start[/ time], end[/ time]]]
```

次の構文でも指定できます。

```
stdoffset [ dst[ offset]]
```

TZ 環境変数の構文で使用するパラメータは次のとおりです。

| | |
|---------------------------|---|
| <i>std</i> および <i>dst</i> | 標準時間帯 (<i>std</i>) または夏時間帯 (<i>dst</i>) を示す 3 文字以上の文字列を指定します。 |
|---------------------------|---|

注意

夏時間はロケールにより daylight savings time または summer time と呼ばれます。

dst 変数を指定しない場合、夏時間は適用されません。任意の大文字および小文字が使用できます。先頭に、コロン (:), コンマ (,), ハイフン (-), 正符号 (+), および ASCII NUL を付けることはできません。

offset

GMT とローカル時間の差を指定します。*offset* 変数は 24 時間表示法を使用して、次の構文で指定します。

```
hh [ :mm [ :ss ]]
```

dst 変数の後に *offset* 変数を指定しない場合、夏時間には、標準時間よりも 1 時間早い時刻が使用されます。*offset* 変数の前に負符号 (-) を指定すると、その時間帯は子午線の東側にあることを意味します。省略時の設定は子午線の西側であり、正符号 (+) を付けて指定することもできます。

start および *end*

夏時間の開始および終了時期を指定します。*start* および *end* 変数は次の構文で指定します。

Jj

n

Mm.w.d

最初の構文では、*j* 変数はユリウス暦の 1 ~ 365 の日です。うるう年のうるう日 (2 月 29 日) はカウントしません。

2 番目の構文では、*n* 変数はゼロ (0) を基数とするユリウス暦の 0 ~ 365 の日です。うるう日はカウントされます。

3 番目の構文では、*m* 変数は月 (1 ~ 12)、*w* 変数は週の番号 (1 ~ 5)、*d* 変数は曜日 (0 ~ 6) を指定します。*d* 変数では、ゼロ (0) が日曜日を、6 が土曜日を意味します。

time

夏時間への移行または夏時間からの復帰時間をローカル時間で指定します。*time* 変数は 24 時間表示法を使用して次の構文で指定します。

hh [*:**mm* [*:**ss*]]

省略時の設定は 02:00:00 です。

下記の TZ 環境変数の指定例では、次のことが指定されています。

- EST (東部標準時) は GMT から 5 時間遅れの標準時を指定します。
- EDT (東部夏時間) は GMT から 4 時間遅れの夏時間を指定します。
- EDT は 4 月の第 1 日曜日に始まり、10 月の最終日曜日に終了します。夏時間への移行および夏時間からの復帰は、午前 2 時 (省略時の設定) に行われます。

EST5EDT4,M4.1.0,M10.5.0

次の構文を指定できます。

:pathname

pathname 変数は、*tzfile* ファイル・フォーマットで、時間の変換情報を含むファイルのパス名を指定します。たとえば次のように指定します。

:America/New_York

ファイル・フォーマットについての詳細は、`tzfile(4)` を参照してください。

パス名がスラッシュ (/) で始まっている場合は、絶対パス名を指定しています。それ以外の場合は、`/etc/zoneinfo` ディレクトリからの相対パスです。指定したファイルが使用できない場合または壊れている場合、システムはグリニッジ標準時 (GMT) を省略時の値として使用します。

タイム・ゾーン・フォーマットは、`SVID2` と `SVID3` で異なります。`SVID2` の場合、`/usr/sbin/timezone` が `/etc/svid2_tz` ファイルを作成します。`TZ` および `TZC` 変数の内容は、ユーザが `/usr/sbin/timezone` を実行して提供した情報をベースにしています。

`SVID3` の場合、インストール時に `/etc/svid3_tz` ファイルが作成されます。`TZ` 変数と `TZC` 変数の内容は、`/usr/sbin/timezone` 実行時にユーザが指定した時間帯に関する情報をベースにしています。

詳細については、`timezone(3)` を参照してください。

3.5 電力管理のカスタマイズ

オペレーティング・システムには電力を節約する機能が備わっており、適切なハードウェアを装備したシステムであれば、この機能を使って節電することができます。お手持ちのシステムが電力管理をサポートしているかどうかは、システムのオーナーズ・マニュアルを参照してください。電力管理ユーティリティを使用すれば、次のことができます。

- サポート対象モニタ (Energy Star) 上で省エネ機能を有効にし、電力モードと休止時間を制御する。
- 選択した休止時間が経過したときにどのディスクをスピンドownするか、を選択する。システムによっては、ある種の省エネ機能が省略時で使用可能になって出荷されます。ディスクが予期せずにスピンドownしたり、データの転送に時間がかかったりする場合は、この機能が有効になっているかどうかを確認してください。
- CPU の電力使用方法を設定する。この機能は、CPU がスロー・ダウン節電モードをサポートしているシステムでだけ使用できます。
- システム管理ユーティリティまたはコマンド行インタフェースを通して、1つのワークステーションごと、またはシステムのグループごとに、これらの機能は表示させ、設定する。オペレーティング・システム

は、ネットワーク全体にわたってハードウェアを管理しモニタするユーティリティを用意しています。

- イベント管理 (EVM) インタフェースを使って、電力管理イベントをモニタする。

節電機能は、次のユーティリティを使い、いくつかの方法で起動し管理することができます。

- 個々のワークステーションを、X11 に準拠したグラフィカル・ユーザ・インタフェース `/usr/bin/X11/dxpower` を使って管理します。これらのインタフェースを起動する方法の詳細は、オンライン・ヘルプと `dxpower(8)` を参照してください。
- `sysconfig` と `sysconfigdb` を使って、カーネル属性をロードし設定します。コマンド・オプションのリストは、`sysconfig(8)` および `sysconfigdb(8)` を参照してください。これらの電力管理メソッドは、将来のリリースで新しいメソッドに置き換わる可能性があります。

3.5.1 dxpower ユーティリティのグラフィカル・ユーザ・インタフェースの使用

グラフィカル・ユーザ・インタフェース `dxpower` は、ホスト・システムのグラフィック・コンソールで使用するか、またはコマンド行から呼び出すことができます。機能によっては、パスワードで保護されているものがあります。そのような機能は、`root` でログインするシステム管理者だけが使えます。特権のないユーザは、モニタの省エネ機能など、限られた機能だけを使用することができます。CDE を使っている場合は、次の手順に従って `dxpower` 電力管理ユーティリティをオープンすることができます。

1. 「アプリケーション・マネージャ」アイコンをクリックします。
2. 「システム管理」アプリケーション・グループ・アイコンをダブルクリックします。
3. 「日常管理」アプリケーション・グループ・アイコンをダブルクリックします。
4. 「パワー・マネジメント」アイコンをダブルクリックします。

端末または他の X11 ウィンドウ環境を使用している場合は、コマンド行から `dxdp` ユーティリティを起動することができます。次のように入力してください。

```
# /usr/bin/X11/dxdp
```

`dxdp` ユーティリティを起動すると、電力管理ウィンドウが画面に表示されます。このウィンドウには、動作モードを選択するためのチェックボックスと休止時間の限度を設定するためのスライディング・スケール(バー)があります。休止時間は、装置が節電モードになるまでの時間であり、1 ~ 60 分の間で設定します。ログイン特権で許可されれば、グラフィカル・インタフェースで次のことができます。

- ホスト・システムにあるすべてのサポート対象装置について、電力管理を有効または無効する。
- 電力管理を有効にする時間帯を指定する。たとえば、システムを夜の間だけ節電モードにするよう設定することができます。
- グラフィック・モニタの省エネ機能を有効にし、`standby`、`suspend`、および電源オフ・モードへ移るまでの最小休止時間を指定する。たとえば、システムがあまり使用されない場合、2、3 分の休止時間の後にシステムを電源オフ・モードにするよう設定することができます。
- 個々のディスクごとに節電モードの有効無効を切り替える。たとえば、ブート・ディスクは常にフル電源モードにし、使用していないユーザ・ファイル・システムは、節電のため、指定した休止時間後にスピンドアウンさせます。

注意

DPMS (Display Power Management Signaling) をサポートしていないモニタ(ディスプレイ)は、DPMS 機能を起動すると、故障する恐れがあります。オーナーズ・マニュアルに記載されているモニタの仕様を確認してください。DPMS をサポートしているモニタの場合、節電モードに置かれてから通常モードに戻る時間はさまざまです。電源オフ状態に置かれる時間が長いほど、マウスやキーボードを動かすことによって画面が再表示されるまでの時間は長くなります。これは、電力管理ソフトウェ

アの機能ではなく、モニタの蛍光体が冷えていて、暖め直すのに時間がかかるためです。

dxpowerユーティリティの使用法についての詳細を知りたい場合は、アプリケーションを開始し、ウィンドウの右下にある [ヘルプ] ボタンを選択してください。

3.5.2 sysconfig コマンドの使用

sysconfigdb データベースを管理する sysconfig コマンドを使用すると、コマンド行から電力管理属性を制御することができます。たとえば、リモート端末やローカル・コンソール端末からシステムの電力管理機能を起動する場合は、sysconfig コマンドを使用する必要があります。

CDE を実行していないコンソール端末から電力管理ツールを起動する場合には、graphics_powerdown および graphics_off_dwell 属性のみが適用されます。graphics_standby_dwell および graphics_suspend_dwell 属性を変更しても、何の影響もありません。これらの属性については、3.5.2.1 項を参照してください。

注意

sysconfig コマンドと dxpower コマンドは同時に使用しないでください。同時に使用すると、予測できない動作をすることがあります。

3.5.2.1 電力管理値の変更

カーネルの再起動時に有効になる電力管理値を変更するには、stanza を作成します。詳細については、stanza(4) を参照してください。stanza ファイルには、次の電力管理属性を含めることができます。

- default_pwrmgr_state
グローバルな電力管理状態。この属性を有効にするには 1 を指定し、無効にするには 0 を指定します。
- cpu_slowdown
CPU スローダウンの現在の状態。この属性を有効にするには 1 を指定し、無効にするには 0 を指定します。

- `disk_dwell_time`
登録されているディスクの省略時の休止時間を分単位で指定します。
- `disk_spindown`
ディスク・スピンドウンの現在の状態。この属性を有効にするには 1 を指定し、無効にするには 0 を指定します。
- `graphics_powerdown`
グラフィックスの電源切断の現在の状態。この属性を有効にするには 1 を指定し、無効にするには 0 を指定します。
- `graphics_standby_dwell`
standby DPMS (Display Power Management Signaling) モードの省略時の休止時間を分単位で指定します。この属性を無効にする場合は値 0 を指定します。

モニタ電力管理とスクリーン・セーバを同時に有効にしている場合は、DPMS 対応モニタは、アクティブ電源オフ・モードになります。また、Energy-Star 準拠のプラットフォームでは、ディスクのスピン・ダウン機能もアクティブになることがあります。これらの省エネ機能がアクティブになっているときでも、スクリーン・セーバの実行を継続できるように、X サーバは省エネ機能を無効にすることがあります。電力消費を最小にするには、以下のいずれかの手段で、アクティブなスクリーン・セーバの使用を中止します。
 - 「スタイル・マネージャ」の下に「画面」ダイアログボックスの「スクリーンセーバ」パネルで、「画面のブランク」を選択し、実行中のアクティブなスクリーン・セーバを選択解除します。
 - このダイアログボックスで、「オフ」を選択します。
 - 端末クライアント・ウィンドウから、`xset s off` コマンドを実行します。
- `graphics_suspend_dwell`
suspend DPMS モードの省略時の休止時間を分単位で指定します。この属性を無効にする場合には 0 を指定し、それ以外の場合には `graphics_standby_dwell` 値以上の値を指定します。
- `graphics_off_dwell`

off DPMS モードの省略時の休止時間を分単位で指定します。
この属性を無効にする場合は 0 を指定し、それ以外の場合は
graphics_standby_dwell および graphics_suspend_dwell 値
以上の値を指定します。

たとえば、power_mgr.stanza という名前の stanza ファイルを作成し、
属性に対して次のような値を定義することができます。

```
pwrmgr:
    default_pwrmgr_state=1
    cpu_slowdown=1
    disk_dwell_time=20
    disk_spindown=1
    graphics_powerdown=1
    graphics_standby_dwell=5
    graphics_suspend_dwell=10
    graphics_off_dwell=15
```

disk_dwell_time, graphics_standby_dwell,
graphics_suspend_dwell, graphics_off_dwell の各属性に対して指定
された値は、アイドル状態にあるハードウェアの電源を切断するまでの時間
(分単位)を示しています。この場合、電力管理サブシステムは 20 分待機した
後にディスクをスピンドアウンし、5、10、15 分待機した後、それぞれ DPMS
の standby, suspend, off モードにします。残りの属性については、値 1
が指定されていれば、その機能が有効に設定されていることを示しています。

stanza ファイルを作成して保存したのち、次のコマンドを入力して
/etc/sysconfigtab データベースを更新します。

```
# sysconfigdb -a -f power_mgr.stanza pwrmgr
```

stanza ファイル使用の詳細については、sysconfigdb(8) を参照して
ください。

3.5.2.2 実行中のカーネルまたは X サーバの変更

実行中のカーネルで属性の値を変更するには、sysconfig -r コマンドを使
用します。たとえば、次のように入力します。

```
# sysconfig -r pwrmgr cpu_slowdown=0
```

次の例に示すように、一度に 2 つ以上の属性を変更することができます。

```
# sysconfig -r pwrmgr \  
graphics_powerdown=1 graphics_standby_dwell=10
```

システム属性変更の詳細については、sysconfig(8) を参照してください。

DPMS モードおよび X サーバに関する値の変更についての詳細は、`xdec(1X)` および `xset(1X)` で説明している `dpms` スイッチを参照してください。

3.5.3 SysMan Station の使用

SysMan Station を使用すると、CPU やディスク装置などのシステム・エンティティをシステム・トポロジ・マップから選択することができます。

アイコン上で MB3 をクリックすると、選択した装置の管理アクションのリストを表示することができます。その 1 つに、電力管理アプリケーション `[dxdpower]` があります。このメニュー項目を選択すると、その装置に対して、`dxdpower` を実行することができます。

3.6 スワップ領域の追加

オペレーティング・システムは、物理メモリとディスク上のスワップ領域を組み合わせ、仮想メモリを実現します。仮想メモリは、物理メモリよりはるかに大きく設定することができます。したがって、仮想メモリを使えば、物理メモリだけの場合より多くのプロセスをサポートできます。この節では、スワップ領域を構成する際に考慮すべき、重要な仮想メモリ概念を説明します。

注意

仮想メモリ (vm) の不足を示すメッセージが表示される可能性があります。または、仮想メモリ不足のために、プロセスが強制終了させられる可能性があります。このような場合、仮想メモリが必ずしもスワップ領域を指しているわけではありません。`vm` カーネル・サブシステムが必要とするリソースが不足していることもあります。

スワップ領域が極端には使用されていず、スワップ領域の不足を明確に示すメッセージが表示されていない場合、プロセス毎のメモリを制限していないために問題が発生している可能性があります。詳細は、3.6.5 項を参照してください。

仮想メモリ (vm) カーネル・サブシステムは、物理メモリの一部、ディスクのスワップ領域、およびさまざまなデーモンとアルゴリズムを使って、プロセ

スへのメモリ割り当てを制御します。ページは、システムが割り当てることのできる最小の物理メモリ単位 (8 KB のメモリ) です。

仮想メモリでは、プロセスが最後に参照した仮想ページを物理メモリに保持しようとしています。プロセスが物理メモリにない仮想ページを参照すると、ディスクに保存されていた該当ページが物理メモリに取り込まれます。更新された仮想ページは、新たな仮想ページが参照されたり、より高い優先度を持つページが必要となったりして、その仮想ページの置かれた物理ページ (物理メモリ内のページ) を明け渡さなければならなくなると、ディスクの一時的な保存場所 (スワップ領域) に移されます。したがって、あるプロセスの仮想アドレス空間は、物理メモリに置かれているページ、一時的にスワップ領域に置かれているページ、および実行可能ファイルやデータ・ファイルとしてディスクに固定的に置かれているページから構成されることになります。仮想メモリの処理には、次の操作があります。

ページング ページを再使用するために再生する操作。

スワッピング 中断状態にあるプロセスの更新 (ダーティ) ページをスワップ領域に書き込む操作。この操作により、大きなメモリ領域が解放されます。

ページングでは、1 ページまたは数ページの仮想ページをディスクと物理メモリとの間で移動させます。プロセスが物理メモリにない仮想ページを参照した場合、オペレーティング・システムは、その仮想ページをディスクの固定領域またはスワップ領域から物理メモリに読み込みます。この操作をページインと呼びます。ページインは、通常、プロセスがイメージを新しく実行し、以前に参照したことのない実行可能イメージの場所を参照した場合に発生します。

物理ページに、新しく参照したページ、またはより高い優先度を持つページを置く必要がある場合、オペレーティング・システムは、更新されていても最近参照されていない仮想ページを 1 ページ (またはページ・クラスタ) だけスワップ領域に書き出します。この動作を更新ページ書き出しまたはページアウトと呼びます。更新されたページだけがスワップ領域に書き込まれます。これは、更新されていないページについては、同じものが常にディスクの固定領域にあるためです。

スワッピングでは、物理メモリとディスクの間で大量の仮想ページを移動させます。オペレーティング・システムが効率的に動作するためには、ある

程度の物理メモリが必要です。未使用の物理ページ数がシステムで定義されている下限を下回った場合に、個々の仮想ページまたはページ・クラスタをページアウトしても、システムが十分な物理メモリを再生できないときは、オペレーティング・システムは、低優先度のプロセスを選択し、それが使っているすべての物理メモリを再利用します。この操作をスワップアウトと呼びます。スワップアウトは、メモリに制約のあるシステムで典型的に発生します。

注意

システム・クラッシュ後にクラッシュ・ダンプをとる機能も、スワップ領域のサイズとどの程度利用できるかによって影響されます。スワップ領域の割り当てが十分でないと、エラー回復に役立つ情報が入ったクラッシュ・ダンプを保存することはできません。クラッシュ・ダンプについての詳細は、第 12 章を参照してください。

3.6.1 関連ドキュメントおよびユーティリティ

次のマニュアルとユーティリティは、スワップ領域を管理する際の参照資料です。

3.6.1.1 関連ドキュメント

スワップ領域を管理する方法は、以下のマニュアルで説明されています。

『インストレーション・ガイド — 上級ユーザ編』

初期スワップ領域を計画する方法と、オペレーティング・システムのインストレーション時に初期スワップ領域を設定する方法を説明しています。

『システムの構成とチューニング』

スワップ領域の構成にかかわる、性能チューニングのためのガイドラインなど、仮想メモリとスワップ領域に関する高度な概念を説明しています。

追加スワップ領域の作成についての詳細は、`swapon(8)` と `swapon(2)` を参照してください。関連ユーティリティのリファレンス・ページについては、3.6.1.2 項を参照してください。

3.6.1.2 関連ユーティリティ

次のユーティリティは、スワップ領域の管理の際に使用します。

| ユーティリティ | パス名 | 説明 |
|--------------------|----------------------------|---|
| Disk Configuration | /usr/sbin/diskconfig | このグラフィカル・ユーザ・インタフェースは、スワップ領域として割り当てることのできる未使用のパーティションを探すために使用します。このユーティリティを起動して使用方法については、 <code>diskconfig(8)</code> を参照してください。 |
| カーネル・チューナ | /usr/bin/X11/dxkerneltuner | このグラフィカル・ユーザ・インタフェースは、システム構成ファイルにあるカーネル・スワップ属性を修正するために使用します。このユーティリティを起動して使用方法は、 <code>dxkerneltuner(8)</code> を参照してください。 |

| ユーティリティ | パス名 | 説明 |
|-----------|-----------------|---|
| sysconfig | /sbin/sysconfig | このコマンド行インタフェースは、システム構成ファイルにあるカーネル・スワップ属性を設定するために使用します。システム属性を変更する方法は、sysconfig(8)を参照してください。 |
| disklabel | /sbin/disklabel | このコマンド行インタフェースは、システム構成ファイルにあるカーネル・スワップ属性を設定するために使用します。ディスクにラベル付け (ディスク・パーティションのフォーマットともいいます) する方法は、disklabel(8)を参照してください。 |

3.6.2 スワップ領域の割り当て

スワップ領域は、システムの要件に従って最初に計画し、システムのインストールの際に割り当てます。しかし、システムの性能を改善するためにスワップ領域を追加したり、システムに物理メモリを追加したい場合が出てきます。利用可能なスワップ領域がないという警告メッセージがシステム・コンソールに示され、スワップ領域の拡張が促されることもあります。しかし、スワップ領域を追加する前に、突然の領域不足がシステムの問題によるのではないことを確かめてください。次のコマンドを使用して、プロセスの暴走やユーザの異常な活動によって、スワップ領域が占有されていないことを確認してください。

```
# ps agx
```

(あるいは、システム・ログおよびイベント・ファイルでスワップのエラー・メッセージを調べます。) このコマンドから得られるリストが通常どおりなら、スワップ領域を追加する必要があります。

スワップ領域は、swapon コマンドを実行することで一時的に追加することができます。追加したスワップ領域を永久的なものにしたい場合、

/etc/sysconfigtab ファイルの `vm` セクションにエントリを追加する必要があります。 その手順は次のとおりです。

1. `swapon` コマンドを使用して、データを上書きすることがないことと、パーティションがオーバーラップしていないことを確認します。 ディスクを選択できる場合は、スワップ領域の場所として、恐らく、I/O が頻繁には行われない高速なディスクを選択することになるでしょう。たとえば、ユーザのファイルが置かれているディスクは I/O の頻度が高くなっています。

`diskconfig` ユーティリティを使って、ディスクを調べ、適切なパーティションを選択します。

2. 次に示す例のように `swapon` を実行し、スワップ領域を作成します。

```
# /sbin/swapon /dev/disk/dsk0b
```

たとえば、通常のパーシャル・ダンプの代わりにフル・クラッシュ・ダンプをとる場合などは、必要となる追加領域を一時的なスワップ領域ですませられるので、これ以上の操作は必要ありません。これで、スワップ・パーティションは、使用可能な状態になります。現在のスワップ構成を調べるために、次のコマンドを実行します。

```
# /sbin/swapon -s
```

必要ならば、ここでもう一度ステップ 1 に戻ってパーティションを追加することもできます。

3. 追加したスワップ領域を永久的なものにするために /etc/sysconfigtab ファイルの `vm` セクションを次のように編集します。

- 復元する必要がある場合に備えて、現在のファイルをコピーし、一時的な名前を付けて保存します。テキスト・エディタを使ってファイルをオープンし、`vm:` という文字列を探します。
- 初期スワップ領域としてインストール時に作成した `swapdevice=` というエントリがあります。ここに新しいスワップ・パーティションのデバイス特殊ファイル名を追加します。それぞれのスワップ・デバイスのエントリは、次のようにコンマで区切ります。

`vm:`

```
swapdevice=/dev/disk/dsk1b, /dev/disk/dsk3h  
vm-swap-eager=1
```

新しいスワップ・パーティションは、システムをリブートするかまたは次のコマンドを実行したときに自動的にオープンされます。

```
# /sbin/swapon -a
```

このコマンドがオーバーラップしたパーティションに対してどのように動作するかについては、`swapon(8)` を参照してください。

システムに必要なスワップ領域の大きさは、使用するスワップ領域割り当ての方法と、システムの負荷によって異なります。割り当ての方法を、次の項で説明します。

3.6.3 スワップ領域の必要量の見積り

スワップ領域割り当てには、2つの方法、つまり、即時モードと延期モード(オーバコミット・モード)があります。この2つの方法の違いは、いつスワップ領域が割り当てられるかにあります。即時モードでは、更新の可能性がある仮想アドレスが作成された時点でスワップ領域が割り当てられます。延期モードでは、スワップ領域は予約されず、更新された仮想ページをシステムがスワップ領域に書き込む必要が生ずるまで、割り当てられません。

注意

オペレーティング・システムは、更新された仮想ページをスワップ領域に書き込もうとする際に、スワップ領域が足りないと、プロセスを終了させます。

即時モードでは、更新の可能性がある仮想ページそれぞれについて、最初にスワップ領域のページを予約するため、見積りは延期モードよりも慎重なものになります。スワップ領域の割り当てに即時モードを用いる場合、少なくとも、システム上に作成されるアドレス空間のうち、更新の可能性がある空間の合計量に等しいスワップ領域を割り当てる必要があります。即時モードでは、更新の可能性がある仮想ページがいつ更新されても大丈夫なように十分なスワップ領域を保証するため、延期モードよりはるかに大きなスワップ領域を必要とします。

スワップ領域の割り当てに延期モードを用いる場合、作成され更新される仮想アドレス空間の総量を見積り、この総量とシステムの物理メモリサイズを比較しなければなりません。この総量が、物理メモリの半分より大きい場合は、更新の可能性があるとしても物理メモリに入りきらない仮想ページを収容

できるだけ十分大きなスワップ領域を確保しなければなりません。システムの負荷が複雑で、この方法では適切なスワップ領域の量を見積ることができない場合は、まず省略時のスワップ領域を使ってみてから、必要に応じてスワップ領域を調整していく方法を用います。物理メモリの約半分の量のスワップ領域を使用してみてください。

常にシステムのスワップ領域の使用状況を監視する必要があります。スワップ領域が枯渇しつつあるというメッセージをシステムが出した時点で、`swapon` コマンドを使って、スワップ領域を追加して割り当てます。即時モードを使っている場合は、スワップ領域が枯渇すると、更新の可能性がある仮想アドレス空間は新たに作成することができません。延期モードを使っている場合は、スワップ領域が枯渇すると、1 つ以上のプロセスが強制的に終了させられることがあります。

仮想メモリについての詳細は、『システムの構成とチューニング』を参照してください。

3.6.4 スワップ領域割り当て方法の選択

どのスワップ領域割り当て方法を使用しているかは、`/etc/sysconfigtab` の `vm:` セクションを見て調べることができます。また、`dxkerneltuner` または `sysconfig` ユーティリティを使って、カーネルの属性値を調べることもできます。次のようなエントリが表示されるはずです。

```
vm:
    swapdevice=/dev/disk/dsk1b, /dev/disk/dsk3h
    vm-swap-eager=1
```

`vm-swap-eager=` のエントリによって、次のように割り当て方法を調べることができます。

`vm-swap-eager=1` システムは即時スワップ・モードを使用している。

`vm-swap-eager=0` システムは、延期スワップ・モードを使用している。

`/etc/sysconfigtab` ファイルを修正して現在の値を変更するか、`dxkerneltuner` または `sysconfig` ユーティリティを使って、動的に属性を変更します。

システムをリブートし、新しいスワップ方法を有効にします。延期モードに切り替えた場合、または延期方式を使っているシステムをリブートした場合、次のメッセージがリブート時に表示されることがあります。

```
vm_swap_init: warning/sbin/swapdefault swap device not found
vm_swap_init: in swap over-commitment mode
```

3.6.5 明白なスワップ領域不足の修正

個々のプロセスが使用できる仮想メモリの量には、制限があります。これらの制限は、利用可能なスワップ領域の合計量とは関係ありません。このため、スワップ・モニタ (たとえば `dxsysinfo` ユーティリティ) でスワップ領域不足が表示されなくても、プロセスの仮想メモリが足りないというエラー・メッセージが表示される可能性があります。割り当てられた仮想メモリを超えると、プロセスが自動的に強制終了されることがあります。

スワップ割り当てモードの効果についての詳細は、`sys_attrs_vm(5)` の `vm_swap_eager` 属性の説明を参照してください。

スワップ領域が適切であるかどうかを検証するには、次のコマンドを使います。

```
# swapon -s
```

「In-use space:」というタイトルのデータ・フィールドは、利用可能なスワップ領域の大半が使用されているかどうかを示します。

利用可能なスワップ領域を使い切っていないのに、大きなプロセスの実行に問題がある場合は、プロセスに対してさらにリソースを割り当てる必要があります。proc サブシステムには、いくつかのカーネル属性があり、プロセス毎の仮想メモリのリソースを制御するために使用することができます。

| | |
|--------|--|
| スタック制限 | <code>per-proc-stack-size</code> 属性と <code>max-per-proc-stack-size</code> 属性。 |
|--------|--|

| | |
|-------|--|
| データ制限 | <code>per-proc-data-size</code> 属性と <code>max-per-proc-data-size</code> 属性。 |
|-------|--|

| | |
|--------|--|
| アドレス空間 | <code>max-per-proc-address-space</code> 属性と <code>per-proc-address-space</code> 属性。 |
|--------|--|

メモリ不足による問題が発生した場合は、次の方法の使用を検討してください。

- `ksh(1)` など、ご使用中のコマンド・シェルを説明しているリファレンス・ページを参照してください。コマンド・シェルには、`limit` や

`ulimit` オプションがあり、プロセスの仮想メモリ・リソースを変更することができます。

- `sysconfigdb` コマンドまたはカーネル・チューナ GUI を使い、`/etc/sysconfigtab` ファイル内のプロセス毎のリソース制限の値を変更します。カーネル属性の変更の説明は、第 4 章にあります。
- 『システムの構成とチューニング』を参照して、`vm-maxvas` のような仮想メモリ (vm) サブシステムの属性を調整する方法を調べてください。システム属性についての詳細は、`sys_attrs(5)` を参照してください。



カーネルの構成

この章では、インストレーション中およびインストレーション後のカーネル構成と、動的構成および静的構成について説明します。この章で説明する情報は、以下のとおりです。

- カーネル構成の概要 (4.1 節)
- カーネル・サブシステムで使用可能なすべての属性を説明した各リファレンス・ページなどの、関連ドキュメントの参照先 (4.2 節)
- インストレーション時のカーネル構成 (4.3 節)
- カーネルを構成する時期の決定 (4.4 節)
- カーネル・チューナのグラフィカル・ユーザ・インタフェース
`/usr/bin/X11/dxkerneltuner` などの、動的システム構成 (4.5 節)
- 静的システム構成 (4.6 節)
- 構成ファイル (4.7 節)

4.1 概要

オペレーティング・システムのカーネルは、メモリ常駐の実行可能イメージであり、ハードウェア割り込み、メモリ管理、プロセス間通信、プロセス・スケジューリングなど、すべてのシステム・サービス进行处理するとともに、他の作業をオペレーティング・システム上で実行可能にします。これらのコア・サービスをサポートするコードに加えて、カーネルには数多くのサブシステムが含まれています。

サブシステムはカーネル・モジュールであり、コア・カーネル・サービスの範囲を超えてカーネルを拡張します。ファイル・システムやネットワーク・プロトコル・ファミリ、物理ドライバおよび擬似デバイス・ドライバはすべて、サポートされているサブシステムの例です。サブシステムの中には、カーネルで必須のものあれば、オプションのものもあります。カーネルの構成は、これらオプションのサブシステムを追加したり、削除することによって行います。この構成は、インストレーションのときに行うこ

ともできますし、インストール後にシステムを変更する必要がある場合に行うこともできます。

この他のカーネルの構成方法は、カーネル内に格納されている値を調整することです。たとえば、カーネル内の値には、ディスク・アクセスを速くするように調整できるものがあります。値を変更して、ディスク・アクセスを最適化すると、システムの性能を改善することができますが、他の性能に影響を及ぼすことがあります。システムのチューニングおよび属性値の影響についての詳細は、『システムの構成とチューニング』に説明しています。

カーネルの構成には、動的メソッドと静的メソッドの2つがあります。

| | |
|--------|---------------------------------|
| 動的メソッド | コマンドを使用して、カーネルの実行中に、カーネルを構成します。 |
|--------|---------------------------------|

| | |
|--------|-----------------------------|
| 静的メソッド | システム・ファイルを変更して、カーネルを再構築します。 |
|--------|-----------------------------|

システム・ファイルを変更してカーネルを再構築する処理は、多くの場合、難しいため、可能であれば動的カーネル構成を使用してください。すべての変更が動的に行えるとは限りません。また、動的な変更は、システムのリブート後も持続するわけではありません。

4.2 関連ドキュメントとユーティリティ

次に、システム属性、構成ツールとユーティリティについての情報、および構成オプションの詳細な参照情報を示します。以下の項で説明する資料の形式は、マニュアル、リファレンス・ページ、およびオンライン・ヘルプです。

4.2.1 マニュアル

Tru64 UNIX オペレーティング・システムのドキュメント・セットにある以下のマニュアルでは、システム構成が説明されています。

- 『インストール・ガイド』および『インストール・ガイド — 上級ユーザ編』では、インストール時のカーネルの初期構成について説明しています。
- 『ネットワーク管理ガイド：接続編』と『ネットワーク管理ガイド：サービス編』では、ネットワークの構成について説明しています。

- 『システムの構成とチューニング』では、システムの構成とチューニングについて詳細に説明しています。

4.2.2 リファレンス・ページ

ここで示すリファレンス・ページでは、システム属性とユーティリティについて説明しています。

`sys_attrs(5)`

システム属性を説明するとともに、`streams` や `socket` など、各カーネル・サブシステムについて説明しているいくつかの `sys_attrs*` リファレンス・ページを参照先として示しています。構成可能な属性を持たないサブシステムがいくつかありますが、ここには挙げていません。

注意

属性値を変更する前に、該当するリファレンス・ページと『システムの構成とチューニング』を参照してください。

`sys_attrs_vm(5)`

カーネル構築時に必須となるサブシステムの属性について説明しています。これらの必須サブシステムには、構成マネージャ (`cm`)、汎用カーネル (`generic`)、プロセス間通信 (`ipc`)、プロセス (`proc`)、仮想ファイル・システム (`vfs`)、および仮想メモリ (`vm`) があります。

`sys_attrs_advfs(5)`

Advanced File System (`advfs`) カーネル・サブシステムの属性について説明しています。

`sys_attrs_atm(5)`

Asynchronous Transfer Mode (ATM) カーネル・サブシステムの属性について説明し

ています。これらのカーネル・サブシステムには、Base ATM support (atm)、ATM Forum Integrated Layer Management Interface (atmilmi3x)、Classical IP services (atmip)、ATM Forum signaling と Integrated Layer Management Interface support (atmuni)、ATM Forum LAN Emulation (lane)、および ATM Forum signaling (uni3x) があります。

sys_attrs_*(5)

各サブシステムの属性について説明しています。以下の表には、これらのカーネル・サブシステムのリファレンス・ページを示していますが、調整可能な属性を持たないカーネル・サブシステムは省いてあります。

| リファレンス・ページ | 説明 |
|----------------------|--|
| sys_attrs_ace(5) | シリアル・デバイス (ace) カーネル・サブシステム |
| sys_attrs_alt(5) | ギガビット・イーサネット・アダプタで 用される alt カーネル・サブシステム |
| sys_attrs_autofs(5) | AutoFS カーネル・サブシステム |
| sys_attrs_bparm(5) | ブート・パラメータ・カーネル・サブシステム |
| sys_attrs_bsd_tty(5) | BSD 端末カーネル・サブシステム |
| sys_attrs_cam(5) | SCSI CAM I/O カーネル・サブシステム |
| sys_attrs_cm(5) | 構成管理カーネル・サブシステム |
| sys_attrs_dli(5) | データ・リンク・インタフェース・ カーネル・サブシステム |
| sys_attrs_dlpi(5) | データ・リンク・プロバイダ・インタフェー ス・カーネル・サブシステム |
| sys_attrs_ee(5) | 10/100 MB イーサネット・アダプタで使 用される、ee カーネル・サブシステム |
| sys_attrs_eisa(5) | EISA バス・カーネル・サブシステム |

| リファレンス・ページ | 説明 |
|------------------------|---|
| sys_attrs_fta(5) | FDDI (Fiber Distributed Data Interface) アダプタで使用する fta カーネル・サブシステム |
| sys_attrs_generic(5) | 汎用カーネル・サブシステム |
| sys_attrs_gpc_input(5) | GPC インプット・カーネル・サブシステム |
| sys_attrs_io(5) | 入出力カーネル・サブシステム |
| sys_attrs_ipc(5) | プロセス間通信カーネル・サブシステム |
| sys_attrs_iptunnel(5) | インターネット・プロトコル・トンネリング・カーネル・サブシステム |
| sys_attrs_ipv6(5) | インターネット・プロトコル・バージョン 6 のカーネル・サブシステム |
| sys_attrs_isa(5) | ISA バス・カーネル・サブシステム |
| sys_attrs_kevm(5) | カーネル・イベント管理カーネル・サブシステム |
| sys_attrs_lag(5) | リンク・アグリゲーション・グループ・カーネル・サブシステム |
| sys_attrs_lfa(5) | lfa カーネル・サブシステム |
| sys_attrs_lsm(5) | Logical Storage Manager カーネル・サブシステム |
| sys_attrs_net(5) | ネットワーク・カーネル・サブシステム |
| sys_attrs_netrain(5) | NetRAIN (Redundant Array of Independent Network) アダプタ・カーネル・サブシステム |
| sys_attrs_pci(5) | PCI カーネル・サブシステム |
| sys_attrs_ppp(5) | ポイント・ツー・ポイント・プロトコル・カーネル・サブシステム |
| sys_attrs_presto(5) | PrestoServe カーネル・サブシステム |
| sys_attrs_proc(5) | プロセス・カーネル・サブシステム |
| sys_attrs_psm(5) | プロセス・セット・マネージャ・カーネル・サブシステム |
| sys_attrs_pts(5) | 擬似端末カーネル・サブシステム |
| sys_attrs_pwrmgr(5) | 電力管理 (pwrmgr) カーネル・サブシステム |
| sys_attrs_rt(5) | リアルタイム・カーネル・サブシステム |

| リファレンス・ページ | 説明 |
|--|--|
| <code>sys_attrs_scc(5)</code> | シリアル・ドライバ・カーネル・サブシステム |
| <code>sys_attrs_scc_input(5)</code> | シリアル・ドライバ・キーボード・ドライバ・カーネル・サブシステム |
| <code>sys_attrs_sec(5)</code> | セキュリティ・カーネル・サブシステム |
| <code>sys_attrs_snmp_info(5)</code> | SNMP (Simple Network Management Protocol) カーネル・サブシステム |
| <code>sys_attrs_socket(5)</code> | ソケット・カーネル・サブシステム |
| <code>sys_attrs_streams(5)</code> | STREAMS カーネル・サブシステム |
| <code>sys_attrs_tc(5)</code> | TURBOchannel バス・カーネル・サブシステム |
| <code>sys_attrs_tu(5)</code> | 10/100 MB イーサネット・アダプタで使われる <code>tu</code> カーネル・サブシステム |
| <code>sys_attrs_ufs(5)</code> | UFS (UNIX ファイル・システム) カーネル・サブシステム |
| <code>sys_attrs_uipc(5)</code> | AF_UNIX プロセス間通信カーネル・サブシステム |
| <code>sys_attrs_vfs(5)</code> | 仮想ファイル・システムのカーネル・サブシステム |
| <code>doconfig(8)</code> | 現在のシステム構成ファイルで指定されている設定に従ってカーネルを構築するためのユーティリティについて説明しています。 |
| <code>kopt(8)</code> | カーネル・オプションを選択するためのユーティリティについて説明しています。 |
| <code>sysconfig(8)</code> , <code>sysconfigtab(4)</code> , および <code>sysconfigdb(8)</code> | コマンド行ユーティリティと、カーネル・サブシステムの構成管理やカーネル・サブシステム属性の変更または表示に使用するデータベースについて説明しています。 <code>sysconfigtab</code> コマンドでは、構成データベースのファイル形式について説明しています。 <code>sysconfigdb</code> ユーティ |

リティを使用して、この構成データベースを管理してください。

`sysconfigdb(8)` と `stanza(4)`

サブシステム構成データベースを管理するコマンド行ユーティリティについて説明しています。`stanza` コマンドでは、スタンザ・ファイルの構成形式について説明しています。このファイルの内容は、`sysconfigdb` を実行したときに構成データベースに組み込まれます。

`autosysconfig(8)`

自動構成の対象にする一連の動的カーネル・サブシステムを管理するユーティリティについて説明しています。

`cfgmgr(8)`

`sysconfig` および他のユーティリティがカーネル・サブシステムを管理するために使用するサーバについて説明しています。カーネル・ロード・サーバについて説明している `kloadsrv(8)` リファレンス・ページも参照してください。

`dxkerneltuner(8)`

カーネル・サブシステムの属性を変更したり表示したりするグラフィカル・ユーザ・インタフェース (カーネル・チューナ) について説明しています。

`sys_check(8)`

さまざまなシステム属性を検査し、それら属性の適正値を表示する、`sys_check` ユーティリティについて説明しています。詳細については、第 3 章 も参照してください。

4.2.3 オンライン・ヘルプ

カーネル・チューナ・グラフィカル・ユーザ・インタフェースには、独自のオンライン・ヘルプがあります。

4.3 インストール時のシステム構成

オペレーティング・システムをインストールすると、インストール・プログラムは最初にシステム・ディスクのルート・パーティションにカーネル・イメージをコピーします。汎用カーネルとも呼ばれるこのカーネル・イメージは、オペレーティング・システムの現在のバージョンで使用するすべてのプロセッサおよびハードウェア・オプションをサポートしています。このように、どのような構成であっても、システムをブートできることをインストール・プログラムは保証します。汎用カーネルのファイルは、`/genvmunix` です。

選択したすべてのサブセットがディスクに書き込まれ、その確認が終わると、インストール処理の最後に、インストール・プログラムは `/usr/sbin/doconfig` プログラムを呼び出します。呼び出された `/usr/sbin/doconfig` プログラムは `sizer` という別のプログラムを呼び出します。`sizer` プログラムは、システムにインストールされているハードウェアおよびソフトウェア・オプションを判断して、システム固有のターゲット構成ファイルを作成します。この構成ファイルは、カーネルにリンクされるハードウェアおよびソフトウェア・サポートを制御するシステム・ファイルです。その後、このターゲット構成ファイルをもとに、`/usr/sbin/doconfig` プログラムがカスタマイズした `/vmunix` カーネルを構築します。このカーネルの構築では、すべてのサブシステム属性に対してその省略値が使用されます。

インストール時にシステムにコピーされる汎用カーネルとは異なり、ターゲット・カーネルは、使用するシステムに合わせてカスタマイズされています。システムで利用できるハードウェアおよびソフトウェア・オプションだけがコンパイルされて、ターゲット・カーネル内に組み込まれます。このため、ターゲット・カーネルは汎用カーネルよりも小さく、かつ効率的なものになります。

インストールが終了すると、システムの構築方法により、ターゲット・カーネルは、メモリまたはシステム・ディスクのルート・パーティションのいずれかに存在することになります (カーネルの異なる構築方法については、4.6 節を参照)。コンソール・ブート変数を適切に設定していれば、システムは自動的にターゲット・カーネルをブートします。コンソール・ブート変数の設定方法と使用方法については第 2 章を参照してください。

4.4 カーネルを再構成する時期と方法の決定

インストレーション・プロシージャによりターゲット・カーネルが構築されて起動されると、次のような状況が起こらないかぎり、そのカーネルを変更せずに使用することができます。

- 新しいデバイスのインストールまたは Asynchronous Transfer Mode (ATM) のような追加オプションの使用など、カーネルに新しいサブシステムを追加するとき
- デバイスや Logical Storage Manager (LSM) 機能の削除など、カーネルからサブシステムを削除するとき
- システムの性能が良くない(集中型のアプリケーションを実行しているなど)ため、カーネルの属性値を省略時の値から変更するとき。このような集中型のアプリケーションには、インターネット Web サーバやデータベースなどがあります。システムをチューニングする前に、カーネル属性を変更したときの影響を十分に理解しておく必要があります。この影響を理解することで、使用できないカーネルを作成したり、システムの性能を低下させることがなくなります。

たとえば、通常のシステム監視の 1 つの操作として、`sys_check` ユーティリティを実行します。`sys_check` がシステムの使用状況の分析をもとに生成したレポートによって、カーネル属性の新しい値または追加サブシステムのローディングが提示されることもあります。しかし、属性値の変更は、『システムの構成とチューニング』でシステム性能以外に及ぼす影響を調べてから行ってください。

ほとんどのデバイスはシステムによって自動的に認識され、ブート時にカーネルへ組み込まれます。デバイスの追加についての詳細は、『ハードウェア管理ガイド』を参照してください。ただし、サードパーティ製のディスク・ドライブ、古いタイプのドライブやスキャナや PCMCIA カードなどのデバイスは手動で追加する必要があります。これらのデバイスを追加する場合は、動的であれ静的であれ、カーネルを再構成する必要があります。カーネルの再構成に使用するメソッドは、サブシステムまたはサブシステム属性によって提供されるサポートにより異なります。

`envmon` 環境モニタリング・サブシステムなどのカーネル・サブシステムは、動的にロードすることができます。つまり、カーネルを再構築しなくても、カーネルに対してサブシステムの追加や削除を行うことができます。

また、動的にロード可能なサブシステムのほとんどは、その属性値を動的に構成することができます。したがって、カーネルを再構築しなくても、これらのサブシステムの性能を調整することができます。属性が動的に構成可能かどうかを調べる場合は、`sysconfig` で `-m` を使い、次のように動的識別子を検索します。

```
# sysconfig -m | grep dynamic
lat: dynamic
envmon: dynamic
hwautoconfig: dynamic
```

カーネルに対してこれらのサブシステムの追加や削除を行ったり、属性の値を構成したりする場合には、4.5 節で説明する手順に従ってください。

必須サブシステムなど、一部のサブシステムは動的にロードできません。ただし、これらのサブシステムでは、属性の値を動的に構成することができます。その場合、これらのサブシステムの属性値は、カーネルを再構築しなくても構成できます。

次のメソッドを使用して、属性を動的に構成できます。

- `sysconfig -r` コマンドを使用して、実行中のカーネルの属性値を構成する。この実行時構成をサポートしているのは、数個のカーネル・サブシステムだけです。
- グラフィカル・ユーザ・インタフェースである、カーネル・チューナ (`dxkerneltuner`) を使用する。このユーティリティは、`sysconfig` とほぼ同じ表示機能および設定機能を持っています。このユーティリティは、コマンド行で次のように起動します。

```
# /usr/bin/X11/dxkerneltuner
```

または、CDE フロント・パネルでアプリケーション・マネージャをオープンし、「モニタリング/チューニング」フォルダを選択して、フォルダが開いたら、「カーネル・チューナ」アイコンを選択します。カーネル・チューナの使用についての詳細は、`dxkerneltuner(8)` リファレンス・ページおよびアプリケーションのオンライン・ヘルプを参照してください。

カーネル・チューナ GUI は、使用可能なカーネル・サブシステムをすべてメイン・ウィンドウに表示します。サブシステムを選択して、その属性とともに、各属性の現在値、最大値、および最小値を表示させます。

変更可能な属性がある場合は、その属性値がテキストの入力フィールドに表示されます。このフィールドに新しい属性値を入力します。

- 動的サブシステム・データベース `/etc/sysconfigtab` 内の属性値を構成する。新しい属性値でカーネルを実行したい場合は、新しいカーネルを指定してシステムをリブートします。

これらのサブシステムの属性を構成する場合には、4.5.8 項で説明する手順に従ってください。`/etc/sysconfigtab` などのシステム・ファイルは、手動で編集しないことをお勧めします。代わりに、`dxkerneltuner` などの、コマンドやユーティリティを使用して変更を行ってください。

デバイス・ドライバまたはその他のカーネル・サブシステムをサードパーティのベンダから購入した場合は、その製品は動的ロードが可能であったり、属性値の動的構成が可能なことがあります。他のベンダの製品と一緒に使用する場合に、カーネルを動的に構成する方法については、その製品に付属のマニュアルおよび 4.5 節を参照してください。

追加、削除、または構成を行うサブシステムが動的な構成をサポートしていない場合には、静的な構成メソッドを使用しなければなりません。また、動的な構成をサポートしていないシステム・パラメータを構成する場合にも、このメソッドを使用する必要があります。静的構成メソッドについては、4.6 節を参照してください。

4.5 動的システム構成

動的サブシステムのロード、アンロード、変更を行う必要がある場合は、`/sbin/sysconfig` コマンドを使用します。このコマンドの構文は次のとおりです。

```
/sbin/sysconfig [-h hostname] [-i index]  
[-v | -c | -d | -m | -o | -q | -Q | -r | -s | -u] [subsystem-name]  
[attribute-list | opcode]
```

サブシステムのロードおよびアンロードを行うには、スーパーユーザでなければなりません。また、管理するサブシステムの名前も知っておく必要があります。サブシステムの名前は、そのサブシステムに付属のマニュアルを参照するか、またはサブシステムがインストールされているディレクトリを見て調べてください。サブシステムは、ディレクトリ `/subsys` または `/var/subsys` のいずれかにインストールされます。サブシステムがインストールされると、`subsystem-name.mod` という名前のファイルが、これ

ら 2 つのディレクトリのいずれかに格納されます。そのサブシステム名を `/sbin/sysconfig` コマンドの入力として使用してください。以降の各項で、サブシステムの管理に使用するコマンドについて説明します。

サブシステムのロードおよびアンロードは、ローカル・システム上でもリモート・システム上でも行うことができます。リモート・システム上でサブシステムの追加および削除を行う方法については、4.5.7 項を参照してください。

ロード可能なデバイス・ドライバや他のロード可能なサブシステムを記述している場合には、そのデバイス・ドライバのマニュアルおよび『プログラミング・ガイド』を参照してください。デバイス・ドライバのマニュアルでは、ロード可能なデバイス・ドライバをインストールした場合に、システムが実行するタスクについて説明しています。また、これらのマニュアルでは、ロード可能なデバイス・ドライバの作成およびパッケージ方法についても説明しています。『プログラミング・ガイド』では、動的に構成できるサブシステムの作成に関する一般情報を記載するとともに、サブシステムおよび属性の動的構成をサポートするフレームワークについても説明しています。

4.5.1 サブシステムの構成

サブシステムを構成(ロード)するには、`-c` オプションを使用して `sysconfig` コマンドを入力します。このコマンドは、新しくインストールしたサブシステム、または `/sbin/sysconfig -u` (構成除外オプション) コマンドで削除したサブシステムを構成する場合に使用します。たとえば、環境モニタリング (`envmon`) サブシステムを構成する場合、次のコマンドを入力します。

```
# /sbin/sysconfig -c envmon
```

4.5.2 構成サブシステムのリスト

カーネルがサブシステムを認識することができても、使用できるとは限りません。使用できるサブシステムを判断するには、`/sbin/sysconfig -s` コマンドを使用します。このコマンドを実行すると、すべてのサブシステムの状態が表示されます。サブシステムは次のいずれかの状態をとります。

- ロードされて構成されている (使用可能)
- ロードされているが構成されていない (使用できないがロードされている)
この状態は、静的サブシステムが構成除外できるが、アンロードできない場合のみ適用されます。
- アンロード (使用不可)

この状態は、システム管理者によって構成除外されると自動的にアンロードされるロード可能サブシステムについてのみ適用されます。

サブシステム名を指定しないで `/sbin/sysconfig -s` コマンドを使用すると、構成されているサブシステムがすべて表示されます。たとえば、次の例を参照してください。

```
# /sbin/sysconfig -s
cm: loaded and configured
hs: loaded and configured
ksm: loaded and configured
generic: loaded and configured
io: loaded and configured
ipc: loaded and configured
proc: loaded and configured
sec: loaded and configured
socket: loaded and configured
rt: loaded and configured
advfs: loaded and configured
.
.
.
envmon: unloaded
```

このリスト (途中から省略) には、静的にリンクされているサブシステムと動的にロードされているサブシステムの両方が含まれています。

特定のサブシステムの状態情報を取得するには、次のように、コマンド行にサブシステム名を指定します。

```
# /sbin/sysconfig -s lsm
lsm: unloaded
```

4.5.3 サブシステムのタイプの判断

サブシステムが動的にロード可能であるか、または静的なサブシステムであるかを判断するには、`/sbin/sysconfig -m` コマンドを次のように使用します。

```
# /sbin/sysconfig -m kinfo lat
kinfo: static
lat: dynamic
```

このコマンドの実行結果から、`kinfo` サブシステムは静的であることがわかるため、カーネルに対してそのサブシステムの追加や削除を行うには、カーネルを再構築する必要があります。一方、`lat` サブシステムは動的で

す。このサブシステムは、`sysconfig -c` コマンドで構成し、`sysconfig -u` コマンドで構成除外を行うことができます。

4.5.4 サブシステムのアンロード

サブシステムの構成除外（および、おそらくアンロード）を行うには、次のように `/sbin/sysconfig -u` コマンドを使用します。

```
# /sbin/sysconfig -u hwautoconfig
```

デバイス・ドライバを頻繁に構成したり構成除外する場合には、特定のデバイス・ドライバに関連するデバイス特殊ファイルがその都度異なることに注意してください。これは正常な動作です。`/sbin/sysconfig` コマンドを使用してデバイス・ドライバを構成すると、システムはデバイス特殊ファイルを作成します。そのデバイス・ドライバをアンロードして、同じ `cdev` または `bdev` 主デバイス番号を使用する別のデバイス・ドライバをロードすると、システムは、アンロードされたデバイス・ドライバ用のデバイス特殊ファイルを削除します。したがって、次にそのデバイスを構成するときに、システムは新しいデバイス特殊ファイルを作成する必要があります。デバイス特殊ファイルについての詳細は、『ハードウェア管理ガイド』と `dsfmgr(8)` リファレンス・ページを参照してください。

4.5.5 自動構成サブシステムのリストの保守

システムは、自動構成サブシステムのリストを検査することにより、リブート時にカーネル内で構成するサブシステムを決定します。システムは、システム・リブート時に `sysconfig -c` コマンドを使用し、そのリストにある各サブシステムについて構成を行います。

自動構成サブシステムのリストの保守を行うには、`/sbin/init.d/autosysconfig` コマンドを使用します。

このコマンドの構文は次のとおりです。

```
/sbin/init.d/autosysconfig [list] [add subsystem-name] [delete  
subsystem-name]
```

リブート時にシステムが自動的に構成するロード可能なサブシステムのリストを確認するには、`/sbin/init.d/autosysconfig list` コマンドを使用します。

サブシステムをリストに追加するには、`/sbin/init.d/autosysconfig add` コマンドを使用します。たとえば、`lat` サブシステムを追加するには、次のコマンドを入力します。

```
# /sbin/init.d/autosysconfig add lat
```

自動構成サブシステムのリストにあるサブシステムをアンロードする場合は、そのサブシステムをリストから削除する必要があります。削除しない場合、システムは、次のリポート時にそのサブシステムをカーネルに戻して、構成を行います。サブシステムを自動構成サブシステムのリストから削除する場合は、`/sbin/init.d/autosysconfig delete` コマンドを使用します。たとえば、`lat` サブシステムを削除するときは、次のコマンドを入力します。

```
# /sbin/init.d/autosysconfig delete lat
```

4.5.6 サブシステム属性の管理

サブシステムやシステム全体の性能または動作を改善するために、サブシステム属性の値を変更することがあります。これらの変更には、`sysconfig`、`sysconfigdb` またはカーネル・チューナ GUI を使用します。また、クラッシュしたシステムの回復などの特定の状況では、デバッガ `dbx` を使用して、クラッシュしたカーネル内の属性を調べて変更する必要があります。この手順の詳細については、『*Kernel Debugging*』を参照してください。

実行時に属性を変更すると、その変更は、システムが実行している間だけ持続します。システムをシャットダウンしてリブートすると、変更は失われます。サブシステム属性の変更がリブート後も持続するようにするには、4.5.8 項で説明するように、その属性の変更値を `/etc/sysconfigtab` データベースに格納する必要があります。変更した属性値の持続性は、使用したコマンドまたはユーティリティのオプションによって異なります。ガイドラインを次に示します。

- リブートした後も変更を持続させるには、コマンド行で `sysconfigdb` (または `dbx`) を使用します。または、カーネル・チューナ GUI を使用して、変更を「ブート時の値」フィールドに保存します。
- リブート後にその変更が失われる一時的な変更の場合は、コマンド行で `sysconfig -r` を使用します。または、カーネル・チューナ・グラフィカル・ユーザ・インタフェースを使用して、現在の属性値を変更します。

注意

以前のオペレーティング・システムのリリースでは、
/etc/sysconfigtab ファイルは、vi などのテキスト・エディタで変更可能なシステム・ファイルとして記述されています。最近のリリースでは、アップデート・インストールおよびカーネルが変更を認識するうえで、サブシステム・スタンザの管理が重要になってきました。/etc/sysconfigtab の構造を正しく維持するために、sysconfigdb コマンドまたはカーネル・チューナ・グラフィカル・ユーザ・インタフェースだけを使用して変更を行います。

詳細については、sysconfig(8)、sysconfigdb(8)、sysconfigtab(4)、および dxkerneltuner(8) を参照してください。

システム構成ファイルに設定されたシステム・パラメータは、システム・テーブルを定義し、カーネルの省略値とみなされます。必要な場合には、/sbin/sysconfig コマンドを使用するか、または値を/etc/sysconfigtab データベースに格納することによって、これらの値を上書きすることができます。構成ファイル (および param.c) についての詳細は、4.6 節を参照してください。

動的サブシステム属性は、ローカルでもリモートからでも管理することができます。/sbin/sysconfig コマンドをリモートから使用する方法については、4.5.7 項を参照してください。

4.5.6.1 サブシステム属性の現在値の確認

/sbin/sysconfig -q コマンドまたは dxkerneltuner を使用して、サブシステム属性に割り当てる値を確認することができます。/sbin/sysconfig -q コマンドを入力するとき、コマンド行で指定するサブシステムはロードされて構成されていなければなりません。ロードされて構成されているサブシステムをリスト表示する方法については、4.5.2 項を参照してください。

次の例は、このコマンドを使用して、generic サブシステムに属する属性を調べる方法を示しています。

```
# /sbin/sysconfig -q generic
generic:
```

```

booted_kernel = vmunix
booted_args = vmunix
lockmode = 0
lockdebug = 0
locktimeout = 15
max_lock_per_thread = 16
lockmaxcycles = 0
rt_preempt_opt = 0
cpu_enable_mask = 0x1
binlog_buffer_size = 0
msgbuf_size = 32768
dump_sp_threshold = 4096
use_faulty_fpe_traps = 0
partial_dump = 1
make_partial_dumps = 1
compressed_dump = 1
make_compressed_dumps = 1
expected_dump_compression = 500
expect_1000b_to_compress_to = 500
dump_to_memory = 0
dump_allow_full_to_memory = 0
leave_dumps_in_memory = 0
dump_user_pte_pages = 0
live_dump_zero_suppress = 1
live_dump_dir_name = /var/adm/crash
include_user_ptes_in_dumps = 0
lite_system = 0
physio_max_coalescing = 65536
kmem_percent = 25
kmemreserve_percent = 0
kmem_debug = 0
kmem_debug_size_mask = 0
kmem_protected_size = 0
kmem_protected_lowat = 1000
kmem_protected_hiwat = 0
kmem_protected_kmempercent = 75
kmem_audit_count = 1024
kmemhighwater_16 = 4
.
.
.
kmemhighwater_12k = 4
old_obreak = 1
user_cfg_pt = 45000
memstr_buf_size = 0
memstr_start_addr = 0
memstr_end_addr = 0
memlimit = 0
insecure_bind = 0
memberid = 0

```

```

memberseq = 0
clu_configured = 0
clu_active_member = 0
old_vers_high = 0
old_vers_low = 0
act_vers_high = 1441151880873377792
act_vers_low = 15044
new_vers_high = 1441151880873377792
new_vers_low = 15044
versw_switch = 0
versw_transition = 0
rolls_ver_lookup = 0
login_name_max = 12
enable_async_printf = 1

```

(ここでは、途中で表示を省略してあります。)

4.5.6.2 実行時に構成可能なサブシステム属性の識別

/sbin/sysconfig -Q コマンドを使用すると、どのサブシステムの属性が実行時に構成できるかを識別することができます。次の例を参照してください。

```

# /sbin/sysconfig -Q vfs max-vnodes
vfs:
max-vnodes -      type=INT op=CRQ min_val=0 max_val=1717986918

```

この例では、-Q オプションを使用して、vfs サブシステムの max-vnodes 属性に関する情報を取得しています。max-vnodes 属性は integer データ型であり、最小値が 0 で最大値が 1717986918 になります。op フィールドは、max-vnodes 属性で実行できるオペレーションを示しています。次に、このフィールドに表示される値を示します。

- C サブシステムが最初にロードされるときに、属性を変更できる。
- R サブシステムが動作しているときに、属性を変更できる。
- Q 属性を照会できる。

4.5.6.3 属性値の実行時における変更

属性の値を実行時に変更するには、/sbin/sysconfig -r コマンド、dxkernel tuner、またはソース・レベル・デバッガ dbx を使用します。加えた変更は、システムがリブートされるまで持続します。システムをリブートすると、/sbin/sysconfig -r コマンドで行った変更は失われま

す。これは、新しい値が保存されていないためです。 `/sbin/sysconfig` コマンドに `-r` オプションを指定すると、新しいサブシステム属性値のテストに役立ちます。新しい値でシステムが予想どおりに実行された場合には、4.5.8 項で説明するように、その値をサブシステム属性データベースに格納しておきます。詳細は、`dbx(1)` および『システムの構成とチューニング』を参照してください。

`/sbin/sysconfig -r` コマンドを使用する場合には、属性、新しい値、およびサブシステム名をコマンド行に指定します。たとえば、`generic` サブシステムに対して、`dump-sp-threshold` 属性を変更する場合には、次のようなコマンドを入力します。

```
# /sbin/sysconfig -r generic dump-sp-threshold=20480
```

一度に複数の属性を変更するときには、`/sbin/sysconfig` コマンド行にリストを指定します。たとえば、`dump-sp-threshold` 属性と `locktimeout` 属性を変更するには、次のようなコマンドを入力します。

```
# /sbin/sysconfig -r generic dump-sp-threshold=20480 \  
locktimeout=20
```

2 つの属性指定はコンマで区切らないことに注意してください。

属性値を永続的な値にするには、その値を適切な方法で `/etc/sysconfigtab` ファイルに追加する必要があります。たとえば、カーネル・チューナ・グラフィカル・ユーザ・インタフェースを使用して、その属性値をブート時の値として指定します。

4.5.7 サブシステムおよび属性のリモート管理

`/sbin/sysconfig -h` コマンドを使用すると、ローカル・エリア・ネットワーク (LAN) 上の構成可能サブシステムおよび動的サブシステム属性をリモートで管理することができます。これを使用すると、1 台のマシンから複数のシステムを管理できるようになります。

リモートで管理する各システムには、ローカル・システムの完全ドメイン名を含む `/etc/cfgmgr.auth` ファイルが必要です。バークレー・インターネット・ドメイン (BIND) またはネットワーク情報サービス (NIS) を使用している場合には、`/etc/cfgmgr.auth` ファイル内の名前は、`/etc/hosts` ファイル、あるいは、BIND または NIS ホスト・データベースにある名前と同じでなければなりません。省略時の設定でシステムに

/etc/cfgmgr.auth ファイルがない場合には、そのファイルを作成する必要があります。 cfgmgr.auth ファイルの例を次に示します。

```
salmon.zk3.dec.com
trout.zk3.dec.com
bluefish.zk3.dec.com
```

リモート・システムでサブシステムと属性を管理するには、
/sbin/sysconfig コマンドに -h オプションとホスト名を指定します。たとえば、環境モニタリング lat サブシステムを MYSYS という名前のリモート・ホストにロードするには、次のコマンドを入力します。

```
# /sbin/sysconfig -h MYSYS -c lat
```

この例では、指定したサブシステムをロードする前に、lat.mod ファイルが、リモート・システム上の /subsys ディレクトリまたは /var/subsys ディレクトリのいずれかに存在している必要があります。ロード可能サブシステムのサブセットが正しくキッティングされていれば、setld コマンドを使用してそのロード可能サブシステムをインストールするときに、*subsystem-name.mod* ファイルがリモート・システムにインストールされます。

4.5.8 サブシステム属性データベースの管理

動的に構成可能なサブシステム属性の情報は /etc/sysconfigtab データベースに格納されています。このデータベースは、システムがリブートされるたびに、またはサブシステムが構成されるたびに、サブシステム属性に割り当てられた値を記録します。このデータベースには、自動的に属性が登録されることはありません。/etc/sysconfigtab データベースの変更は、必ずスーパーユーザで sysconfigdb コマンド行ユーティリティまたはカーネル・チューナ・グラフィカル・ユーザ・インタフェースを使用して行う必要があります。このメソッドの使用についての詳細は、カーネル・チューナのオンライン・ヘルプを参照してください。

注意

/etc/sysconfigtab データベースには、構成可能サブシステムの stanza.loadable ファイルからスタンザ・エントリが登録されていることがあります。このファイルおよび /etc/sysconfigtab データベースのエントリは、任意の構成可

能サブシステムをインストールしたとき、自動的に作成されます。
データベース内のこれらのエントリは変更してはなりません。

/etc ディレクトリには複数のバージョンの `sysconfigtab.*` ファイルがありますが、通常の動作で使用されるのは `/etc/sysconfigtab` バージョンだけです。これらのバージョンは、`/vmunix` カーネルを作成するときのモジュールのダイナミック・リンクをサポートするために存在します。この機能はブートリンクと呼ばれ、『*Guide to Preparing Product Kits*』で説明されています。`sysconfigtab.*` ファイルのいずれかを削除すると、ブートリンクが使用できなくなります。

データベースのエントリに対して追加、更新、削除を行うには、変更する属性の名前と値の入っているスタンザ・フォーマットのファイルを作成します。スタンザ・フォーマットのファイルについては、`stanza(4)` を参照してください。たとえば、`generic` サブシステムの `lockmode` 属性を 1 に設定したい場合について考えてみます。この属性を設定するには、たとえば、`generic_attrs` という名前のファイルを作成して、次の内容を含めます。

```
generic:
    lockmode = 1
```

スタンザ・フォーマットのファイルを作成すると、`/sbin/sysconfigdb` コマンドを使用して、`/etc/sysconfigtab` データベースを更新することができます。コマンド行で `-f` オプションを使用して、スタンザ・フォーマットのファイルを指定します。`sysconfigdb` コマンドは、指定されたファイルを読み込んで、ディスク上およびメモリ内にある両方のデータベースのコピーを更新します。ただし、動作中のカーネルは更新されません。動作中のカーネルを更新する場合は、4.5.6.3 項で説明しているように、`sysconfig -r` コマンドを使用します。

`sysconfigdb` コマンドの構文は、次のとおりです。

```
/sbin/sysconfigdb {-s}
```

```
/sbin/sysconfigdb -t outfile [-f infile -a | -u subsystem-name]
```

```
/sbin/sysconfigdb -t outfile [-f infile -m | -r subsystem-name]
```

```
/sbin/sysconfigdb -t outfile [-f infile -d subsystem-name]
```

```
/sbin/sysconfigdb -t outfile [-f infile -l [ subsystem-name...]]
```

以降の各項では、`/sbin/sysconfigdb` コマンドを使用して、`/etc/sysconfigtab` データベースのエントリを管理する方法について説明します。

4.5.8.1 データベース内の属性のリスト

`/etc/sysconfigtab` データベース内のエントリをリストするには、`/sbin/sysconfigdb -l` コマンドを使用します。コマンド行にサブシステム名を指定した場合には、そのサブシステムの属性がリストされます。サブシステム名を指定しなければ、データベースに定義されているすべての属性がリストされます。

たとえば、`generic` サブシステムの属性設定をリストするには、次のコマンドを入力します。

```
# /sbin/sysconfigdb -l generic
generic:
    memberid = 0
    new_vers_high = 1441151880873377792
    new_vers_low = 15044
```

4.5.8.2 データベースへの属性の追加

`/etc/sysconfigtab` データベースにサブシステムの属性を追加するには、`sysconfigdb -a` コマンドを入力します。

たとえば、`add_attrs` という名前のファイルに格納されているエントリをデータベースに追加するには、次のコマンドを入力します。

```
# /sbin/sysconfigdb -a -f add_attrs generic
```

4.5.8.3 新しい定義の既存のデータベース・エントリへのマージ

属性の新しい定義を、`/etc/sysconfigtab` データベースの既存のエントリにマージするには、`sysconfigdb -m` コマンドを入力します。

`sysconfigdb` コマンドは、次のように、新しい定義を既存のデータベース・エントリにマージします。

- 属性名がデータベースにない場合は、その属性の定義がデータベースに追加される。
- 属性名がある場合、属性は、新しい定義によって指定された値を受け取る。

- 属性がデータベース内にあるが新しい定義にない場合、その定義はデータベース内に保持される。

たとえば、`/etc/sysconfigtab` データベースの `generic` サブシステムに次のエントリがあるとします。

```
generic:
    lockmode = 4
    dump-sp-threshold = 6000
```

このエントリを更新するために、次の情報を持つ `merge_attrs` という名前のファイルを作成します。

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
```

`merge_attrs` ファイルの情報を `/etc/sysconfigtab` データベースにマージするには、次のコマンドを入力します。

```
# /sbin/sysconfigdb -m -f merge_attrs generic
```

コマンドが終了すると、データベース内の `generic` サブシステムのエントリは、次のようになります。

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
    dump-sp-threshold = 6000
```

`sysconfigdb -m` コマンドを 1 度実行するだけで、`/etc/sysconfigtab` データベースに、複数のサブシステムの定義をマージできます。たとえば、`merge_attrs` ファイルには、サブシステム `lsm` と `generic` の属性に対する新しい定義を含めることができます。`merge_attrs` ファイルに複数のサブシステム名を含めている場合は、次のように、コマンド行でサブシステム名を省略します。

```
# /sbin/sysconfigdb -m -f merge_attrs
```

4.5.8.4 データベース内の属性の更新

`/etc/sysconfigtab` データベースに既に存在するサブシステムを更新するには、`/sbin/sysconfigdb -u` コマンドを使用します。たとえば、`/etc/sysconfigtab` ファイルで `generic` サブシステムが次のように定義されているとします。

```
generic:
    lockmode = 4
    dump-sp-threshold = 6000
```

このエントリを更新するために、次の情報を持つ `update_attrs` という名前のファイルを作成するものとします。

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
```

属性を更新するには、次のように `sysconfigdb` コマンドを入力します。

```
# /sbin/sysconfigdb -u -f update_attrs generic
```

コマンドが終了すると、データベース内の `generic` サブシステムのエントリは、次のようになります。

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
```

4.5.8.5 データベースからの属性定義の削除

`/etc/sysconfigtab` データベースから、選択した属性の定義を削除するには、`/sbin/sysconfigdb -r` コマンドを入力します。`-r` オプションは、ファイルに保管されている属性定義をデータベースから削除することを指定します。

たとえば、`generic` サブシステムが、`/etc/sysconfigtab` データベースに次のように定義されているとします。

```
generic:
    lockmode = 4
    dump-sp-threshold = 6000
```

`dump-sp-threshold` 属性の定義を削除するには、最初に、次の情報を含む `remove_attrs` という名前のファイルを作成します。

```
generic:
    dump-sp-threshold = 6000
```

その後、次のコマンドを入力します。

```
# /sbin/sysconfigdb -r -f remove_attrs generic
```

コマンドが終了すると、データベース内の `generic` サブシステムのエントリは、次のようになります。

```
generic:
    lockmode = 4
```

`/sbin/sysconfigdb` コマンドは、同一のエントリだけを削除します。つまり、同じ属性名と同じ値を持つエントリが削除されます。

`sysconfigdb -r` コマンドを 1 度実行するだけで、複数の属性の定義を削除することができます。たとえば、`remove_attrs` ファイルには、`lsm` サブシステムと `generic` サブシステムから削除する属性定義を含めることができます。`remove_attrs` ファイルに複数のサブシステムを含めている場合、次のように、コマンド行でサブシステム名を省略します。

```
# /sbin/sysconfigdb -r -f remove_attrs
```

4.5.8.6 データベースからのサブシステム・エントリの削除

`/etc/sysconfigtab` データベースからサブシステムの定義を削除するには、`/sbin/sysconfigdb -d` コマンドを入力します。

たとえば、`generic` サブシステム・エントリをデータベースから削除するには、次のコマンドを入力します。

```
# /sbin/sysconfigdb -d generic
```

次にそのサブシステムを構成するとき、`generic` サブシステムに省略時の値が渡されます。

4.6 静的システム構成

静的システム構成とは、新しいカーネルを作成、ブートするためのコマンドおよびファイルと、静的サブシステムを表します。サブシステムは作成時にカーネルに直接リンクされているため、カーネルからは静的に認識されます。静的にリンクされたカーネルを構築する手順は、カーネルを変更する理由によって異なります。

デバイス・ドライバを追加するためにカーネルを変更するときは、次の一般的な手順で行います。

- デバイス・ドライバをインストールする。
- 必要であれば、ターゲット構成ファイルを編集する。

場合によっては、サブセット・コントロール・プログラム (SCP) がデバイス・ドライバによって提供されています。その場合は、SCPがインストール・プロシージャで実行され、必要なシステム構成ファイルにドライバを登録します。この場合、ターゲット構成ファイルを編集する必要はありません。

- 新しいカーネルを構築する。

デバイス・ドライバに SCP が含まれている場合は、4.6.3 項で説明しているように、`/usr/sbin/doconfig` プログラムを実行して新しいカーネルを構築します。新しいカーネルを構築する前にターゲット構成ファイルを編集する必要がある場合は、4.6.1 項を参照してください。

- システムをシャットダウンして、リブートする。

カーネルを変更して、特定のカーネル・オプションをサポートするには、`/usr/sbin/doconfig` プログラムを実行し、処理中に表示されるメニューからカーネル・オプションを選択することによって、新しいカーネルを構築できます。その後、システムをシャットダウンして、リブートします。

この方法で構築するカーネル・オプションを決定するには、`/usr/sbin/kopt` コマンドを入力します。このコマンドは、カーネル・オプションのリストを表示して、カーネル・オプションを選択するように求めます。オプションを選択せずに `/usr/sbin/kopt` コマンドを終了するときは、Return キーを押します。`/usr/sbin/doconfig` プログラムを実行して、メニューからカーネル・オプションを追加する方法についての詳細は、4.6.2 項を参照してください。

別の理由で新しい静的カーネルを再構成する場合には、カーネルの再構築時に 1 つまたは複数のシステム・ファイルを変更する必要があります。変更するシステム・ファイルは、カーネルに対して行う変更により異なります。

- たとえば、構築するカーネルの定義や、デバイスの定義、あるいは擬似デバイスの定義を行うキーワードなどを変更するには、ターゲット構成ファイルを変更します。また、このファイルを編集してシステム・パラメータの値を変更することもできます。ターゲット構成ファイルの内容についての詳細は、4.7 節を参照してください。
- 特定の静的サブシステムをカーネルから削除するには、`/usr/sys/conf` ディレクトリ内のファイルからそのエントリを削除 (またはコメント・アウト) します。このファイルについての詳細は、4.7.2 項を参照してください。

システム・ファイルを編集した後、`/usr/sbin/doconfig` プログラムを実行してカーネルを構築する方法については、4.6.3 項を参照してください。

デバイスを追加して構成する例については、『ハードウェア管理ガイド』を参照してください。

4.6.1 新しいデバイスをサポートするためのカーネルの構築

新しいデバイスをシステムに追加する場合、そのデバイス・インストレーションに SCP が含まれていなければ、ターゲット構成ファイルを編集して、オペレーティング・システムで新しいデバイスがサポートできるように定義する必要があります。ターゲット構成ファイルにはデバイス定義キーワードをいれます。オペレーティング・システムでは多くのデバイスをサポートしているため、ターゲット構成ファイルに追加するキーワードの決定が困難なこともあります。

次の手順は、ターゲット構成ファイルに追加するデバイス定義キーワードを決定する方法を示しています。また、ターゲット構成ファイルの編集が終わった後、カーネルを再構築する方法も同時に示しています。この手順では、追加する適当なキーワードがわかっていないことを前提としています。場合によっては、ハードウェアやオペレーティング・システムの新しいバージョンに付属するマニュアルを参照することによって、適当なキーワードを判断できることがあります。また、同じデバイスが接続されている別のシステム上に構成ファイルが存在している場合は、それによって判断できることもあります。システムに追加するキーワードがわかっている場合は、4.6.3 項で説明している方法で、ユーティリティを使用し、そのキーワードをターゲット構成ファイルに追加して、カーネルを再構築します。

注意

この手順は危険なため、カスタマイズした `/vmunix` カーネル・ファイル、汎用カーネル `/genvmunix`、および現在の構成ファイルのそれぞれのコピーを取っていることを確認してください。必要な場合には、これらのコピーを使用して、以前の構成に戻します。

システムのターゲット構成ファイルに追加するキーワードがわかっていない場合は、ハードウェアのマニュアルに従って、新しいデバイスをシステムに接続し、次の手順を実行します。

1. この手順の後半で汎用カーネルをブートする必要があるので、
`/genvmunix` ファイルが存在することを確認してください。
`/genvmunix` ファイルがシステム上に存在していない場合、または追加するデバイスを汎用カーネルが認識できない場合は、汎用カーネル

をソフトウェア CD-ROM からコピーします。CD-ROM 上の汎用カーネルの格納場所については、『インストール・ガイド』を参照してください。

まれに、汎用カーネルの再構築が必要になることがあります。ただし、レイヤード・ソフトウェアまたはサードパーティ製のデバイス・ドライバがインストールされている場合には、汎用カーネルを再構築することはできません。このような場合で、元の `/genvmunix` が壊れてしまったかまたは削除されてしまったにもかかわらず、ソフトウェア CD-ROM がない場合には、技術サポート部門に連絡して汎用カーネル `/genvmunix` のコピーを入手してください。

レイヤード・アプリケーションがインストールされているかどうかを調べるには、`/usr/sys/conf` ディレクトリの `.product.list` ファイルの内容を確認します。

汎用カーネルを再構築するときは、必須およびオプションのカーネル構築サブセットをすべてインストールしておく必要があります。次のコマンドを実行すると、カーネル・サブセットのリストが表示されて、それぞれのカーネル・サブセットがインストールされているかどうかわかります。

```
# /usr/sbin/setld -i | grep Kernel Build
```

すべてのカーネル・サブセットをインストールしたら、次のコマンドを入力します。

```
# doconfig -c GENERIC
```

`-c` オプションは、既存の構成ファイルを使用してカーネルを構築することを指定します。この場合は、`GENERIC` 構成ファイルになります。既存の構成ファイルからカーネルを構築する方法についての詳細は、4.6.3 項を参照してください。

汎用カーネルが動作して新しいデバイスが認識されたら、ステップ 5 に進みます。構築が終了したら、`strip` コマンドを使用して、カーネルのサイズを縮小することを検討してください。詳細は、`strip(1)` を参照してください。

2. `root` でログインするか、またはスーパーユーザになって、省略時のディレクトリを `/usr/sys/conf` ディレクトリにします。
3. 既存の `/vmunix` ファイルのコピーを保存します。可能であれば、このファイルを、次のようにしてルート (`/`) ディレクトリに保存します。

```
# cp /vmunix /vmunix.save
```

ディスク・スペースに制約がある場合は、カーネル・ファイルをルート以外のファイル・システムに保存しても構いません。たとえば、次のようにします。

```
# cp /vmunix /usr/vmunix.save
```

4. システムをシャットダウンして停止させます。次のように入力します。

```
# shutdown -h now
```

5. コンソール・プロンプトで、汎用カーネル `/genvmunix` をブートします。汎用カーネルでは、すべての有効なデバイスがサポートされているため、新しいデバイスをターゲット・カーネルに追加するときにブートすると、新しいデバイスが汎用カーネルによって認識されます。汎用カーネルをブートするには、次のコマンドを入力します。

```
>>> boot -fi "genvmunix"
```

6. シングルユーザ・モードのプロンプトで、次のコマンドを実行して、ローカル・ファイル・システムの検査とマウントを行います。ただし、Logical Storage Manager (LSM) を使用している場合は、その限りではありません。

```
# /sbin/bcheckrc
```

Logical Storage Manager (LSM) ソフトウェアを使用している場合は、次のコマンドを実行して、ローカル・ファイル・システムを検査して LSM を起動します。

```
# /sbin/lsmbootstrap
```

7. `sizer` プログラムを実行して、システム・ハードウェアの大きさを判定し、新しいデバイスを含む新しいターゲット構成ファイルを作成します。

```
# sizer -n MYSYS
```

`sizer -n` コマンドは、ユーザのシステムに対する新しいターゲット構成ファイルを作成します。この構成ファイルには、新しいデバイスに対応する適切なデバイス定義キーワードが含まれています。このプロセスは、システムのインストール時のプロセスと似ています。詳細については、4.3 節を参照してください。`sizer` プログラムは、新しいターゲット構成ファイルを `/tmp` ディレクトリに格納します。

8. `sizer` で作成した新しいターゲット構成ファイルを、システム上にある既存のターゲット構成ファイルと比較します。

```
# diff /tmp/MYSYS MYSYS
```

これらの2つのファイルの違いを検査して、新しいデバイス定義キーワードを探します。maxusers オプションで省略時以外の値を指定して既存の構成ファイルをカスタマイズしている場合などは、他の箇所にも違いが出てくる場合があります。

9. 好みのテキスト・エディタを使用して、新しいデバイス定義キーワードを既存の構成ファイル(この場合 MYSYS)に追加します。新しいキーワードを追加すると、既存の構成ファイルで新しいデバイスがサポートできるようになります。なお、そのファイルに対して今まで行った変更が失われることはありません。

注意

構成ファイルに対して通信デバイスを追加または削除している場合には、/etc/inittab ファイルと /etc/securettys ファイルを編集して、新しい構成、つまり、/dev/ttynn デバイス特殊ファイルに一致させる必要があります。詳細については、inittab(4) および securettys(4) を参照してください。

10. 次の /usr/sbin/doconfig コマンドを実行して、新しいカーネルを構築します。

```
# /usr/sbin/doconfig -c MYSYS
```

```
*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***
```

```
Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
```

次のプロンプトが表示されたら、構成ファイルの編集は必要ないと答えます。

```
Do you want to edit the configuration file? (y/n) [n]:n
```

```
*** PERFORMING KERNEL BUILD ***
```

```
.  
.  
.
```

```
The new kernel is /usr/sys/MYSYS/vmunix
```

11. カーネルの構成と構築プロセスがエラーなしで終了したら、新しい `vmunix` ファイルを `/vmunix` にコピーします。 `MYSYS` という名前のシステム上では、次のようなコマンドを入力します。

```
# cp /usr/sys/MYSYS/vmunix /vmunix
```

ファイルを移動する (`mv`) のではなく、必ずファイルをコピー (`cp`) して、コンテキスト依存シンボリック・リンク (CDSL) を維持します。CDSL についての詳細は、第 6 章を参照してください。

12. 次のようにして、システムをリブートします。

```
# /usr/sbin/shutdown -r now
```

新しい `/vmunix` ファイルのブートに失敗した場合には、この手順の最初に保存したカーネルを使用してブートします。保存したカーネルを使用するときは、次の手順で行います。

1. 次のように `fsck -p` コマンドを使用して、すべてのローカル・ファイル・システムを検査します。

```
# fsck -p
```

2. `mount -u` コマンドを使用して、`root` ファイル・システムを書き込み可能にします。次のように入力してください。

```
# mount -u /
```

3. 必要であれば、`/vmunix.save` ファイルが保存されているファイル・システムをマウントします。たとえば、`/vmunix` ファイルを `/usr` ファイル・システムにコピーしている場合は、次のコマンドを入力します。

```
# mount /usr
```

4. 保存したファイルのコピーをリストアします。たとえば、動作しているカーネルを `/vmunix.save` ファイルに保存している場合、次のコマンドを入力します。

```
# mv /vmunix.save /vmunix
```

5. システムをシャットダウンしてリブートします。次のように入力します。

```
# shutdown -r now
```

システムが再度稼働を始めたら、ターゲット構成ファイルを必要に応じて変更し、ステップ 3 からの手順に従って、カーネルを再構築することができます。

4.6.2 選択したカーネル・オプションを追加するためのカーネルの構築

オプションを指定しないで `/usr/sbin/doconfig` プログラムを起動すると、メニューを使用してカーネルを変更することができます。メニューを使用してカーネルを変更するには、次の手順に従ってください。

1. `root` でログインするか、またはスーパーユーザになって、省略時のディレクトリを `/usr/sys/conf` ディレクトリにします。
2. 既存の `/vmunix` ファイルのコピーを保存します。可能であれば、次のようにしてファイルをルート (`/`) ディレクトリに保存します。

```
# cp /vmunix /vmunix.save
```

ディスク・スペースに制約がある場合は、カーネル・ファイルをルート以外のファイル・システムに保存しても構いません。たとえば、次のようにします。

```
# cp /vmunix /usr/vmunix.save
```

3. 次のように、オプションを指定しないで `/usr/sbin/doconfig` プログラムを実行します。

```
# /usr/sbin/doconfig
```

```
*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***
```

```
Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
```

4. 次のプロンプトに対して、構成ファイルの名前を入力します。

```
Enter a name for the kernel configuration \
file. [MYSYS]: MYSYS
```

カーネル構成プロセスは、省略時の構成ファイル名を決定するときに、`MYSYS` のように、システム名を大文字に変換します。たとえば、`mysys` という名前のシステムでは、省略時の構成ファイルは `MYSYS` という名前になります。

指定した構成ファイル名が現在存在していなければ、`/usr/sbin/doconfig` プログラムによりそのファイル名でシステム構成ファイルが作成されます。ステップ 10 のカーネル・オプションを選択して、この処理を継続してください。

5. 指定した構成ファイル名が存在する場合、次のプロンプトが表示されるので、ファイルを重ね書きするよう応答します。

A configuration file with the name MYSYS already exists.
Do you want to replace it? (y/n) [n]: **y**

6. 次のようなメニューが表示されるので、その中からカーネル・オプションを選択します。

*** KERNEL OPTION SELECTION ***

```
-----
Selection  Kernel Option
-----
1      System V Devices
2      NTP V3 Kernel Phase Lock Loop (NTP_TIME)
3      Kernel Breakpoint Debugger (KDEBUG)
4      Packetfilter driver (PACKETFILTER)
5      Point-to-Point Protocol (PPP)
6      STREAMS pckt module (PCKT)
7      Data Link Bridge (DLPI V2.0 Service Class 1)
8      X/Open® Transport Interface (XTISO, TIMOD, TIRDWR)
9      ISO 9660 Compact Disc File System (CDFS)
10     Audit Subsystem
11     Alpha CPU performance/profiler (/dev/pfcntr)
12     ACL Subsystem
13     Logical Storage Manager (LSM)
14     ATM UNI 3.0/3.1 ILMI (ATMILMI3X)
15     IP Switching over ATM (ATMIFMP)
16     LAN Emulation over ATM (LANE)
17     Classical IP over ATM (ATMIP)
18     ATM UNI 3.0/3.1 Signalling for SVCs (UNI3X)
19     Asynchronous Transfer Mode (ATM)
20     All of the above
21     None of the above
22     Help
23     Display all options again
-----
```

Enter the selection number for each kernel option you want.
For example, 1 3 [18]:

7. 次のプロンプトが表示されたら、構成ファイルの編集が必要かどうか応答します。

Do you want to edit the configuration file? (y/n) [n]:

メニューのサブシステムをカーネルに追加する以外の変更がなければ、構成ファイルを編集する必要はありません。

構成ファイルの編集を選択した場合、`/usr/sbin/doconfig` プログラムが、`EDITOR` 環境変数で指定されているエディタを起動します。

構成ファイルの詳細については、4.7 節を参照してください。

構成ファイルの編集が終了すると、`/usr/sbin/doconfig` プログラムは新しいカーネルを構築します。

8. カーネルの構成と構築プロセスがエラーなしで終了したら、新しい `vmunix` ファイルを `/vmunix` に移動します。MYSYS という名前のシステム上では、次のコマンドを入力します。

```
# mv /usr/sys/MYSYS/vmunix /vmunix
```

9. 次のようにして、システムをリブートします。

```
# /usr/sbin/shutdown -r now
```

新しい `/vmunix` ファイルのブートに失敗した場合には、この手順の最初に保存したカーネルを使用してブートします。保存したカーネルを使用するときは、次の手順に従ってください。

1. 次のように `fsck -p` コマンドを使用して、すべてのローカル・ファイル・システムを検査します。

```
# fsck -p
```

2. `mount -u` コマンドを使用して、`root` ファイル・システムを書き込み可能にします。次のように入力してください。

```
# mount -u /
```

3. 必要であれば、`/vmunix.save` ファイルが保存されているファイル・システムをマウントします。たとえば、`/vmunix` ファイルを `/usr` ファイル・システムにコピーしている場合は、次のコマンドを入力します。

```
# mount /usr
```

4. 保存したコピーをリストアします。たとえば、動作しているカーネルを `/vmunix.save` ファイルに保存している場合は、次のコマンドを入力します。

```
# mv /vmunix.save /vmunix
```

5. システムをシャットダウンしてリブートします。次のように入力します。

```
# shutdown -r now
```

システムが再度稼働を始めたら、ターゲット構成ファイルを必要に応じて変更し、ステップ 3 からの手順に従って、カーネルを再構築することができます。

4.6.3 システム・ファイルを変更した後のカーネルの構築

ユーザまたは SCP が、ターゲット構成ファイルのようなシステム・ファイルを変更した場合は、`/usr/sbin/doconfig -c` コマンドを使用して、カーネルを再構築できます。`-c` オプションを使用して既存の構成ファイルを指定すると、`/usr/sbin/doconfig` プログラムがそのファイルを使用してカーネルを構築します。既存の構成ファイルを使用して新しいカーネルを構築するときは、次の手順に従ってください。

1. `root` でログインするか、またはスーパーユーザになって、省略時のディレクトリを `/usr/sys/conf` ディレクトリにします。
2. 既存の `/vmunix` ファイルのコピーを保存します。可能であれば、次のようにしてファイルをルート (`/`) ディレクトリに保存します。

```
# cp /vmunix /vmunix.save
```

ディスク・スペースに制約がある場合は、カーネル・ファイルをルート以外のファイル・システムに保存しても構いません。たとえば次のようにします。

```
# cp /vmunix /usr/vmunix.save
```

3. `-c` オプションでターゲット構成ファイル名を指定して、`/usr/sbin/doconfig` プログラムを実行します。たとえば `MYSYS` という名前のシステム上では、次のコマンドを入力します。

```
# /usr/sbin/doconfig -c MYSYS
```

```
*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***
```

```
Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
```

4. 次のプロンプトが表示されたら、構成ファイルの編集が必要かどうかを応答します。

```
Do you want to edit the configuration file? (y/n) [n]:
```

この手順を始める前に構成ファイルを変更している場合は、構成ファイルの編集が必要ないことを応答します。

構成ファイルの編集を選択した場合は、`/usr/sbin/doconfig` プログラムによって、`EDITOR` 環境変数で指定されているエディタが起動されます。

構成ファイルの詳細については、4.7 節を参照してください。

構成ファイルの編集が終了すると、`/usr/sbin/doconfig` プログラムは新しいカーネルを構築します。

5. カーネルの構成と構築がエラーなしで終了すると、新しい `vmunix` ファイルを `/vmunix` に移動します。MYSYS という名前のシステム上では、次のコマンドを入力します。

```
# mv /usr/sys/MYSYS/vmunix /vmunix
```

6. 次のようにして、システムをリブートします。

```
# /usr/sbin/shutdown -r now
```

新しい `/vmunix` ファイルのブートに失敗した場合には、この手順の最初に保存したカーネルを使用してブートします。保存したカーネルを使用するときは、次の手順に従ってください。

1. 次のように `fsck -p` コマンドを使用して、すべてのローカル・ファイル・システムを検査します。

```
# fsck -p
```

2. `mount -u` コマンドを使用して、`root` ファイル・システムを書き込み可能にします。次のように入力してください。

```
# mount -u /
```

3. 必要であれば、`/vmunix.save` ファイルが保存されているファイル・システムをマウントします。たとえば、`/vmunix` ファイルを `/usr` ファイル・システムにコピーしている場合、次のコマンドを入力します。

```
# mount /usr
```

4. 保存したコピーをリストアします。たとえば、動作しているカーネルを `/vmunix.save` ファイルに保存している場合、次のコマンドを入力します。

```
# mv /vmunix.save /vmunix
```

5. システムをシャットダウンしてリブートします。次のように入力します。

```
# shutdown -r now
```

システムが再度稼働を始めたら、ターゲット構成ファイルを必要に応じて変更し、ステップ 3 からの手順に従って、カーネルを再構築することができます。

4.7 構成ファイル

作業カーネルを作成し、実行するためには、`/usr/sys` ディレクトリの下に特定のディレクトリが存在している必要があります。図 4-1 に、システム構成ファイルのディレクトリ構造を示します。点線は、他社の静的サブシステムのためのオプションのディレクトリおよびファイルを示しています。

図 4-1: 構成ファイルのディレクトリの階層構造

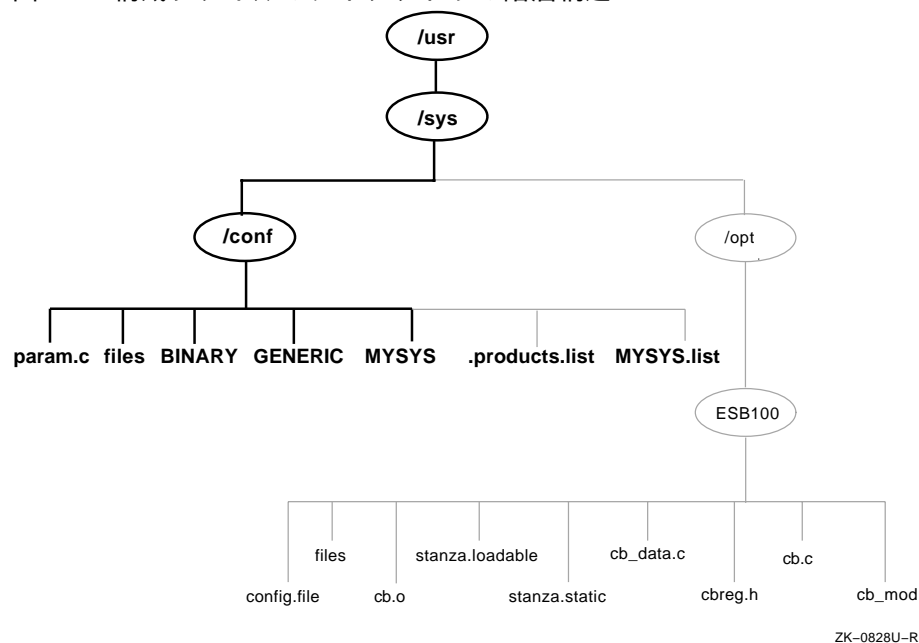


図 4-1 に示すように、`/usr/sys/conf` ディレクトリには、すべてを包含する汎用カーネルとターゲット・マシンのカーネルの両方のカーネル構成を定義するためのファイルが含まれています。これらのファイルは、カーネル構成の静的な部分について表現しています。システム・ファイル进行处理してカーネルを再構成する場合には、主に次の 5 つのファイルを使用します。

- `/usr/sys/conf/MYSYS`。ここで、`MYSYS` はシステム名です。
- `/usr/sys/conf/GENERIC`
- `/usr/sys/conf/.product.list`
- `/usr/sys/conf/NAME.list`
- `/usr/sys/conf/param.c`

以降の各項で、これらのファイルについて詳しく説明します。

4.7.1 `/usr/sys/conf` の構成ファイル

`/usr/sys/conf` ディレクトリには、2つの重要なシステム構成ファイルである、ターゲット構成ファイルとGENERIC 構成ファイルが含まれています。

4.7.1.1 ターゲット構成ファイル

ターゲット構成ファイル `/usr/sys/conf/NAME` は、システムがカーネルに組み込む構成要素を定義するテキスト・ファイルです。慣例により、パス名の `NAME` の部分には、システム名を大文字で指定します。たとえば、システム名が `MYSYS` の場合には、`/usr/sys/conf/MYSYS` というファイル名になります。ターゲット構成ファイルは、システムのインストール中に、`sizer` プログラムにより各システムに対して作成されます。次のようなキーワード定義の変更を行いたい場合には、ターゲット構成ファイルを変更します。

- 複数のシステムを管理する場合、それらのシステムで同じように定義されるグローバル・キーワード
- 特定のシステムで構築するカーネルを記述するシステム定義キーワード
- 特定のシステムに接続するデバイスを記述するデバイス定義キーワード
- シェル・コマンド・サブプロセスをカーネル構成中に実行できるようにする `callout` キーワード定義
- ソフトウェアをコンパイルしてシステム内に組み込むことを指定する `options` キーワード定義
- カーネルの構築時に、コンパイラ、アセンブラ、およびリンカに引き渡される `makeoptions` キーワード定義
- システム上の擬似デバイスを記述する `pseudodevice` キーワード定義

4.7.1.2 GENERIC 構成ファイル

`/usr/sys/conf/GENERIC` 構成ファイルは、汎用カーネルを記述する構成ファイルです。汎用カーネルは、すべての有効なデバイスをサポートしているため、新しいデバイスをシステムに追加するときを使用すると便利です。汎用カーネルは、ターゲット・カーネルが何らかの理由で破損した場合に備えて、バックアップ・カーネルとして使用することもできます。

/genvmunix ファイルには汎用カーネルが含まれているため、このファイルは削除しないでください。誤って汎用カーネルを削除した場合は、`doconfig -c GENERIC` コマンドを使用して、再構築することができます。既存の構成ファイルを使用してカーネルを構築する方法については、4.6.3 項を参照してください。

注意

/usr/sys/conf/GENERIC ファイルは削除しないでください。

4.7.2 ターゲット構成ファイルの機能拡張

/usr/sys/conf ディレクトリには、ターゲット構成ファイルの拡張を記述する 2 つのオプションの構成ファイルが含まれています。これらのファイルは、/usr/sys/conf/.product.list ファイルと /usr/sys/conf/NAME ファイルです。これらのファイルには静的カーネル・サブシステムに関する情報が格納されるため、カーネル・レイヤード・プロダクトと呼ばれることがあります。

静的サブシステムをインストールする場合、通常、その SCP は /usr/sys/conf/.product.list ファイルを編集して、そのサブシステムのエントリを追加します。SCP が終了した後、/usr/sbin/doconfig プログラムを実行して、新しいサブシステムをカーネル内に構成してください。

/usr/sbin/doconfig プログラムは NAME.list ファイルを作成します。NAME 変数は、ターゲット構成ファイルと同じで、慣例により、大文字のシステム名になります。たとえば、MYSYS という名前のシステムの NAME.list ファイルは、MYSYS.list になります。

他社のレイヤード・プロダクトのために（たとえば、構築中のカーネルからレイヤード・プロダクトを削除するために）システムを変更する必要がある場合には、NAME.list ファイルに対して必要な変更を行ってから新しいカーネルを構築します。

NAME.list ファイルの各エントリは、コロン (:) で区切られた 6 つのフィールドから構成されています。次の例は NAME.list ファイルの一部であり、/usr/sys/opt/ESB100 ディレクトリにロード済みの静的カーネル・サブシステムのエントリを示しています。

```
/usr/sys/opt/ESB100:UNXDASH100:920310100739:DASH Systems:controlsys:100
  1         2         3         4         5         6
```

このエントリのフィールドには、次のような情報が含まれています。

- ❶ 最初のフィールドには、入力データに対する拡張データをシステム構成ツールが探すディレクトリの絶対パス名を指定します。このロケーションには次のようなファイルを含むことができます。
 - 製品固有の構成ファイル
 - config.file ファイル・フラグメント (その製品に関するキーワードを含む)
 - files ファイル・フラグメント (その製品のソース・コードの位置、製品がカーネルにロードされる時期、ソース・コードとバイナリ・コードのどちらが提供されるかについての情報を含む)
 - stanza.static ファイル (静的ドライバの主デバイス番号要件および名前、デバイス特殊ファイルの副デバイス番号についての情報を含む)
 - オブジェクト・ファイル
 - ソース・コード・ファイル
- ❷ 2 番目のフィールドには、setld サブセット識別子を指定します。
- ❸ 3 番目のフィールドには、製品が配布された日付および時間を指定します。
- ❹ 4 番目のフィールドには、そのサブシステムを提供した会社名を指定します。
- ❺ 5 番目のフィールドには、製品名を指定します。
- ❻ 6 番目のフィールドには、setld の 3 桁の製品バージョン・コードを指定します。

NAME.list ファイルの行エントリの順番は、エントリが処理される順番に影響するため注意が必要です。

/usr/sbin/doconfig プログラムは、.product.list ファイルが存在する場合、そのファイルをコピーして NAME.list ファイルを作成します。 /usr/sbin/doconfig -c コマンドを使用する場合には、/usr/sbin/doconfig は既存の NAME.list ファイルを使用します。

新しいカーネル・レイヤード・プロダクトがインストールされた場合など、`.product.list` ファイルが変更されている状態で `-c` オプションを指定する場合には、`NAME.list` ファイルを削除するか、`/usr/sbin/doconfig` を起動する前にこのファイルを手作業で編集し、`.product.list` ファイルの変更点を `NAME.list` ファイルに反映する必要があります。

`.product.list` ファイルをコピーすることによって `NAME.list` ファイルを作成することもできます。カーネルに組み込みたくない行は、`NAME.list` ファイルを編集して削除するか、またはその行の先頭に番号記号 (#) を記述してコメント・アウトすることができます。

注意

`.product.list` ファイルを編集してはなりません。

`NAME.list` および `.product.list` ファイルについての詳細は、デバイス・ドライバのマニュアルを参照してください。

4.7.3 param.c ファイル

`param.c` ファイルには、いくつかのシステム・パラメータの省略時の値が含まれています。これらのパラメータは、マニュアルまたは技術サポート部門で指示されない限り、変更しないでください。

属性値を読み取って使用する優先順位は、次のとおりです。

1. 属性の実行時値
2. `/etc/sysconfigtab` ファイルに記述されている値
3. `/usr/sys/conf/SYSTEM_NAME` ファイルに記述されている値
4. `/usr/sys/conf/param.c` ファイルに記述されている値

`param.c` ファイルのパラメータがターゲット構成ファイルにも存在している場合があります。この場合には、構成ファイルに指定されている値が `param.c` ファイルに指定されている値より優先されます。したがって、`param.c` ファイルのシステム・パラメータ値を変更する場合には、ターゲット構成ファイルの対応するエントリを確実に削除するようにしてください。

4.7.4 システム構成ファイル・エントリ

システム構成ファイルには、次のキーワード定義が含まれています。

- グローバル・キーワード定義
- システム定義キーワード
- デバイス定義キーワード
- `callout` キーワード定義
- `options` キーワード定義
- `makeoptions` キーワード定義
- `pseudo-device` キーワード定義

このファイル内の構成オプションを手動でチューニングしたり、カスタマイズしたりしないでください。カーネルの構成とチューニングについての詳細は、『システムの構成とチューニング』を参照してください。

ディスクの管理

この章では、ディスクの管理に関連する、システム管理作業について説明します。説明する項目は、次のとおりです。

- グラフィカル・ユーザ・インタフェースを用いたディスク・パーティションの作成 (5.1 節)
- 手動でのディスク・パーティションの作成 (5.2 節)
- ディスクのコピー (5.3 節)

AdvFS の管理については、『*AdvFS 管理ガイド*』を参照してください。

5.1 Disk Configuration ユーティリティを用いたディスク・パーティションの作成

Disk Configuration グラフィカル・ユーザ・インタフェース (GUI) の `diskconfig` を使用して、次の作業を行うことができます。

- 存在しているディスクの属性情報表示
- ディスク構成属性の変更
- ディスク・パーティションの管理
- ディスク・パーティションでの AdvFS および UFS ファイル・システムの作成
- ディスクの別名管理

Disk Configuration GUI (`diskconfig`) の起動についての詳細は、`diskconfig(8)` を参照してください。GUI の使い方は、オンライン・ヘルプで見ることができます。

Disk Configuration GUI を、以下のいずれかの方法で起動します。

- システム・プロンプトで、`/usr/sbin/diskconfig` と入力します。

- 以下の手順を実行します。
 1. CDE フロント・パネルの SysMan アプリケーション・ポップアップ・メニューを選択します。
 2. [Configuration] を選択します。「SysMan Configuration」フォルダが開きます。
 3. 「SysMan Configuration」フォルダから「ディスク」アイコンを選択します。

注意

パーティションのサイズを変更しようとする、Disk Configuration から警告が表示されます。しかし、必要なデータを上書きしないためには、変更があることを前もって計画にしておかなければなりません。パーティションのサイズを変更する前に、データ・パーティションをすべてバックアップしておいてください。

「Disk Configuration on *hostname*」というタイトルのウィンドウが表示されます。これが Disk Configuration アプリケーションのメイン・ウィンドウであり、システム内の各ディスクのリストと、各ディスクについて次のような情報が表示されます。

- ディスクのベース名。たとえば `dsk10`。ディスク名についての詳細は、『ハードウェア管理ガイド』を参照してください。
- デバイスの機種。たとえば `RZ1CB-CA`。
- Bus, Target, および LUN (論理ユニット番号) によって表わされる物理位置。デバイスの位置についての詳細は、『ハードウェア管理ガイド』を参照してください。

特定のディスクを構成するために、リスト内のデバイスを選択し、[設定...] を選択します。別の方法として、リスト内のディスクをダブルクリックすることもできます。2つのウィンドウ、「Configure Partitions」と「Partition Table」がオープンします。

Disk Configuration: Configure Partitions: *device name device type*

このウィンドウで、ディスク・パーティションの開始アドレス、終端アドレス、サイズ、および使用オプションを設定できます。

Disk Configuration: Partition Table: *device name device type*

このウィンドウには、現在のパーティションの開始アドレス、終端アドレス、およびサイズが表示されます。

これらのウィンドウについての詳細は、オンライン・ヘルプを参照してください。

パーティションを調整した後、次のように SysMan Menu のオプションを使用して、新しく作成したファイル・システムをマウントします。

1. 新しいファイル・システムを `/etc/fstab` ファイルに追加する。
2. SysMan Menu を起動する。詳細は、第 1 章を参照してください。
3. [ファイル・システム] を選択する。さらにオプションが表示されます。
4. [一般的なファイルシステム・ユーティリティ] を選択する。さらにオプションが表示されます。
5. [ファイルシステムのマウント] を選択する。「マウント操作」のメイン・ウィンドウがオープンします。
6. [AdvFS/UFS ファイル・システムのマウント] を選択し、[次へ] を選択する。「Steps」ダイアログ・ボックスがオープンします。
7. [次へ] を選択する。「Select File System」ダイアログ・ボックスがオープンします。
8. ファイル・システムのリストからファイルを選択し、[次へ] を選択する。「Mount」ダイアログ・ボックスがオープンします。
9. 「Mount Directory」フィールドに `/usr/newusers` のようなマウント・ポイントを入力する。
あるいは、[Browse...] を使用し、マウント・ポイントのディレクトリを見つけて選択する。
10. アクセス・モードを選択する。
11. [次へ] を選択する。「Summary」ダイアログ・ボックスがオープンします。
12. [Finish] を選択する。
13. [Exit] を選択して、SysMan Menu を終了する。

これで新しいファイル・システムがアクセスできるようになります。

5.1.1 パーティション・ウィンドウの構成

このウィンドウには、以下の情報とオプションが表示されます。

- ディスク・パーティションを表わす横棒グラフ。別の色で現在強調表示されているパーティションについては、その詳細が「該当パーティション」ボックスに表示されます。この棒グラフのハンドル(またはフラグ)を用いてパーティションのサイズを変更できます。次の位置にカーソルを置いてください。
 - 隣接する2つのパーティションを変更する場合は、中央にあるハンドル。
 - 右側にあるパーティションの開始位置を上に移動する場合は、上のフラグ。
 - 左側にあるパーティションの終了位置を下に移動する場合は、下のフラグ。

MB1 を押してマウスをドラッグし、ハンドルを移動します。

- プルダウン・メニュー。サイズの単位をこのメニューで切り替えることができます。選択できる単位は、M バイト、バイト、およびブロックです。
- 統計ボックス。デバイス名、および、ディスクのトータル・サイズとその使用量に関する情報が表示されます。このボックスを使用して、ディスク・ラベルの割り当てや編集、また、デバイスの別名の作成を行います。
- 「該当パーティション」ボックス。選択したパーティションのサイズが動的に表示されます。このサイズは、棒グラフを用いたパーティションの変更に合わせて更新されます。このウィンドウにパーティション・サイズを直接入力して、現在の設定を変更することもできます。また、パーティションのファイル・システムを選択したり、AdvFS を使用している場合はドメイン名やファイルセット名を選択したりすることもできます。
- [ディスク属性...] オプション
このボタンを押すと、デバイスの物理属性の一部が表示されます。
- [パーティション・テーブル...] オプション。次の項目で説明します。

5.1.2 パーティション・テーブル・ウィンドウ

このウィンドウには現在使用しているパーティションの棒グラフ，サイズ，および使われているファイル・システムが表示されます。現在のパーティション・サイズを，このデバイスの省略時のテーブルにある値とセッションが開始された時点のオリジナル(開始テーブル)の値の間で切り替えることができます。手動によるパーティション変更に失敗した場合は，このウィンドウを用いてパーティション・テーブルをリセットできます。

5.2 手動によるディスク・パーティションの作成

この項では，ディスクのパーティション・スキームの変更に必要な情報を示します。一般に，ディスク・スペースは，初期インストール時またはディスクをシステム構成に追加する場合に割り当てます。通常は，パーティションを変更する必要はありません。しかし，環境の変化に対応したりシステムの性能を向上させたりするために，ディスクのパーティション・サイズの変更が必要な場合もあります。

5.2.1 ユーティリティ

以下のコマンドを使用すると，手動によるディスク保守ができます。

| | |
|------------------------|--|
| <code>disklabel</code> | このコマンドを使用して，ディスク・ドライブまたはディスク・バックのラベルの，インストール，検査，および変更を行います。ディスク・ラベルには，タイプ，物理パラメータ，およびパーティションなどの，ディスクについての情報が記録されています。/etc/disktab ファイルについての詳細は， <code>disklabel(4)</code> を参照してください。 |
|------------------------|--|

| | |
|--------------------|---|
| <code>newfs</code> | このコマンドを使用して，指定したデバイスに新しい UFS ファイル・システムを作成します。AdvFS (Advanced File System) ドメインを作成する場合は， <code>newfs</code> コマンドではなく， <code>mkfdmn</code> コマンドを使用してください。 |
|--------------------|---|

| | |
|---|---|
| <code>mkfdmn</code> と <code>mkfset</code> | このコマンドは，AdvFS (Advanced File System) ドメインとファイルセットの作成に使用します。詳細は， <code>mkfdmn(8)</code> を参照してください。 |
|---|---|

5.2.2 disklabel ユーティリティの使用

ディスク・ラベルには、ディスクおよびディスクを分割するパーティションのジオメトリに関する詳細な情報が入っています。root ユーザは、disklabel コマンドでラベルを変更することができます。コマンド・オプションの詳細は、disklabel(8) を参照してください。

ディスク・ラベルには2つのコピーがあり、1つはディスクに、もう1つはシステム・メモリに置かれます。I/O を実行するよりはシステム・メモリにアクセスする方が速いため、システムのブート時に、メモリへディスク・ラベルがコピーされます。システムにあるラベルではなく、ディスク上のラベルに直接アクセスする場合は、disklabel -r コマンドを使用してください。

注意

ディスクにデータがある場合は、ディスク・パーティションを変更する前に、すべてのファイル・システムをバックアップしてください。パーティションを変更すると古いファイル・システムのデータは上書きされ、壊れてしまいます。

パーティションの変更に対しては、次の規則が適用されます。

- 開始セクタを示すオフセットを変更したり、マウントされているファイル・システムのパーティションやオープン・ファイル記述子を持つファイル・システムのパーティションを縮小したりすることはできない。
- ディスク全体でパーティションが1つだけ必要な場合は、パーティション c を使用する。
- ラベルを変更する際は、a がマウントされていない限り、パーティション a に対して raw デバイスを指定して、a がディスクの開始部分(セクタ 0) から始まるようにしなければならない。パーティション a がマウントされている場合は、パーティション c を使用してラベルを変更する。この際、パーティション c もセクタ 0 から始まるようにしなければならない。

注意

パーティション a がすでにマウントされている場合は、デバイス・パーティション a を用いてディスク・ラベルを編集しようと

しても、変更できません。また、ラベルを書き込めなかったというエラー・メッセージが表示されることもありません。

ディスク・パーティションのサイズを変更する前に、ディスク・ラベルを表示して現在のパーティションの設定を調べてください。そのために、`disklabel` コマンドでパーティション・サイズを表示させます。パーティションのボトム、トップ、およびサイズは、512 バイト・セクタを 1 単位として表示されます。

現在のディスク・パーティション設定を調べるために、次の `disklabel` コマンドを実行してください。

```
/sbin/disklabel -r device
```

ディレクトリ名 (`/dev`) の後に raw デバイス名、ドライブ番号、およびパーティション `a` または `c` を続けて入力し、デバイスを指定してください。 `dsk1` のように、ディスク・ユニットと番号を指定することもできます。

`disklabel` コマンドでディスク・ラベルを表示させる例を示します。

```
# disklabel -r /dev/rdisk/dsk3a
type: SCSI
disk: rz26
label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0          # milliseconds
track-to-track seek: 0 # milliseconds
drivedata: 0

8 partitions:
#      size offset  fstype [fsize bsize cpgr] # (Cyl.   0 - 164*)
a: 131072      0   4.2BSD  1024  8192  16 # (Cyl. 164*- 492*)
b: 262144 131072   unused  1024  8192      # (Cyl.   0 - 2569)
c: 2050860      0   unused  1024  8192      # (Cyl. 492*- 1185*)
d: 552548 393216   unused  1024  8192      # (Cyl. 1185*- 1877*)
e: 552548 945764   unused  1024  8192      # (Cyl. 1877*- 2569*)
f: 552548 1498312   unused  1024  8192      # (Cyl. 492*- 1519*)
g: 819200 393216   unused  1024  8192      # (Cyl. 1519*- 2569*)
h: 838444 1212416   4.2BSD  1024  8192  16 # (Cyl. 1519*- 2569*)
```

パーティションの変更は、ファイル・システムのデータを上書きしたり、システムの効率を悪くしたりする可能性があるため、慎重に行う必要があります。パーティション・サイズの変更の際にパーティション・ラベルが壊れた場合は、次のように `disklabel` コマンドに `-w` オプションを付けて、省略時のパーティション・ラベルに復元することができます。

```
# disklabel -r -w /dev/rdisk/dsk1a rz26
```

`disklabel` コマンドを使えば、カーネルの再構築およびシステムのリブートを伴わずに、各ディスクのパーティション・ラベルを変更できます。パーティション・サイズを変更する場合は次の手順に従ってください。

1. `df` コマンドを使用して、ファイル・システムのディスク・スペースの情報を表示させます。
2. `/etc/fstab` ファイルを表示させて、スワップ領域として指定されているファイル・システムがないか確認します。
3. `disklabel` コマンドに `-r` オプションを指定して実行し、ディスク・ラベルを調べます (省略時のディスク・パーティションについての説明は、`rz(7)`、`ra(7)`、また `disktab(4)` を参照してください)。
4. ファイル・システムをバックアップします。
5. 変更したいラベルのあるディスクのファイル・システムをアンマウントします。
6. 新しいパーティション・パラメータを計算します。パーティション・サイズは、増減させることができます。また、パーティションどうしを重ねることもできます。
7. 次の構文で `disklabel` コマンドに `-e` オプションを指定して実行することにより、ディスク・ラベルを編集してパーティション・パラメータを変更します。

```
# /sbin/disklabel -e disk
```

`vi` エディタ、または `EDITOR` 環境変数で指定されたエディタが起動されて、ディスク・ラベルを編集できるようになります。このとき、ディスク・ラベルは、`disklabel -r` コマンドの場合と同じ形式で表示されます。

`-r` オプションを指定すると、可能であれば、ラベルはディスクに直接書き込まれ、メモリ内のコピーが更新されます。 `disk` パラメー

タには、アンマウントされているディスク (たとえば `dsk0` または `/dev/rdisk/dsk0a` など) を指定します。

エディタを終了して変更内容を保存すると、次のプロンプトが表示されます。

```
write new label? [?]:
```

新しいラベルを書き込む場合は、`y` を入力し、変更内容を廃棄する場合は、`n` を入力します。

8. `disklabel` コマンドに `-r` オプションを指定して実行し、新しいディスク・ラベルを表示させます。

5.2.3 `newfs` コマンドによるパーティションの重なりのチェック

ファイル・システムのマウントや作成を行うコマンド、新しいスワップ・デバイスを追加するコマンド、および Logical Storage Manager ヘディスクを追加するコマンドは、まず、コマンドに指定されたディスク・パーティションにすでに有効なデータが含まれていないかどうか、また、そのパーティションが使用中の他のパーティションと重なっていないかどうかについて、確認します。その際、指定されたパーティションまたは重なっているパーティションが使用中であるかどうかは、ディスク・ラベルの「`fstype`」フィールドで判断できます。

パーティションが使用されていないければ、コマンドは実行を継続します。ファイル・システムのマウントや作成だけでなく、`mount`、`newfs`、`fsck`、`voldisk`、`mkfmdmn`、`rmfmdmn`、および `swapon` のようなコマンドも、ディスク・ラベルを変更して、「`fstype`」フィールドにパーティションの用途を設定します。たとえば、AdvFS ドメインにディスク・パーティションを追加すると、「`fstype`」フィールドは AdvFS に設定されます。

パーティションが利用できない場合、これらのコマンドは次のようなエラー・メッセージを返して、処理を継続するかどうかを問い合わせてきます。

```
# newfs /dev/disk/dsk8c
WARNING: disklabel reports that basename,partition currently
is being used as "4.2BSD" data. Do you want to
continue with the operation and possibly destroy
existing data? (y/n) [n]
```

オペレーティング・システムのコマンド同様、アプリケーションもディスク・ラベルの `fstype` を変更して、パーティションが使用中であることを示

すことができます。詳細については、`check_usage(3)` および `set_usage(3)` を参照してください。

5.3 ディスクのコピー

ディスクまたはディスク・パーティション全体のコピーには `dd` コマンドを使用します。このコマンドを実行すると、ディスクまたはディスク・パーティションの物理的なコピーが作成されます。

注意

`dd` コマンドは複数のファイルをコピーするためのコマンドではありませんので、データ・ディスクとして使用しているディスクに、あるいは、ファイル・システムを含まないディスクに、単一のディスクまたはパーティションをコピーします。UFS ファイル・システムを含むディスクまたはパーティションをコピーする場合は、第 9 章で説明する `dump` および `restore` コマンドを使用してください。また、AdvFS ファイルセットを含むディスクまたはパーティションをコピーする場合は、『*AdvFS 管理ガイド*』で説明されている `vdump` および `vrestore` コマンドを使用してください。

ディスクの最初のブロックにはディスク・ラベルが保管されているため、オペレーティング・システムでは、有効なディスク・ラベルを持ったディスクのこのブロックを保護します。したがってディスク・ラベルが存在するターゲット・ディスク上のパーティションをコピーする場合は、そのターゲット・ディスク上のディスク・ラベルを保存するかどうかを決めなければなりません。

ターゲット・ディスク上のディスク・ラベルをそのままにしておく場合は、`dd` コマンドに `skip` または `seek` オプションを付けて実行し、ターゲット・ディスク上の保護されているディスク・ラベル領域をスキップします。ターゲット・ディスクの容量は、オリジナルのディスクと同じかそれ以上でなければなりません。

ターゲット・ディスクにラベルがあるかどうかを確認するには、次の `disklabel` コマンドを使用します。

```
# /sbin/disklabel -r target_disk
```

ターゲット・デバイスのディレクトリ名 (/dev) の後に raw デバイス名, ドライブ番号, およびパーティション c を続けて指定します。ディスクにラベルがない場合は, 次のメッセージが表示されます。

```
Bad pack magic number (label is damaged, or pack is unlabeled)
```

ディスクにすでにラベルが付いている場合の例は次のようになります。

```
# disklabel -r /dev/rdisk/dsk1c
type: SCSI
disk: rz26
label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0 # milliseconds
track-to-track seek: 0 # milliseconds
drivedata: 0

8 partitions:
#      size  offset  fstype [fsize bsize  cpgh]
a: 131072      0  unused 1024 8192 # (Cyl.  0 - 164*)
b: 262144 131072  unused 1024 8192 # (Cyl. 164*- 492*)
c: 2050860      0  unused 1024 8192 # (Cyl.  0 - 2569)
d: 552548 393216  unused 1024 8192 # (Cyl. 492*- 1185*)
e: 552548 945764  unused 1024 8192 # (Cyl. 1185*- 1877*)
f: 552548 1498312 unused 1024 8192 # (Cyl. 1877*- 2569*)
g: 819200 393216  unused 1024 8192 # (Cyl. 492*- 1519*)
h: 838444 1212416 unused 1024 8192 # (Cyl. 1519*- 2569*)
```

ターゲット・ディスクにすでにラベルが付いており, そのラベルがもう必要ない場合は, disklabel -z コマンドを使用してラベルをクリアしなければなりません。たとえば, 次のようにします。

```
# disklabel -z /dev/rdisk/dsk1c
```

オリジナル・ディスクをターゲット・ディスクにコピーし, ターゲット・ディスクのラベルをそのまま保存する場合は, dd コマンドを使用します。このコマンドには, デバイス・ディレクトリ名 (/dev) の後に raw デバイス名, ドライブ番号, オリジナルおよびターゲットのディスク・パーティションを続けて指定します。たとえば次のように指定します。

```
# dd if=/dev/rdisk/dsk0c of=/dev/rdisk/dsk1c \
skip=16 seek=16 bs=512k
```



UNIX ファイル・システム (UFS) の管理

この章では、ファイル・システムの概要を説明するとともに、ファイル・システムに関連する基本的なシステム管理作業について説明します。さまざまなファイル・システムがサポートされていますが、アプリケーションおよび UNIX オペレーティング・システムの構成要素で使用する主なファイル・システムは、AdvFS (Advanced File System) と UNIX ファイル・システム (UFS) です。出荷時にオペレーティング・システムがインストール済みのシステムでは、省略時のファイル・システムとして AdvFS が構成されています。AdvFS の管理についての詳細は、『*AdvFS 管理ガイド*』を参照してください。

この章では、各ファイル・システムに関して、次のようなシステム管理作業について説明します。

- 使用可能なファイル・システムの概要 (6.1 節)
- クラスタへの結合を容易にする、コンテキスト依存シンボリック・リンク (CDSL) (6.2 節)
- UFS ファイル・システムを手動で作成する方法 (6.3 節)
- SysMan Menu タスクを使用して UFS ファイル・システムを作成する方法 (6.4 節)
- ユーザにクォータを割り当てて UFS ファイル・システム・リソースを制御する方法 (6.5 節)
- UFS ファイル・システムのバックアップ方法についての参考資料 (6.6 節)
- ファイル・システムのモニタ機能とチューニング機能 (6.7 節)
- UFS ファイル・システムの問題のトラブルシューティング (6.8 節)

この他にも、システム管理作業およびファイル・システムに関する情報源があります。この章では、これらの情報源について必要に応じて説明します。

6.1 ファイル・システムの概要

ユーザがオペレーティング・システムをインストールする場合には、1 つまたは複数の UFS ファイル・システムを作成することができます。出荷時に AdvFS が構成されているシステムでも、UFS ファイル・システムを作成することができます。両方のファイル・システムを、システム上で同時に使用することができます。システム・ディスクの場合や、AdvFS の拡張機能を必要としない場合には、多くの管理者は使い慣れている UFS ファイル・システムを使用します。

オペレーティング・システムは、以下のファイル・システムを含む、さまざまなファイル・システムの最新のバージョンをサポートしています。

- AdvFS (Advanced File System)。このファイル・システムには、独立したドキュメントと拡張インタフェースがあります。詳細については、『*AdvFS 管理ガイド*』およびリファレンス・ページの `advfs(4)` を参照してください。AdvFS 用の拡張管理ユーティリティもあります。これらの拡張管理ユーティリティが使用可能な場合は、CDE アプリケーション・マネージャの「ストレージ管理」フォルダに「Advanced File System」という起動アイコンが表示されます。拡張管理ユーティリティの設定とその使用については、『*AdvFS 管理ガイド*』を参照してください。

基本 AdvFS Utilities は、SysMan Menu タスクとして提供されています。これらのタスクへのアクセスについては、第 1 章を参照してください。SysMan で提供されているユーティリティについては、オンライン・ヘルプが利用できます。

- この章で説明されている UNIX ファイル・システム (UFS)。属性とユーティリティについては、`ufs_fsck(8)`、`sys_attrs_ufs(5)`、および `tunefs(8)` を参照してください。
- ISO 9660 コンパクト・ディスク・ファイル・システム (CDFS)。詳細については、リファレンス・ページの `cdfs(4)` を参照してください。
- メモリ・ファイル・システム (mfs)。詳細については、`newfs(8)` を参照してください。
- `ffm` (File on File Mounting) ファイル・システム。詳細については、`ffm(4)` を参照してください。

また、次のマニュアルも参照してください。

- AdvFS および UFS の両方のファイル・システムでの LSM (Logical Storage Manager) の使用については、『*Logical Storage Manager*』を参照してください。
- UFS ファイル・システムから AdvFS ファイル・システムへの変換，またはその逆の変換については、『*AdvFS 管理ガイド*』を参照してください。
- UFS ファイル・システムの高度なチューニングについては、『*システムの構成とチューニング*』を参照してください。

この節の後半および以降の節では，ファイル・システムの作成とその管理での重要な概念について説明します。この情報は，基本的なファイル・システムの作成とその管理に必要な事項ではありませんが，高度な作業やトラブルシューティングを行う場合に役立ちます。

以降の項で詳細を説明する項目について，概要を次に示します。

ディレクトリ階層

ファイル・システムは，ローカルにマウントされているかリモートでマウントされているかにかかわらず，システムやクラスタのディレクトリ階層全体の一部です。ファイル・システムは，基本システム階層にディレクトリ階層を追加するたびにルート・ファイル・システム (/) から成長し，分岐するツリーであると考えられます。
/usr/usrs/projects などの UFS ファイル・システムを作成する場合は，既存の /usr/usrs ブランチの下に，新しいブランチとして追加します。

ディスク・パーティション

どのシステムでも，ファイル・システムを格納するための一般的な記憶域はハード・ディスクです。これらのデバイスの管理については，『*ハードウェア管理ガイド*』で説明しています。1つのディスクは，論理パーティションに分割されます。論理パーティションはディスク全体 (パーティション c) とすることも，パーティション a ~ h のようにディスクの一部とすることもできます。ディスクのサイズによりパーティションのサイズは異なり，そのサイズは通常 M バイト (MB) で示されます。最初にファイル・システムを作成する場合は，ディスク・パーティションにファイル・システムを作成し，限られたサイズ (ディスク・スペース) を割り当てます。UFS ファイル・システムのサ

イズを大きくするには、より大きなパーティションまたはディスクにそのファイル・システムを移動する必要があります。

ファイル・システムの構造

ファイル・システムには、物理メディア上のデータのレイアウトを示す、ディスク上のデータ構造があります。ファイル・システムのトラブルシューティングまたはチューニングなどの高度な作業を行う場合には、この構造を理解していなければなりません。大半の一般的な作業では、この情報を細かい点まで理解していなくても構いません。参照情報については、以降の項で説明します。

ディレクトリおよびファイルのタイプ

さまざまなディレクトリおよびファイルのタイプが、一般的なコマンドの出力に表示されます。シンボリック・リンクやソケットなどのファイル・タイプを識別できるように、参照情報が用意されています。詳細については、次のリファレンス・ページを参照してください。

| | |
|------------|---|
| 通常ファイル | <code>file(1)</code> |
| ディレクトリ | <code>ls(1)</code> および <code>dir(4)</code> |
| デバイス特殊ファイル | 『ハードウェア管理ガイド』 |
| ソケット | <code>socket(2)</code> 、『ネットワーク管理ガイド：接続編』、および『ネットワーク・プログラミング・ガイド』 |
| パイプ | <code>pipe(2)</code> |
| シンボリック・リンク | <code>link(1)</code> および <code>ln(1)</code> |

6.1.1 ファイル・システムのディレクトリ階層

ファイル・システムの位置は、ルート (/) ディレクトリから始まる UNIX ディレクトリ階層を基準にしています。作成したファイル・システムは、ディレクトリ階層のマウント・ポイントにマウントされたときに使用可能

(有効) になります。たとえば、オペレーティング・システムのインストール中に、usr ファイル・システムを (UFS として) 作成したとします。このファイル・システムは、ルート (/) に自動的にマウントされ、パス名はこの階層内の /usr となります。

システムの標準ディレクトリ階層は、効率的に構成されています。ディレクトリ階層により、機能および目的別にファイルが分類されます。ファイル・システムの効果的な使用方法の一例としては、.cshrc、.profile または .login などのユーザ設定ファイルで指定された通常の検索パスにあるディレクトリにコマンド・ファイルを置くことがあります。ディレクトリの中には、実際にはシンボリック・リンクされているものもあります。X11 ウィンドウ・システムの階層を含む、オペレーティング・システムのディレクトリ階層についての詳細は、リファレンス・ページの hier(5) を参照してください。

ファイル・システムは、マウントすることにより使用可能になります。システムのルート・ディレクトリ下のファイル・システム階層にファイル・システムを接続 (マウント) するには、mount コマンドを使用します。ファイル・システムの接続解除 (アンマウント) には、umount コマンドを使用します。ファイル・システムをマウントする場合、ファイル・システムを接続する位置 (システムのルート・ディレクトリ下のマウント・ポイント) を指定します。ファイル・システムのマウントおよびアンマウントについての詳細は、mount(8) を参照してください。

マウントされたファイル・システムのルート・ディレクトリは、そのマウント・ポイントでもあります。システムは、ルート・ディレクトリをシステム初期化ファイルのソースとして使用するため、システムにはルート・ディレクトリは 1 つしか存在しません。この結果、オペレーティング・システムのローカル・ファイル・システムはすべて、システムのルート・ディレクトリの下にマウントされます。

6.1.2 ディスク・パーティション

ディスクは、セクタと呼ばれる物理的な記憶ユニットから構成されています。各セクタは通常 512 バイトです。セクタは論理ブロック番号 (LBN) によってアドレス指定されます。LBN は、アドレス指定によるユーザ・アクセスが可能なデータ領域の基本単位です。最小の LBN は 0 で、最大の LBN は、ユーザ・アクセス可能領域の LBN 数より 1 つ少ない値です。

各セクタはグループ化され、最大 8 個のディスク・パーティションを形成します。ただし、パーティションの数とサイズはディスクによって異なります。/etc/disktab ファイルには、システムでサポートされるディスクおよび省略時のパーティション・サイズのリストが含まれています。詳しい説明は、disktab(4) を参照してください。

ディスク・パーティションはディスクの論理的な区画で、これにより、ファイルをさまざまなサイズの別々の領域に書き込むことが可能になります。パーティションは、ファイル・システムと呼ばれる構造でデータを保管し、ページングおよびスワッピング等のシステム操作で使うことができます。ファイル・システムは、hier(5) で説明されているようなディレクトリおよびファイルの階層構造を持っています。

ディスク・パーティションにはディスク・タイプごとに省略時のサイズがあり、このサイズは disklabel コマンドまたは diskconfig グラフィカル・ユーザ・インタフェースを使用して変更することができます。各パーティションには、a から h までの名前を割り当てることができます。1 つのパーティションに割り当てられたスペースが別のパーティションと重なることは可能ですが、省略時のパーティションが重なり合うことはありません。また、適切に使用されているディスクでは、相互に重なり合うパーティションにはファイル・システムを置くことはありません。

例 6-1 では、モデル RZ1DF-CB ディスクの省略時のパーティションの例を、次のコマンドを使用して示します。

```
# disklabel -r /dev/rdisk/dsk0a
```

この出力例では、ディスク・テーブルの部分だけを示します。また、rz(7) に示されている RZ1DF-CB ディスクと HSZ RAID ディスクの例も、次にリストします。

例 6-1: 省略時のパーティション

(RZ1DF-CB Disk)

```
8 partitions:
#          size      offset      fstype    [fsize bsize    cpgh] # NOTE: values not exact
a:      262144         0      4.2BSD    1024  8192     16   # (Cyl.   0 - 78*)
b:      1048576    262144      swap                    # (Cyl.  78* - 390*)
c:      17773524         0    unused          0      0                    # (Cyl.   0 - 5289*)
d:      1048576    1310720      swap                    # (Cyl. 390* - 702*)
e:      9664482    2359296    AdvFS                    # (Cyl.  702* - 3578*)
f:      5749746    12023778    unused          0      0                    # (Cyl. 3578* - 5289*)
g:      1433600     524288    unused          0      0                    # (Cyl. 156* - 582*)
h:      15815636    1957888    unused          0      0                    # (Cyl. 582* - 5289*)
```

例 6-1: 省略時のパーティション (続き)

HSZ10, HSZ40, HSZ50, HSZ70 (RAID) Partitions

| Disk | Start | Length |
|-------|--------|--------------|
| dsk?a | 0 | 131072 |
| dsk?b | 131072 | 262144 |
| dsk?c | 0 | end of media |
| dsk?d | 0 | 0 |
| dsk?e | 0 | 0 |
| dsk?f | 0 | 0 |
| dsk?g | 393216 | end of media |
| dsk?h | 0 | 0 |

ディスク・ラベルは、ディスクの最初のセクタ内のブロック 0 (ゼロ) に置かれます。ディスク・ラベルは、ディスクのジオメトリと、そのディスクがどのようなパーティションに分割されているかに関する詳細な情報を提供します。システム・ディスク・ドライバおよびブート・プログラムは、ディスク・ラベルの情報を使用して、ドライブ、ディスク・パーティション、およびファイル・システムを認識します。その他の情報は、ディスクを最も効率的に使用したり、重要なファイル・システムの情報を検索したりするためにオペレーティング・システムによって使用されます。

各パーティションのディスク・ラベルの記述には、パーティションのタイプ (たとえば、標準ファイル・システム、スワップ領域など) を識別する識別子が含まれています。ディスク・ラベルには 2 つのコピーが存在し、ひとつはディスク上に、もうひとつはシステム・メモリの中にあります。I/O を実行するよりもシステム・メモリにアクセスする方が速いため、システムがディスクを認識すると、メモリにそのラベルがコピーされます。ファイル・システムについての情報が不完全な場合、ファイル・システムはラベルのメモリ内コピーをアップデートします。ラベルは `disklabel` コマンドを使用して変更することができます。コマンド行インタフェースについての詳細は、`disklabel(8)` を参照してください。ディスク構成ユーティリティ `diskconfig` については、『ハードウェア管理ガイド』を参照してください。

6.1.3 UFS バージョン 4.0

UFS の現在のバージョンは、リビジョン 4.0 です。このバージョンのディスク上のデータ・レイアウトは、6.1.4 項に説明されている UFS バージョン 3.0 のデータ・レイアウトと同じですが、データ容量が大きくなっています。

バージョン 3.0 では、32767 個のハード・リンクまたはサブディレクトリをサポートしていますが、バージョン 4.0 では、65533 個のハード・リンクまたはサブディレクトリをサポートしています。実際のディレクトリ数は、32765 (32k) と 65531 (64k) です。これは、空のディレクトリに、自分自身と親ディレクトリへのハード・リンクがすでに存在するためです。ls -a コマンドを使用すると、これらのリンクはそれぞれ . と . . で表示されます。以降この項の例は、32k 個を超えるハード・リンクを持つファイルにも適用されますが、すべて 32k のサブディレクトリについて説明します。

考慮すべき事項と重要な制限 (特に両方のバージョンを使用する場合) について次に示します。

newfs コマンドまたは diskconfig コマンドによるファイル・システムの作成

newfs コマンドまたは diskconfig 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と、非互換の問題を最小限にするために、新しいファイル・システムは必ずバージョン 3.0 (32k のサブディレクトリまたはハード・リンク) として作成されます。

fsck コマンドによるファイル・システムの検査

fsck コマンドを使用して、問題のあるファイル・システム (システム・クラッシュなどの理由により正常にアンマウントされていないなど) を検査する場合は、見つかったサブディレクトリの最大数により、そのファイル・システムがバージョン 3.0 かバージョン 4.0 かがマークされます。fsck コマンドで 32K を超える数のサブディレクトリが見つかった場合は、そのファイル・システムはバージョン 4.0 とマークされます。fsck コマンドで 32K を超える数のハード・リンクを持つディレクトリが見つからない場合は、そのファイル・システムはバージョン 3.0 とマークされます。ファイル・システムは通常、ユーザが 32K のサブディレクトリの制限数を超えて使用するとすぐに、バージョン 4.0 に変換されます。

fsck コマンドに新しいオプション -B が追加されました。このオプションを使用すると、バージョン 4.0 のファイル・システムをバージョン 3.0 のファイル・システムに変換することができます。fsck コマンドでこのオプションを使用すると、そのファイル・システムに 32K を超える数のサブディレクトリを持つディレクトリがなく、か

つ 32K を超える数のハード・リンクを持つファイルが存在しない場合にだけ、変換が行われます。

以前のバージョンのオペレーティング・システム (V4.0F など) を実行しているシステムで UFS バージョン 3.0 とバージョン 4.0 の両方を使用する場合の重要な制限を次に示します。

- 32K を超える数のサブディレクトリを含むディレクトリがそのファイル・システムに存在しないことを確認しない限り、バージョン 4.0 のファイル・システムに対して、以前のバージョンの `fsck` を `-p` または `-y` オプションを使用して実行しないでください。使用した場合には、32K を超える数のサブディレクトリを含むディレクトリがそのファイル・システムから永久に削除されます。
- 32K を超える数のサブディレクトリを含むディレクトリは、`/etc/fstab` ファイル中で、ルート (/) パーティション、`/usr` パーティション、または他のパーティションにリストしないでください。ブート時に、`/etc/fstab` ファイルに定義されているすべてのファイル・システムに対して `fsck -p` コマンドが自動的に実行されます。

この実行を防ぐために、バージョン 4.0 ではメイン・スーパーブロックと代替スーパーブロックが不一致となるようにします。この不一致により、`fsck -p` の古いバージョンをバージョン 4.0 のファイル・システムに対して実行することはできなくなります。32K を超える数のサブディレクトリを持つバージョン 4.0 のファイル・システムに対して `fsck -p` の古いバージョンを実行すると、代替スーパーブロックとスーパーブロックの不一致により、処理が失敗します。代替スーパーブロックの指定を促すプロンプトが表示された場合には、必ず `n` を入力してください。誤って `y` を入力した場合でも、次のプロンプトに対して `y` と入力しなければ、バージョン 4.0 のファイル・システムには何も処理が行われません。

```
CLEAR? [yn]
```

必要に応じて、ここで、`FREE BLK COUNT` と `UPDATE STANDARD SUPERBLOCK` を修正することができます。ただし、バージョン 4.0 のファイル・システムに対して、2 度目の `fsck -p` を実行したときには、この不一致による保護は行われません。32K を超える数のサブディレクトリを含むディレクトリは、永久に削除されます。

バージョン 3.0 から バージョン 4.0 への自動変換

この 2 つの UFS バージョンの間にはディスク上のデータ・レイアウトに違いがないため、従来のバージョン 3.0 のファイル・システムを最新のリリースのオペレーティング・システムにマウントすることができます。バージョン 3.0 のファイル・システム上で 32K を超える数のハード・リンクを作成しようとする、そのファイル・システムはバージョン 4.0 のファイル・システムに自動的に変換されます。この変換時に出力されるシステム・メッセージの例を次に示します。

```
Marking /dev/disk/dsk023 as Tru64 UNIX UFS v.4
```

バージョン 3.0 から バージョン 4.0 への手動変換

32K を超える数のサブディレクトリを持たないバージョン 4.0 のファイル・システムを共用またはマウントする場合は、バージョン 3.0 のファイル・システムだけをサポートする以前のオペレーティング・システム（たとえば Tru64 UNIX バージョン 4.0F）を実行しているシステムにマウントすることができます。ただし、最初にバージョン 4.0 のファイル・システムを次のように変換しなければなりません。

- バージョン 3.0 をサポートするシステムで、次の例のように、`fsck` コマンドをファイル・システム・パーティションに対して実行します。

```
# fsck /dev/rrz03
```

- バージョン 4.0 をサポートするシステムで、次の例のように、`fsck` コマンドをファイル・システム・パーティションに対して実行します。

```
# fsck -B /dev/disk/dsk34d
```

6.1.4 ファイル・システムの構造: UFS

UFS ファイル・システムは次の 4 つの主要部分から構成されます。

- ブート・ブロック
- スーパブロック
- i ノード・ブロック
- データ・ブロック

これらのブロックについて、以降の項で説明します。

AdvFS ファイル・システムの構造については『*AdvFS* 管理ガイド』で説明されています。

6.1.4.1 ブート・ブロック

各ファイル・システムの最初のブロック (ブロック 0) は、ブートまたは初期化プログラムのために予約されています。

6.1.4.2 スーパブロック

各ファイル・システムのブロック 1 はスーパブロックと呼ばれ、次の情報が含まれています。

- ファイル・システムのトータル・サイズ (ブロック単位)
 - *i* ノード用に予約されているブロック数
 - ファイル・システムの名前
 - デバイスの識別子
 - スーパブロックの最終更新日付
 - ファイル・システム内のすべての空きブロック (割り当てに利用可能なブロック) が収められている、空きブロック・リストのヘッド
- 新しいブロックをファイルに割り当てる場合、そのブロックは空きブロック・リストから取得されます。ファイルを削除すると、ブロックは空きブロック・リストに戻されます。
- 新しく作成されるファイルへの割り当てに利用可能な、未使用の *i* ノードのリスト

6.1.4.3 *i* ノード・ブロック

スーパブロックの次のブロックのグループ。これらの各ブロックには多くの *i* ノードが含まれています。各 *i* ノードには、*i* ノード番号が対応しています。*i* ノードは、ファイル・システム内の個々のファイルの管理情報を記述しています。ファイル・システムのファイルごとに 1 つの *i* ノードがあります。各ファイル・システムには *i* ノード数の上限があるため、ファイル・システムに収めることができるファイルには上限があります。*i* ノード数の上限はファイル・システムのサイズによって異なります。

各ファイル・システムの最初の *i* ノード (*i* ノード 1) には名前がなく、使用されません。2 番目の *i* ノード (*i* ノード 2) はファイル・システムのルート・

ディレクトリと対応していなければなりません。ファイル・システム内のファイルはすべて、ファイル・システムのルート・ディレクトリの下に存在します。i ノード 3 以降は、どの i ノードをどのファイルに割り当てても構いません。また、どのデータ・ブロックをどのファイルに割り当てても構いません。i ノードとブロックの割り当てには、特に順番はありません。

i ノードをファイルに割り当てた場合、i ノードには次の情報を含めることができます。

- ファイル・タイプ

ファイル・タイプには、通常ファイル、デバイス・ファイル、名前付きパイプ、ソケット、およびシンボリック・リンク・ファイルがあります。

- ファイル所有者

i ノードには、ファイルの所有者を識別するためのユーザおよびグループの識別番号が含まれています。

- ファイル保護情報

保護情報によって、ファイル所有者、グループ・メンバ、その他のユーザに対する読み取り、書き込み、および実行アクセスが指定されます。保護情報には、`chmod` コマンドで指定されるその他のモード情報もあります。

- リンク数

ディレクトリ・エントリ (リンク) は、ファイル名および i ノード番号から構成されます。リンク数は、ファイルを参照するディレクトリ・エントリの数を示します。リンク数が 0 の場合、ファイルは削除されます。つまり、ファイルの i ノードは、未使用 i ノードのリストに戻され、対応するデータ・ブロックは空きブロック・リストに戻されます。

- ファイル・サイズ (バイト単位)

- 最終ファイル・アクセス日付

- 最終ファイル変更日

- 最終 i ノード変更日

- データ・ブロックへのポインタ

このポインタは、物理ディスク上でのデータ・ブロックの実際の位置を示します。

6.1.4.4 データ・ブロック

データ・ブロックは、ユーザ・データあるいはシステム・ファイルを保持します。

6.1.5 ディレクトリおよびファイルのタイプ

オペレーティング・システムはファイルをビット・ストリームとして認識し、ディスク上のデータ、名前付きパイプ、UNIX ドメイン・ソケット、および端末をファイルとして処理することができます。このオブジェクト・タイプ透過性により、さまざまなストレージおよび通信機能の定義と動作のためのメカニズムが簡素化されています。オペレーティング・システムは、内部処理を構成して管理する際に、さまざまなレベルの抽象表現を扱います。

外部インタフェースだけでなく、システムが認識するさまざまなファイル・タイプを理解する必要があります。システムは、次のファイル・タイプをサポートしています。

- 通常ファイル
たとえば、プログラムやテキスト・ファイル、またはソース・コードなどの形式でデータが含まれています。
- ディレクトリ
通常ファイルの一種で、ファイルの名前や他のディレクトリの名前が含まれます。
- 文字型デバイス特殊ファイルおよびブロック型デバイス特殊ファイル
システムの物理デバイスと擬似デバイスを識別します。
- UNIX ドメイン・ソケット・ファイル
ネットワーク・プロセス間の接続を提供します。ソケット・ファイルは `socket` システム・コールによって作成されます。
- 名前付きパイプ
プロセスが互いに通信し合うために使用するデバイス・ファイルです。
- ターゲット・ファイルまたはディレクトリをポイントするリンク・ファイル
リンク・ファイルには、ターゲット・ファイルの名前が含まれています。シンボリック・リンクされたファイルとそのターゲット・ファイルは、同じファイル・システムに存在しても、別のファイル・システムに存在

しても構いません。ハード・リンクされたファイルとそのターゲット・ファイルは、同じファイル・システムに存在しなければなりません。

6.1.6 デバイス特殊ファイル

デバイス特殊ファイルは、物理デバイス、擬似デバイス、および名前付きパイプを表しています。デバイス特殊ファイルは /dev ディレクトリに置かれます。デバイス特殊ファイルは、システムとデバイス・ドライバの間のリンクの役割を果たします。それぞれのデバイス特殊ファイルは、物理デバイス(たとえば、ディスク、テープ・ドライブ、プリンタ、または端末)、または擬似デバイス(たとえば、ネットワーク・インタフェース、名前付きパイプ、または UNIX ドメイン・ソケット)に対応しています。デバイス・ドライバはすべての読み取りおよび書き込み操作を行い、デバイスが必要とするプロトコルに従います。

次の 3 種類のデバイス特殊ファイルがあります。

- ブロック型デバイス特殊ファイル

ドライバが大きなブロック単位で I/O を処理するようなデバイス、およびカーネルが I/O バッファリングを処理するようなデバイスに使用されます。ディスクなどの物理デバイスは、ブロック型デバイス特殊ファイルとして定義されます。/dev ディレクトリのブロック型デバイス特殊ファイルの例は次のとおりです。

```
brw----- 1 root system 8, 1 Jan 19 11:20 /dev/disk/dsk0a
brw----- 1 root system 8, 1 Jan 19 10:09 /dev/disk/dsk0b
```

- 文字型デバイス特殊ファイル

ドライバが独自の I/O バッファリングを処理するようなデバイスに使用されます。ディスク、端末、擬似端末、およびテープ・ドライバは、通常、文字型デバイス・ファイルとして定義されます。/dev ディレクトリの文字型デバイス特殊ファイルの例は次のとおりです。

```
crw-rw-rw- 1 root system 7, 0 Jan 31 16:02 /dev/ptyp0
crw-rw-rw- 1 root system 7, 1 Jan 31 16:00 /dev/ptyp1
crw-rw-rw- 1 root system 9,1026 Jan 11 14:20 /dev/rtape/tape_01
```

文字型デバイス特殊ファイルは、次のような raw ディスク・デバイスのケースもあります。

```
crw-rw-rw- 1 root system 7, 0 Jan 10 11:19 /dev/rdisk/dsk0a
```

- ソケット・デバイス・ファイル

プリンタ・デーモン (lpd) およびエラー・ログ・デーモン (syslogd) に使用されます。/dev ディレクトリのソケット・デバイス・ファイルの例は次のとおりです。

```
srw-rw-rw- 1 root  system    0 Jan 22 03:40 /dev/log
srwxrwxrwx 1 root  system    0 Jan 22 03:41 /dev/printer
```

デバイス特殊ファイルとその命名規則の詳細については、『ハードウェア管理ガイド』を参照してください。

6.2 コンテキスト依存シンボリック・リンクとクラスタ

この節では、システムを結合してクラスタにするディレクトリ階層機能である、コンテキスト依存シンボリック・リンク (CDSL) について説明します。CDSL には、すべてのシステムのファイル・システムとディレクトリ階層 (現在クラスタのメンバではないものも含む) に対する、一定の必要条件があります。必要条件は、次のとおりです。

- ルート (/), /var, および /usr の各ファイル・システムには、単一システムでは使用されることのない /cluster サブディレクトリがあります。このサブディレクトリは削除しないでください。削除すると、システムをクラスタに追加できなくなります。
- システムがクラスタに結合されると、そのシステムはクラスタのメンバとなります。どのファイルにも、メンバ・システムで一意的識別子 (メンバ固有) を含む、一意のパス名があります。これらのパス名は、コンテキスト依存シンボリック・リンク (CDSL) と呼ばれます。CDSL は、パス名に変数要素を持つシンボリック・リンクです。変数要素は各クラスタ・メンバごとに異なり、アプリケーションまたはコマンドで解決されてコンテキストになります。
- 一部の重要なシステム・ファイルは、ターゲット・ロケーションを指す一意の CDSL を持つターゲット・ディレクトリにあります。これにより、共用 (クラスタ全体) ファイルと非共用 (メンバ固有) ファイルを分離することができます。
- CDSL が移動または破壊されている場合には、アップデート・インストールが失敗することがあります。

ディレクトリの構造については、hier(5) を参照してください。

CDSL を使用すると、クラスタのメンバとして結合されたシステムは、共用する必要があるすべてのファイルとディレクトリに対してグローバルな

名前領域を持つことができます。CDSL により、基本構成要素とレイヤード・アプリケーションをクラスタ対応にすることができます。共用ファイルとディレクトリは、クラスタと単一システムで同じように動作します。また、ファイル・システムの管理ツールも、クラスタと単一システムで同じように動作します。

システムが将来的にクラスタ・メンバになる可能性があるために、CDSL が重要な場合には、以降の項を参照してください。CDSL がないことを示すエラー（アップデート・インストレーションの失敗など）が発生した場合は、以降の項で説明するように、CDSL の保守や、確認、修正を行わなければならないことがあります。

6.2.1 関連ドキュメント

CDSL の説明については、次のマニュアルを参照してください。

- 『インストレーション・ガイド』では、アップデート・インストレーションについて説明しています。アップデート・インストレーション処理についての詳細は、`installupdate(8)` を参照してください。

TruCluster のドキュメントでは、システムのクラスタへの追加処理と、動作中のクラスタでの CDSL の使用方法について説明しています。このドキュメントは、ベース・ドキュメント・セットには入っていません。

- `local(4)`、`ls(1)`、`ln(1)`、および `hier(5)` のリファレンス・ページには、参照情報とコマンドの情報が 있습니다。

リファレンス・ページの `cdslinvchk(8)` では、単一システムのインストールまたはアップデート時にそのシステム上の全 CDSL のインベントリを作成するために使用する、`/usr/sbin/cdslinvchk` スクリプトについて説明しています。

6.2.2 CDSL の説明

個々のシステムを接続してクラスタとし、ユーザからは 1 つのシステムに見えるようにできます。1 つのクラスタ内の単一システムは、メンバと呼ばれます。Tru64 UNIX クラスタの説明については、TruCluster のドキュメントを参照してください。クラスタ処理を容易にするために、ファイル・システムには、一部のファイルを個々のクラスタ・メンバ内で一意とする構造と識別可能なパス名、およびメンバ固有の情報がなければなりません。

この他のファイルは、クラスタ内のすべてのメンバで共用する必要があります。CDSL パス名を使用すると、クラスタ内の異なるシステムが同じファイル階層を共用することができます。ユーザとアプリケーションは、共用であるかメンバ固有であるかに関係なく、従来のパス名を使用してファイルとディレクトリにアクセスできます。

たとえば、2つのシステムがスタンドアロンである場合または単にネットワーク・リンクで接続されている場合は、登録ユーザの情報を含む `/etc/passwd` ファイルが、それぞれのシステムにあります。2つのシステムが1つのクラスタのメンバの場合は、両方のシステムの登録ユーザの情報を含む共通の `/etc/passwd` ファイルを共用します。

他の共用ファイルは次のとおりです。

- `/etc/timezone` または `/etc/group` などの、システム固有ではなく、サイト固有の構成ファイルとディレクトリ
- `/bin` または `/usr/bin` などの、カスタマイズされた情報を含まないファイルとディレクトリ
- クラスタ全体で使用可能なディスクとテープのデバイス特殊ファイル

一部のファイルは、必ずメンバ固有でなければなりません。つまり、共用可能でないファイルがあります。メンバ固有ファイルの例は `/etc/rc.config` で、共用ファイルの例は `rc.config.common` です。これらのファイルには、個別のシステムにだけ適用される構成情報、またはクラスタ内のすべてのメンバに適用される構成情報が含まれます。クラスタ・システムで CDSL を使用すると、ファイルを共用したり、メンバ固有ファイルを識別して保守することができます。この他にメンバ固有ファイルには、次のカテゴリがあります。

- `/var/adm/crash` などのディレクトリ。これらのディレクトリには、アプリケーション、ユーティリティ、またはデーモンが作成した、個別のクラスタ・メンバにだけ適用されるファイルが置かれます。
- `/dev` および `/devices` に置かれたデバイス特殊ファイルの一部。
- メンバ固有のデバイス特殊ファイルを参照する、`/etc/securettys` などの構成ファイル。
- ブート時または構成時に使用する、`/vmunix` および `/etc/sysconfigtab` などのプロセス固有ファイル。

システムがクラスタに接続されていない場合でも、パス名はまだ存在しています。ただし、これはユーザには透過的です。クラスタ・ファイルの命名規則を理解し、そのファイル構造を維持しなければなりません。CDSL を誤って削除した場合は、再作成する必要があります。

6.2.2.1 CDSL の構造

CDSL は単に、`ln(1)` で説明されているシンボリック・リンクです。このリンクには、クラスタ・メンバである各システムを識別する変数が含まれています。この変数は、実行時にターゲットに対応して解決されます。CDSL の構造は次のとおりです。

```
/etc/rc.config -> /cluster/members/{memb}/etc/rc.config
```

クラスタがサポートされる前は、このファイルのパス名は `/etc/rc.config` でした。このファイルは現在、CDSL によってメンバ固有のターゲットにリンクされています。このリンクの構造は、次のように解釈されます。

- `/cluster` ディレクトリはルート・ディレクトリに存在し、共用ファイルまたはメンバ固有ファイル (この例の場合) のパスを含みます。
- `/cluster/members/` ディレクトリには、ローカル・メンバの識別子 `member0` のディレクトリと、変数パス要素 `{memb}` へのリンクが置かれます。`/cluster/members/member0` ディレクトリには、`devices` や `etc` などの、メンバ固有のシステム・ディレクトリが置かれます。
- `{memb}` 変数パス要素を使用して、クラスタの個々のメンバが識別されます。この変数は、実行時に `sysconfigtab` 変数 `generic:memberid` の値が付加されて、`member` になります。この変数の省略値はゼロです。この値は、クラスタの各メンバに固有です。

ルート内の `/.local..` ファイルは、`cluster/members/{memb}` へのリンクで、システム固有ファイルを定義します。システム固有ファイルは、`/.local..` パス名で参照または作成できます。`/.local../etc/[filename]` として作成されたファイルは、`/etc` が共用ディレクトリのため、`/etc/[filename]` のパスでアクセスすることはできません。このファイルには、`/.local../etc/[filename]` および `/cluster/members/{memb}/etc/[filename]` のパス名でのみアクセスできます。

単一システムが他のシステムと一緒にクラスタ化されない場合は、`generic:memberid` 変数には自動的にゼロが設定されます。単一システム上の一般的な CDSL の例を次に示します。

```
/cluster/members/{memb}/etc/rc.config
```

この CDSL は、次のように解決されます。

```
/cluster/members/member0/etc/rc.config
```

システムが他の 2 つのシステムと一緒にクラスタ化され、`generic:memberid` 変数に 3 が設定された場合、同じ CDSL は次のように解決されます。

```
/cluster/members/member3/etc/rc.config
```

クラスタで動作中の場合は、メンバ固有のファイルは次の 3 つの方法で参照できます。

- 特定のシステムからメンバ固有の形式または共用の形式で参照する (たとえば、`/var/adm/crash/crash-data.5`)。
- 特定のシステムからメンバ固有の形式でのみ参照する (たとえば、`/.local../var/adm/crash/crash-data.5`)。
- クラスタの任意のメンバから参照する (たとえば、`/cluster/members/member0/var/adm/crash/crash-data.5`)。

クラスタのメンバに対してだけ、次の 2 つの特別な CDSL があります。

- ミニルート
- 特別な非共用ディレクトリ
 - `/dev -> /cluster/members/{memb}/dev`
 - `/tmp -> /cluster/members/{memb}/tmp`

詳細については、TruCluster のドキュメントを参照してください。

6.2.3 CDSL の管理

シンボル・リンクされたファイルに適用される保護は、すべてのファイルで行われている一般的な、ユーザおよびファイル・アクセス・モードによる保護だけです。特別な保護はありません。また、CDSL にも特別な保護はありません。単独システムで CDSL が破壊された場合に発生する障害を、次にいくつか示します。

- オペレーティング・システムのアップデート・インストールが行われたとき。

クラスタ外のシステムで CDSL がないことが分かるのは、アップデート・インストール処理 `installupdate(8)` を使用してオペレーティング・システムをアップデートして、障害が発生した場合だけです。この問題を回避するには、`/usr/sbin/cdslinvchk` スクリプトを必ず実行してシステムの CDSL の状態を調べてから、アップデート・インストールを実行します。

- ユーザまたはアプリケーションがメンバ固有の CDSL を移動または削除したとき。

メンバ固有の CDSL が、`rm` または `mv` コマンドで誤って削除されることがあります。この問題を回避するために、手動でファイルを編集したり、作成しないでください。代わりに、`vi` (`/etc/passwd` の編集用) などのツールを使用してファイルを編集してください。システム管理のツールとユーティリティはすべて CDSL を認識しているため、システム・ファイルの管理に適しています。

6.2.3.1 CDSL インベントリの検査

単独システム上の CDSL インベントリを検査するには、`/usr/sbin/cdslinvchk` スクリプトを使用します。定期的にインベントリを更新して、CDSL を確認します。詳細は、`cdslinvchk(8)` を参照してください。

6.2.3.2 CDSL の作成

CDSL が誤って破壊された場合または CDSL を新たに作成する必要がある場合のリンクの修正処理または作成処理については、`ln(1)` を参照してください。たとえば、`/etc/rc.config` リンクが破壊された場合は、リンクを次のように作成します。

- `sysconfigtab` 変数 `generic:memberid` で定義されている `{memb}` の値を確認します。
- ファイルが存在するかどうか確認します。たとえば、次のように確認します。

```
# ls /cluster/members/member3/etc/rc.config
```
- `generic:memberid` が 3 の場合は、次のように新しいリンクを作成します。

```
# cd /etc
# ln -s /cluster/members/member3/etc/rc.config
```


6.3 UFS ファイル・システムの手動作成

オペレーティング・システムの基本ファイル・システムの構成は、インストール中にシステムのルート・ファイル・システムが構築されるときに定義されます。インストール後は、必要に応じてファイル・システムを作成することができます。以降の項では、ファイル・システムをコマンド行で手動で作成する方法について説明します。コンソールで作業していて、システムがシングルユーザ・モードでグラフィカル・ユーティリティが使用できないときは、ファイル・システムに対する操作をコマンド行で行わなければなりません。

AdvFS ファイル・システムの作成については、『*AdvFS 管理ガイド*』を参照してください。

6.3.1 newfs による新しいファイル・システムの作成

ファイル・システムの一般的な作成手順を次に示します。

1. 新しいパーティションとして使用するディスク・デバイスと raw ディスク・パーティションを確認します。このパーティションが正しくラベル付けされてフォーマットされていることと、まだ使用されていないことを確認します。コマンド行インタフェース `hwmgr` および `dsfmgr` を使用して、デバイスを指定したり、新しいデバイスを追加してデバイス特殊ファイルを作成したりします。この手順については、『ハードウェア管理ガイド』を参照してください。このコマンドのオプションについては、`hwmgr(8)` と `dsfmgr(8)` を参照してください。

必要な場合には、`disklabel -p` コマンドを使用して、ディスクの現在のパーティションの状態を読み取ります。`/etc/fstab` ファイルを調べて、パーティションがファイル・システムにまだ割り当てられていないことと、スワップ・デバイスとして使用されていないことを確認します。詳細については、`disklabel(8)` および `fstab(4)` を参照してください。

2. 使用する未使用の raw (文字型) ディスク・パーティションの確認が完了すると、そのパーティションのデバイス特殊ファイルの名前が分かります。たとえば、ディスク 2 のパーティション `g` のデバイス特殊ファイル名は `/dev/rdisk/dsk2g` です。デバイス特殊ファイルの名前については、『ハードウェア管理ガイド』を参照してください。

3. `newfs` コマンドを使用して、ファイル・システムをターゲット・パーティションに作成します。詳細については、`newfs(8)` を参照してください。
4. マウント・ポイントのディレクトリを作成し、`mount` コマンドで新しいファイル・システムをマウントして使用できる状態にします。リブート後もマウントされるようにしたい場合は、`mount` コマンドを `/etc/fstab` ファイルに追加します。ファイル・システムをエクスポートしたい場合は、そのファイル・システムを `/etc/exports` ファイルに追加します。詳細については、`mount(8)` を参照してください。
5. `chmod` コマンドを使用して、アクセス制御の制限のチェックと調整を行います。詳細については、`chmod(1)` を参照してください。

これらの手順については、この項の残りの部分で詳細に説明します。

`newfs` コマンドは、ディスク・パーティションを初期化して UFS ファイル・システムを作成します。`newfs` コマンドは、ディスク・ラベルの情報または `/etc/disktab` ファイルに指定された省略時の値を使用して、特定のディスク・パーティションにファイル・システムを作成します。`newfs` コマンドのオプションを使用して、ディスクのジオメトリを指定することもできます。

注意

省略時のディスクのジオメトリの値を変更すると、標準スーパブロックが失われた場合に、`fsck` プログラムは代替スーパブロックを検出できなくなります。

`newfs` コマンドの構文は次のとおりです。

`/sbin/newfs` `[-N]` `[newfs_options]` `special_device` `[disk_type]`

デバイス名には、アンマウントされている raw デバイス (たとえば `/dev/rdisk/dsk0a`) を指定しなければなりません。

ファイル・システム固有のコマンド・オプションについての説明は、`newfs(8)` を参照してください。また、このリファレンス・ページでは、`mfs` コマンドについて説明し、メモリ・ファイル・システム (`mfs`) の作成方法についても説明しています。

新しいファイル・システムの作成例を次に示します。

1. ターゲット・ディスクとパーティションを確認します。ほとんどのシステムの場合、ローカルの管理ログを調べることで、システムに接続されているディスク・デバイスおよび割り当てられているパーティションが分かります。ただし、管理するシステムの状態が不明の場合もあります。つまり、デバイスが取り外されたり、追加されたりした場合です。次のコマンドとユーティリティを使用して、ターゲット・ディスクとパーティションを確認します。
 - a. `/dev/disk` ディレクトリの内容を調べます。検出されている各ディスク・デバイスには、パーティション・レイアウトに対応した一連のデバイス特殊ファイルがあります。たとえば、`/dev/disk/dsk1a ~ /dev/disk/dsk1h` は、`dsk1` という名前のデバイスが存在することを示します。
 - b. デバイスは使用可能な状態であっても、デバイス特殊ファイルがシステムに存在しないこともあります。次のように `hwmgr` コマンドを使用して、物理的にシステムのバスに接続されているすべてのデバイスを調べます。

```
# hwmgr -view devices -category disk
HWID:          DSF Name      Model      Location
-----
15:  /dev/disk/floppy0c      3.5in      fdi0-unit-0
17:  /dev/disk/dsk0c      RZ1DF-CB   bus-0-targ-0-lun-0
19:  /dev/disk/dsk0c      RZ1DF-CB   bus-0-targ-1-lun-0
19:  /dev/disk/cdrom0c      RRD47      bus-0-targ-4-lun-0
```

デバイス特殊ファイルがないデバイスが見つかった場合は、`dsfmgr` ユーティリティを使用して、デバイス特殊ファイルを作成することができます。

注意

通常、デバイス特殊ファイルは、新しいディスク・デバイスがシステムに追加されたときに自動的に作成されます。デバイス特殊ファイルを手動で作成しなければならないのは、『ハードウェア管理ガイド』で説明している状況の場合だけです。

- c. デバイスの確認ができれば、次のように `disklabel` コマンド使用して、使用されているパーティションを調べます。

```
# disklabel -r /dev/rdisk/dsk0a
```

```
8 partitions:
#      size  offset fstype [fsize bsize cpgh] #NOTE: values not
      exact
a:   262144      0 4.2BSD 1024 8192 16  # (Cyl.  0 -78*)
b:  1048576  262144  swap                # (Cyl. 78*-390*)
c:  17773524      0 unused      0    0      # (Cyl.  0 -5289*)
d:  1048576  1310720  swap                # (Cyl. 390*-702*)
e:  9664482  2359296  AdvFS              # (Cyl. 702*-3578*)
f:  5749746 12023778  unused      0    0      # (Cyl.3578*-5289*)
g:  1433600   524288  unused      0    0      # (Cyl. 156*-582*)
h: 15815636 1957888  unused      0    0      # (Cyl. 582*-5289*)
```

2. `disklabel` コマンドの出力で、使用されていないパーティションがいくつかあることが分かります。ただし、パーティション `c` は他のパーティションと重なっているため、使用することはできません。ディスクにカスタム・ディスク・ラベルが作成されていない限り、使用可能な標準パーティションのテーブルは、表 6-1 に示す 3 種類だけです。

表 6-1: ディスク・パーティション・テーブル

| パーティション・ テーブル | 説明 |
|------------------|---|
| c | ディスク全体が、シングル・パーティションとしてラベル付けされます。他のパーティションは <code>c</code> と重なり合うため、使用することはできません。 |
| a b g h | ディスクは 4 つのパーティションに分割されます。パーティション <code>a</code> は、ブート・パーティションとして使用できます。パーティション <code>c</code> , <code>d</code> , <code>e</code> , および <code>f</code> は重なり合うため使用できません。 |
| a b d e f | ディスクは 5 つのパーティションに分割されます。パーティション <code>a</code> は、ブート・パーティションとして使用できます。パーティション <code>c</code> , <code>g</code> , および <code>h</code> は重なり合うため使用できません。 |

ステップ 1.c の `disklabel` コマンドの出力にリストされているディスクでは、すでにパーティション `a`, `b`, `d`, および `e` が使用されています。このため、このディスクには 5 つのパーティションが定義されています。パーティション `f` は、未使用のため、新しいファイル・システム用に使用することができます。

注意

カスタム・ディスク・ラベルがディスクに書き込まれ、パーティションが拡張されている場合は、パーティションが未使用でも使用することはできません。この場合には、`newfs` コマンドはファイル・システムを作成できず、エラー・メッセージを表示します。

3. 次のように `newfs` コマンドを使用して、ターゲット・パーティションにファイル・システムを作成します。

```
# newfs /dev/rdisk/dsk0f
```

```
Warning: 2574 sector(s) in last cylinder unallocated
/dev/rdisk/dsk0f: 5749746 sectors in 1712 cylinders of \
20 tracks, 168 sectors
    2807.5MB in 107 cyl groups (16 c/g, 26.25MB/g, 6336 i/g)
super-block backups (for fsck -b #) at:
    32, 53968, 107904, 161840, 215776, 269712, 323648,
    377584, 431520, 485456, 539392, 593328, 647264, 701200,
    755136, 809072, 863008, 916944, 970880, 1024816, 1078752,
    1132688, 1186624, 1240560,
    .
    .
    .
```

コマンドの出力には、新しいファイル・システムのサイズが表示され、ファイル・システムのチェック・ユーティリティ `fsck` が使用するスーパーブロック・バックアップがリストされます。詳細については、リファレンス・ページの `fsck(8)` を参照してください。

4. 以降の項で説明するように、ファイル・システムをマウントします。

6.3.2 ファイル・システムをユーザからアクセス可能とする

`mount` コマンドを使用してファイル・システムをファイル・システム階層構造に接続すると、ファイル・システムが使用できるようになります。`mount` コマンドは、ファイル・システムをそのマウント・ポイントとなる既存のディレクトリに接続します。

注意

このオペレーティング・システムでは、4 K バイト・ブロック・サイズのファイル・システムはサポートしていません。ファイル・システムの省略時のブロック・サイズは、8 K バイトです。4 K バイト・ブロック・サイズのファイル・システムを含むディスク上のデータにアクセスするには、テープまたは 8 K バイト・ブロック・サイズのファイル・システムを持つディスクにデータのバックアップをとる必要があります。

システムのブート時には、`/etc/fstab` ファイルで定義されたファイル・システムがマウントされます。`/etc/fstab` ファイルには、ファイル・システムが存在するデバイスとパーティション、マウント・ポイント、ファイル・システム・タイプなどファイル・システムに関する情報を指定するエントリが記述されています。シングルユーザ・モードで使用している場合、ルート・ファイル・システムは読み取り専用モードでマウントされます。

ファイル・システムのマウント時に “dirty file system” エラーが表示された場合、そのファイル・システムに対して `fsck` コマンドを実行する必要があります。

ファイル・システムのマウント状態を変更するには、`-u` オプションを指定して `mount` コマンドを実行してください。このオプションは、リブートするときに `/etc/fstab` ファイルが利用できない場合に役立ちます。

リブートしようとして `/etc/fstab` ファイルが壊れている場合は、次のコマンドを使用します。

```
# mount -u /dev/disk/dsk0a /
```

`/dev/disk/dsk0a` デバイスはルート・ファイル・システムです。

6.3.3 `/etc/fstab` ファイルの使用

ルート・ファイル・システムには、AdvFS と UFS のどちらも使用できます。ただし、インストールの際に UFS を指定しないと、省略時のファイル・システムとして AdvFS が使用されます。出荷時にオペレーティング・システムがインストール済みのシステムでは、ルート・ファイル・システムは AdvFS です。オペレーティング・システムは、ルート・ファイル・システムを 1 つだけサポートし、そこから実行可能カーネル (`/vmunix`)

と、ブートや初期化に必要なその他のバイナリやファイルにアクセスします。ルート・ファイル・システムは、ブート時にマウントされ、アンマウントすることはできません。他のファイル・システムは、マウントしなければならず、`/etc/fstab` ファイルで、ブート・システムに対してどのファイル・システムをどこにマウントするかを指示します。

`/etc/fstab` ファイルにはファイル・システムについての情報が記述されており、`mount` コマンドなどのコマンドによって読み取られます。システムをブートすると `/etc/fstab` ファイルが読み取られ、そのファイルに記述されているファイル・システムがファイルに記述されている順番にマウントされます。各ファイル・システムの情報は 1 行で記述されており、各行の情報はタブまたはスペースで区切られています。

`mount` コマンドと `umount` コマンドは、`/etc/fstab` ファイルに記述されている順にファイル・エントリを読み取り実行するため、ファイル内のエントリの順番は重要です。

`/etc/fstab` ファイルを編集するには、`root` ユーザ特権が必要です。ファイルの変更後、すぐに変更を有効にしたい場合は、`mount -a` コマンドを実行します。このコマンドを実行しなかった場合、ファイルに対するすべての変更内容は、システムのリブート時に有効となります。

次に `/etc/fstab` ファイルの例を示します。

| | | | | | |
|-------------------------------|--------------------------------|--------------------|--------------------|----------------|----------------|
| <code>/dev/disk/dsk2a</code> | <code>/</code> | <code>ufs</code> | <code>rw</code> | <code>1</code> | <code>1</code> |
| <code>/dev/disk/dsk0g</code> | <code>/usr</code> | <code>ufs</code> | <code>rw</code> | <code>1</code> | <code>2</code> |
| <code>/dev/disk/dsk2g</code> | <code>/var</code> | <code>ufs</code> | <code>rw</code> | <code>1</code> | <code>2</code> |
| <code>/usr/man@tuscon</code> | <code>/usr/man</code> | <code>nfs</code> | <code>rw,bg</code> | <code>0</code> | <code>0</code> |
| <code>proj_dmn#testing</code> | <code>/projects/testing</code> | <code>advfs</code> | <code>rw</code> | <code>0</code> | <code>0</code> |
| | | | | | <code>6</code> |
| | | | | <code>5</code> | |
| | | | <code>4</code> | | |
| | | <code>3</code> | | | |
| <code>1</code> | <code>2</code> | | | | |

各行にはエントリが 1 つ含まれており、情報はタブまたはスペースで区切られています。`/etc/fstab` ファイルの各エントリは次の情報を表します。

- ① マウントするブロック型特殊デバイスまたはリモート・ファイル・システム。UFS の場合は、特殊ファイル名として文字型特殊ファイル名ではなく、ブロック型特殊ファイル名を指定します。AdvFS の場合には、特

殊ファイル名はファイル・ドメインの名前，番号記号 (#) およびファイルセット名の組み合わせです。

- ② ファイル・システムまたはリモート・ディレクトリのマウント・ポイント (たとえば `/usr/man`)，あるいは `/projects/testing`

- ③ ファイル・システムのタイプ

| | |
|---------------------|--|
| <code>cdfs</code> | ISO 9660 または HS 形式の CD-ROM ファイル・システム |
| <code>nfs</code> | NFS |
| <code>procfs</code> | デバッグに使用する <code>/proc</code> ファイル・システム |
| <code>ufs</code> | UFS ファイル・システムまたはスワップ・パーティション |
| <code>advfs</code> | AdvFS ファイル・システム |

- ④ そのパーティションに対するマウント・オプションを指定します。オプションは，コンマで区切って複数指定できます。通常，次のように，ファイル・システム・タイプに適したマウント・タイプおよび追加オプションを指定します。

`ro` ファイル・システムを読み取り専用アクセスでマウントします。

`rw` ファイル・システムを読み取り書き込みアクセスでマウントします。

`userquota`
`groupquota` ファイル・システムが `quotacheck` コマンドにより自動的に処理され，`quotaon` コマンドでファイル・システム・クォータが有効になるように指定します。

省略時の設定では，ファイル・システムのユーザおよびグループのクォータは `quota.user` ファイルと `quota.group` ファイルに含まれていますが，これらのファイルは，マウント・ポイントによって指定されたディレクトリにあります。たとえば，`/usr` にマウントされているファイル・システムのクォータ定義ファイルは `/usr` ディレクトリに置かれています。また，次に示す形式

で別のファイル名と位置を指定することもできます。 `userquota=/var/quotas/tmp.user`

XX

ファイル・システムのエントリを無視します。

- ⑤ バックアップする UFS ファイル・システムを決定するために `dump` コマンドが使用する値。値 1 を指定すると、ファイル・システムはバックアップされます。値を指定しないか、または 0 (ゼロ) を指定すると、ファイル・システムはバックアップされません。
- ⑥ これはパス番号で、`/etc/fstab` ファイル内のすべてのエントリを処理するときに、`fsck` (UFS) および `quotacheck` (UFS と AdvFS) ユーティリティの並列処理を制御するために使用します。このフィールドを使用して、スタートアップ時のファイル・システムの検査の順番を制御することで、同じ I/O サブシステムへ過度の I/O が集中してシステムが過負荷になるのを防ぎます。

パス番号を指定しないか、0 (ゼロ) を指定した場合は、ファイル・システムは検査されません。パス番号が 1 のエントリはすべて、一度に 1 つずつ処理されます (並列処理は行われません)。ルート・ファイル・システムには、必ず 1 を指定してください。パス番号が 2 以上のエントリは、割り当てられたパス番号に従って並列処理されます (例外もあります)。パス番号が 2 のエントリがすべて処理された後にパス番号 3 のエントリが処理され、パス番号 3 のエントリがすべて処理された後にパス番号 4 のエントリが処理されます。以降、同様に処理が続きます。例外は、同じディスク上の異なるパーティションに複数の UFS ファイル・システムがある場合や、同じドメイン内に複数の AdvFS ファイルセットがある場合です。これらのファイル・システムのパス番号がすべて同じ場合には、1 つずつ処理されます。同じパス番号を持つ他のファイル・システムは、すべて並列処理されます。

フィールドおよびオプションについての詳細は、`fstab(4)` を参照してください。

スワップ・パーティションは、次の例のように、`/etc/sysconfigtab` ファイルで構成されます。

```
swapdevice=/dev/disk/dsk0b,/dev/disk/dsk0d
vm-swap-eager=1
```

スワッピングとスワップ・パーティションの詳細については、『ハードウェア管理ガイド』，第 12 章，および `swapon(8)` を参照してください。

6.3.4 手動による UFS ファイル・システムのマウント

`mount` コマンドを使用してファイル・システムを使用できる状態に設定します。ファイル・システムを `/etc/fstab` ファイルに追加しない限り，マウントは一時的なものとなり，システムをリブートするとマウントが外されます。

`mount` コマンドは，UFS，AdvFS，NFS，CDFS，および `/proc` ファイル・システムをサポートしています。

すべてのファイル・システムに対して次の構文で `mount` コマンドを使用します。

```
mount [-adflruv] [-o option] [-t type] [file_system] [mount_point]
```

AdvFS では，ファイル・システム引数は `domain#fileset` の形式になります。引数として，ファイル・システムおよびマウント・ポイントを指定してください。マウント・ポイントとは，ファイル・システムをマウントしたいディレクトリです。このディレクトリは，すでにシステムに存在するものでなければなりません。リモート・ファイル・システムをマウントする場合は，ファイル・システムの指定には次のいずれかの構文を使用します。

```
host:remote_directory  
remote_directory@host
```

次のコマンドは，現在マウントされているファイル・システムとそのファイル・システム・オプションをリストします。

```
# mount -l  
/dev/disk/dsk2a on / type ufs (rw,exec,suid,dev,nosync,noquota)  
/dev/disk/dsk0g on /usr type ufs (rw,exec,suid,dev,nosync,noquota)  
/dev/disk/dsk2g on /var type ufs (rw,exec,suid,dev,nosync,noquota)  
/dev/disk/dsk3c on /usr/users type ufs (rw,exec,suid,dev,nosync,noquota)  
/usr/share/man@tuscon on /usr/share/man type nfs (rw,exec,suid,dev,  
nosync,noquota,hard,intr,ac,cto,noconn,wsiz=8192,rsiz=8192,  
timeo=10,retrans=10,acregmin=3,acregmax=60,acdirmin=30,acdirmax=60)  
proj_dmn#testing on /alpha_src type advfs (rw,exec,suid,dev,nosync,noquota)
```

次のコマンドは，ホスト `acton` の `/usr/homer` ファイル・システムを，ローカルのマウント・ポイント `/homer` に読み取り/書き込みモードでマウントします。

```
# mount -t nfs -o rw acton:/usr/homer /homer
```

一般的なオプションおよびファイル・システム・タイプに固有のオプションについての詳細は，`mount(8)` を参照してください。

6.3.5 手動による UFS ファイル・システムのアンマウント

ファイル・システムをアンマウントするには、`umount` コマンドを使用します。以下の状況では、ファイル・システムをアンマウントしなければなりません。

- `fsck` コマンドでファイル・システムをチェックする場合。
- `disklabel` コマンドでパーティションを変更する場合。ただし、パーティションを変更するとディスク上のファイル・システムを破壊する危険性があるので注意してください。

`umount` コマンドの構文は次のとおりです。

```
umount [-afv] [-h host] [-t type] [mount_point]
```

`cd` コマンドを含めて、なんらかのユーザ・プロセスがファイル・システムを使用している場合には、ファイル・システムをアンマウントすることはできません。このコマンドを起動したときにファイル・システムが使用中の場合は、次のエラー・メッセージが表示されます。

```
mount device busy
```

`umount` コマンドでルート・ファイル・システムをアンマウントすることはできません。

6.3.6 UFS ファイル・システムの拡張

UFS ファイル・システムの容量は、1 つのディスクまたは Logical Storage Manager (LSM) ボリュームで利用可能なストレージの量まで増やすことができます。UFS ファイル・システムの容量 (サイズ) を増やす処理のことを、ファイル・システムの拡張といいます。

ファイル・システムがオンライン (マウント状態) の場合は、`mount` のオプションで拡張できます。この操作は、ファイル・システムがオフライン (ディスマウント状態) の場合に `extendfs` コマンドを使用して行うこともできます。ファイル・システムが満杯であることをシステムが通知した場合、この手順を一時的ソリューションとして使用することも、恒久的なソリューションとして使用することもできます。ファイル・システムは、ストレージ・デバイスの物理的な限界に達するまで、必要に応じて何度でも拡張できます。

この操作を元に戻すことはできません。ファイル・システムを元のボリュームに戻すためには、`dump` またはバックアップ・ユーティリティを使用して

ファイル・システムをバックアップしてから、ファイル・システムを適切なサイズのディスク・パーティションへリストアする必要があります。

ファイル・システムを拡張するのに必要な条件は、次のとおりです。

ファイル・システムを確認する。

次のように `more` コマンドを使用して `/etc/fstab` ファイルを表示し、ファイル・システムと、ファイル・システムが存在するパーティションを確認します。

```
# /usr/bin/more /etc/fstab
/dev/disk/dsk0a on / type ufs (rw)
/proc on /proc type procfs (rw)
.
.
.
/dev/disk/dsk15e on /databases type ufs (rw)
/dev/disk/dsk4g on /projects type ufs (rw)
```

ファイル・システムのバックアップ・ステータスを確認する

この手順はファイル・システムを変更せず、必要な場合に素早く実行できるように設計されています。ただし、重要なデータ・ファイルはバックアップしてください。

利用可能なディスク・ストレージ容量を調べる

UFS ファイル・システムは、1 つのディスク・パーティションまたは LSM ボリューム上に存在します。ファイル・システムを拡張するには、次の方法でディスク・スペースを大きくします。

- LSM を使用していない場合、未使用の隣接パーティションのサイズを小さくして、ディスク・パーティションのサイズを大きくします。
- LSM を使用している場合、LSM コマンドを使用してボリュームを拡張します。詳細は、『*Logical Storage Manager*』を参照してください。この項の例では、LSM を使用していない UFS ファイル・システムについて説明します。

たとえば、現在使用中の次のようなディスクがあるとします。

- a パーティションが、500 MB の 3 次スワップ・パーティションとして使用されている。

- b パーティションが、ユーザ・ファイル専用の 2 GB の UFS ファイル・システムとして使用されている。
- g パーティションと h パーティションが未使用で、合計 6 GB のストレージ容量がある。

前述の例では、b パーティション上の UFS ファイル・システムは、未使用の 6 GB を使用して拡張することができます。1 回の拡張で 6 GB 全部を使用する必要はありません。拡張を段階的に行って、ディスク・スペースを節約することができます。

ファイル・システムを拡張するためのディスク容量が不足している場合は、ファイル・システムをバックアップし、新しいディスク・ボリュームとしてリストアします (第 9 章を参照)。

パーティション・サイズを再設定する

disklabel コマンドを使用して、ファイル・システムが置かれているパーティションのサイズを再設定します。次の手順は、disklabel コマンドの使用方法です。

注意

グラフィック・ディスク構成ユーティリティ diskconfig を使用して、この操作を実行することはできません。diskconfig ユーティリティはパーティションの検査を省略しないので、使用中のパーティションの再構成ができないためです。

1. パーティションを編集できるように、現在のディスク・ラベルをファイルに保存します。たとえば、次のコマンドを入力します。

```
# /sbin/disklabel -r /dev/disk/dsk4 > d4label
```
2. 保存したラベルを編集して、使用中のパーティションの容量を大きくし、未使用のパーティションの容量を同じだけ小さくします。

```
⋮
b: 10192000 1048576 4.2BSD 1024 8192 16 # (Cyl. 312*- 2750*)
g: 7104147 8669377 unused 1024 8192 # (Cyl. 2580*- 5289*)
```

たとえば、パーティション b のサイズを 3,000,000 ブロック大きくするには、ラベルを次のように変更します。

```
⋮
b: 13192000 1048576 4.2BSD 1024 8192 16 # (Cyl. 312*- 2750*)
g: 4104147 8669377 unused 1024 8192 # (Cyl. 2580*- 5289*)
```

ディスク・ラベルをファイルに保存して、エディタを終了します。

3. 編集したファイルを次のように指定して、raw ディスクにラベルを書き込みます。

```
# /sbin/disklabel -R /dev/rdisk/dsk4 d4label
```

詳細は、disklabel(8) を参照してください。

ディスク・スペースを追加すると、以降の項で説明している方法のいずれかでファイル・システムを拡張できます。

6.3.6.1 ディスマウント状態のファイル・システムの拡張

オフライン状態で使用中のファイル・システムを拡張するには、`extendfs` コマンドを使用します。1 回の操作でパーティション全体に拡張すること、段階的にファイル・システムを拡張することもできます。`extendfs` コマンドを、次のいずれかの方法で使用します。これらの手順は、ディスク・パーティション・サイズを大きくする準備作業 (6.3.6 項を参照) が完了していることを前提としています。

- ファイル・システムをパーティション全体に拡張するには、次のようなコマンドを使用します。

```
# /sbin/extendfs dsk4
Warning: 1324 sector(s) in last cylinder unallocated
/dev/rdisk/dsk4h: 9057236 sectors in 2696 cylinders of 20
tracks, 168 sectors
4422.5MB in 169 cyl groups (16 c/g, 26.25MB/g, 6336 i/g)
super-block backups (for fsck -b #) at:
32, 53968, 107904, 161840, 215776, 269712, 323648, 377584,
⋮
```

このコマンドの出力は、新しい UFS ファイル・システムの作成に使用される、`/sbin/newfs` コマンドの出力に似ています。詳細は、`newfs(8)` を参照してください。

- 利用できるパーティション・スペースの一部だけを使用してファイル・システムを拡張するには、次のように `-s` オプションを使用します。

```
# /sbin/extendfs -s 500000
```

この例では、利用可能なパーティション・スペースのうちの 500,000 ブロックだけを使用し、残りは将来の拡張用にとっておきます。前述の例のように、`/sbin/extendfs` コマンドの出力は、`/sbin/newfs` コマンドの出力に似ています。

6.3.6.2 マウント状態のファイル・システムの拡張

オンライン (マウント状態) で使用中のファイル・システムを拡張するには、`mount` コマンドを使用します。この手順は、マウント状態のパーティションを確認し、ディスク・パーティション・サイズを大きくする準備作業 (6.3.6 項を参照) が完了していることを前提としています。

`mount` コマンドの構文は、6.3.4 項で説明されています。ファイル・システムを拡張するには、`-o` オプションとともに `extend` オプションを使用し、マウント・ポイントを次のように指定します。

```
# /sbin/mount -u -o extend /projects
extending file system, please wait.
```

`mount` コマンドは、この操作の完了やエラーは出力しません。完了するまでの時間はパーティションのサイズによって異なり、場合によっては数分かかります。このため、`df` コマンドを次のように使用して、操作の結果を確認しなければなりません。

```
# /usr/bin/df /projects
```

このコマンドの出力により、操作が成功したかどうかを確認できます。操作が成功しなかった場合は、ディスク・パーティションが正しいかどうかや、マウント・ポイントが存在しているかを確認します。

6.4 SysMan Menu による UFS ファイル・システムの管理

このオペレーティング・システムには、手動によるファイル・システムの作成と管理の他に、グラフィカル・ツールや、さまざまなユーザ環境で利用できる SysMan Menu タスクが用意されています。SysMan の起動と使用については、第 1 章を参照してください。共通デスクトップ環境 (CDE) を使用している場合は、他のグラフィカル・ユーティリティが使用できます。こ

これらのユーティリティは、CDE アプリケーション・マネージャのメイン・フォルダから次のようにしてアクセスします。

1. CDE フロント・パネルでアプリケーション・マネージャのアイコンを選択します。
2. 「アプリケーション・マネージャ」フォルダ・ウィンドウで「システム管理」アイコンを選択します。
3. 「アプリケーション・マネージャ-システム管理」フォルダ・ウィンドウで「ストレージ管理」アイコンを選択します。

システムにインストールされライセンスされているオプションに従って、このウィンドウでは以下のアイコンが使用できます。

Advanced File System

AdvFS グラフィカル・ユーザ・インタフェースを起動するには、このアイコンを選択します。詳細については、『*AdvFS 管理ガイド*』を参照してください。

コマンド行で AdvFS グラフィカル・インタフェースを起動する方法については、`dtadvfs(8)` を参照してください。

ブート・テープ

SysMan Menu のブート可能テープ作成インタフェースを起動するには、このアイコンを選択します。このインタフェースを使用して、ブート可能システム・イメージをテープ上に作成します。このイメージには、スタンドアロン・カーネルと、作成時に選択したファイル・システムのコピーが含まれます。`btextract` ユーティリティを使用すると、このイメージを復元することができます。ブート可能テープ・インタフェースの使用については、第 9 章を参照してください。詳細は、`btcreate(8)`、`btextract(8)`、および `bttape(8)` を参照してください。`bttape` コマンドを使用すると、コマンド行またはスクリプトからブート可能

| | |
|-------------------------------|---|
| | <p>テーブル・グラフィカル・ユーザ・インタフェースを起動できます。</p> |
| ファイル・システム管理 | <p>この項で説明している SysMan Storage ユーティリティを起動するには、このアイコンを選択します。</p> |
| Logical Storage Manager (LSM) | <p>LSM グラフィカル・ユーザ・インタフェースを起動するには、このアイコンを選択します。LSM を使用すると、システムやアプリケーションから単一のデバイスとして見える仮想ディスク・ボリュームを作成することができます。詳細については、『<i>Logical Storage Manager</i>』を参照してください。LSM コマンドの一覧については、<code>lsm(8)</code> を参照してください。</p> <p>コマンド行からこのインタフェースを起動するには、<code>lsmsa</code> コマンドまたは <code>dxlsm</code> コマンドを使用します。詳細については、<code>lsmsa(8)</code> を参照してください。<code>dxlsm</code> は、オペレーティング・システムの今後のリリースでは、削除される予定です。</p> |
| Prestoserve I/O Accelerator | <p>Prestoserve グラフィカル・インタフェースを起動するには、このアイコンを選択します。Prestoserve は、ディスクへの同期書き込みをディスクではなく、不揮発性メモリに書き込みます。格納されたデータは、必要に応じて非同期にディスクに書き込まれるか、マシンが停止するときディスクに書き込まれます。詳細については、『<i>Guide to Prestoserve</i>』を参照してください。コマンド行インタフェースについては、<code>presto(8)</code> を参照してください。</p> <p>コマンド行からこのインタフェースを起動するには、<code>dxpresto</code> コマンドを使用し</p> |

ます。詳細については、`dxpresto(8)` を参照してください。

以降の項では、SysMan Menu にある UFS ファイル・システム・ユーティリティについて説明します。

6.4.1 SysMan Menu でのファイル・システムの作業

SysMan Menu には、[ストレージ] という名前のメイン・メニュー・オプションがあります。このオプションは、次のとおりです。

- ストレージ [storage]
 - ファイルシステム管理ユーティリティ [filesystems]
 - 一般的なファイルシステム・ユーティリティ [generalfs]
 - | ファイルシステムのアンマウント [dismount]
 - | 現在マウントされているファイルシステムの表示 [df]
 - | ファイルシステムのマウント [mount]
 - | ローカル・ディレクトリ(/etc/exports)の共有 [export]
 - | ネットワーク・ディレクトリ(/etc/fstab)のマウント [net_mount]
 - Advanced File System (AdvFS) ユーティリティ [advfs]
 - | AdvFS ドメインの管理 [domain_manager]
 - | AdvFS ファイルの管理 [file_manager]
 - | AdvFS ドメインのデフラグメント [defrag]
 - | 新しい AdvFS ドメインの作成 [mkfdmn]
 - | 新しい AdvFS ファイル・セットの作成 [mkfset]
 - | AdvFS ドメインのファイルの修復 [salvage]
 - | AdvFS ドメインの修復 [verify]
 - UNIX File System (UFS) ユーティリティ [ufs]
 - | 新しい UFS ファイルシステムの作成 [newfs]
 - Logical Storage Manager (LSM) の管理 [lsm]
 - | Logical Storage Manager (LSM) の初期化 [volsetup]
 - | Logical Storage Manager (LSM) 管理 [lsmmgr]

各オプションには、基本的なファイル・システムの管理作業を 1 ステップずつ実行するインタフェースがあります。SysMan Menu の起動と使用方法については、第 1 章を参照してください。SysMan Station からファイル・システム・ユーティリティを起動することもできます。たとえば、SysMan Station を使用して「Mounted_Fileystems」ビューを表示している場合は、MB3 を押して次の作業を行うことができます。

- マウントされているファイル・システムをアンマウントする
「Dismount」などの、利用可能な [Storage] オプションを実行します。
- マウント・ポイントや使用されているスペースなど、ファイル・システムのプロパティを表示します。

SysMan Station の「Physical_Filesystems」ビューでは、物理デバイスにマップされているファイル・システムのグラフィカル・ビューを表示し、既存のドメインで AdvFS ファイルセットの作成などの作業を行うことができます。SysMan Station の起動と使用方法については、第 1 章を参照してください。ファイル・システム・オプションの使用については、オンライン・ヘルプを参照してください。

次の SysMan Menu の [ストレージ] オプションについては、他のマニュアルで説明します。

Advanced File System (AdvFS) Utilities

『*AdvFS 管理ガイド*』を参照してください。

Logical Storage Manager (LSM) ユーティリティ

『*Logical Storage Manager*』を参照してください。

以降の項では、SysMan Menu から使用できる [一般的なファイルシステム・ユーティリティ] と [UNIX File System (UFS) ユーティリティ] のファイル・システム作業について説明します。SysMan Menu のタスクの構成は同じ順序にはなっていませんが、ファイル・システムの一般的な作成手順は、6.3 節に説明されている手順とまったく同じです。これらのタスクは、ファイル・システムの作成と管理を行うためにいつでも使用できる、汎用のユーティリティです。

6.4.2 SysMan によるファイル・システムのディスマウント

ファイル・システムをディスマウントするには、そのマウント・ポイント、デバイス特殊ファイル名、または AdvFS ドメイン名を指定しなければなりません。これらの情報は、`more` コマンドを使用して `/etc/fstab` ファイルの内容を表示したり、6.4.3 項に説明されている SysMan Menu の [ストレージ] オプションの [現在マウントされているファイルシステムの表示] を使用して取得することができます。コマンド行のオプションについては、`mount(8)` と `umount(8)` を参照してください。

[ファイルシステムのアンマウント] オプションは、SysMan Menu の [ストレージ] オプションから使用できます。このオプションが表示されていない場合は、メニューを展開して [一般的なファイルシステム・ユーティリティ] を選択します。[ファイルシステムのアンマウント] オプションを選択すると「ファイルシステムのアンマウント」ウィンドウが表示され、次

のフィールドのどちらかを入力するように求められます。両方のフィールドを入力する必要はありません。

1. マウント・ポイント

/mnt など、ファイル・システムが現在マウントされているマウント・ポイントを入力します。

2. ファイル・システム名

マウントされたパーティションのデバイス特殊ファイル名 (たとえば、/dev/disk/dsk0f) か、AdvFS ドメイン名 (たとえば、accounting_domain#act) を入力します。

[適用] ボタンを選択してファイル・システムをディスマウントした後、他のファイル・システムのディスマウント処理を続けるか、[了解] ボタンを選択してファイル・システムをディスマウントして終了します。

6.4.3 SysMan によるマウントされているファイル・システムの表示

マウントされているファイル・システムを表示するオプションが、SysMan Menu の [ストレージ] オプションで使用できます。メニューを展開して、[一般的なファイルシステム・ユーティリティ - 現在マウントされているファイルシステムの表示] を選択してください。このオプションを選択すると、次のようなファイル・システムのリストを含む、「現在マウントされているファイルシステムの表示」ウィンドウが表示されます。

```
/dev/disk/dsk0a      /  
/proc                /proc  
usr_domain#usr       /usr  
usr_domain#var        /var  
19serv:/share/19serv/tools/tools /tmp_mnt/19serv/tools
```

このウィンドウに表示される情報は次のとおりです。

ファイル・システム 次のいずれかのファイル・システムです。

- マウントされているデバイス・パーティションに対応する、/dev/* ディレクトリのデバイス特殊ファイル名。パス名 /dev/disk/dsk0a は、ディスク 0 のパーティション a を示します。デバイス名とデバイス特殊ファイルにつ

いては、『ハードウェア管理ガイド』を参照してください。

- NFS (Network File System) マウントされたファイル・シェア。このファイル・シェアは、`automount` ユーティリティまたは `autofs` ユーティリティによってマウントされている可能性があります。これらのユーティリティは、エクスポートされているネットワーク・ファイル・システムをローカル・ユーザがアクセス (インポート) したときに、自動的にマウントします。NFS、`automount`、および `autofs` については、『ネットワーク管理ガイド：サービス編』を参照してください。NFS マウントは一般的に、エクスポートしているホスト・システム名とエクスポート・ディレクトリの順で、次のようにリストされます。

```
19serv:/share/19serv/tools/tools /tmp_mnt/19serv/tools
```

`19serv:` は、ホスト名識別子とそれに続くコロン (:) です。

`/share/19serv/tools/tools` は、エクスポートされているディレクトリのパス名です。

`/tmp_mnt/19serv/tools` は、NFS によって自動的に作成される一時的なマウント・ポイントです。

- `usr_domain#var` などの、AdvFS ドメイン名。ドメインについては、『*AdvFS* 管理ガイド』または `advfs(4)` を参照してください。
- `file-on-file mount` などの分かりやすい名前。この名前は、`/usr/net/servers/lanman/.ctrlpipe` などのサービス・マウント・ポイントを示します。

マウント・ポイント `/usr` や `/accounting_files` などの、ファイル・システムがマウントされているディレクトリ。

現在マウントされているファイル・システムの数によっては、大規模なリストになることがあります。このリストでは、`/etc/fstab` ファイルにリストされていない、現在の file-on-file マウント情報を知ることができます。このリストでは、`/etc/fstab` ファイルに指定されているファイル・システムでも、現在マウントされていないものは表示されません。

「現在マウントしているファイルシステム」ウィンドウでは、次のオプションが使用できます。

[詳細...] ファイル・システム・データの詳細 (ファイル・システムのプロパティ) を表示するには、このオプションを使用します。このオプションで表示されるデータは次のとおりです。

```
File system name: /dev/disk/dsk0a
Mount point:      /
File system size: 132 MBytes
Space used:       82 MBytes
Space available:  35 MBytes
Space used %:     70%
```

[アンマウント...] 選択したファイル・システムをディスマウントするには、このオプションを使用します。ディスマウント要求を確認するプロンプトが表示されます。ファイル・システムが現在使用中の場合や、ユーザが `cd` コマンドを使用して、ディスマウントするファイル・システムへディレクトリを変更している場合でも、そのファイル・システムをディスマウントすることはできません。ユーザにファイル・システムの使用をやめるように連絡するには、`wall` コマンドを使用します。

[更新] 「現在マウントしているファイルシステム」リストを再表示して、ディスマウントされたファイル・システムを反映するには、このオプションを使用します。

コマンド行でファイル・システムをマウントした場合や、NFS マウントした場合には、ユーティリティを一度終了してから再度起動するまで、新たにマウントされたファイル・システムは表示されません。

[終了]

「現在マウントしているファイルシステム」ウィンドウを終了して SysMan Menu に戻るには、このボタンを押します。

6.4.4 SysMan によるファイル・システムのマウント

ファイル・システムのマウント操作には、次の前提条件があります。

- ファイル・システムは、`/etc/fstab` ファイルにリストされていなければなりません。
- マウント・ポイントが存在しなければなりません。存在しない場合は、`mkdir` コマンドを使用してマウント・ポイントを作成します。詳細は、`mkdir(1)` を参照してください。
- ファイル・システムがディスク・パーティション上に作成され、そのディスクがオンラインでなければなりません。SysMan Menu による UNIX ファイル・システム (UFS) の作成については、6.4.7 項を参照してください。`newfs` コマンドによるファイル・システムの手動作成については、6.3.1 項を参照してください。詳細は、`newfs(8)` を参照してください。AdvFS ファイル・システムの作成については、『*AdvFS 管理ガイド*』を参照してください。

`diskconfig` グラフィカル・ユーティリティを使用すると、ディスク・パーティションのカスタマイズとパーティション上のファイル・システムへの書き込みを一度に行うことができます。`diskconfig` コマンドについては、『*ハードウェア管理ガイド*』を参照してください。このユーティリティの起動については、`diskconfig(8)` を参照してください。またこのユーティリティは、SysMan Menu、SysMan Station、または CDE アプリケーション・マネージャからも起動できます。

- デバイスがシステムによって認識されている必要があります。

通常、ディスク・デバイスが使用可能どうかはシステムによって自動的に管理されます。ただし、システムの実行中にデバイスを動的に追加した直後は、システムはそのデバイスをまだ認識していないことがあ

ります。この場合は、そのデバイスを見つけてオンラインにするように、システムに要求しなければなりません。

この処理を行い、必要に応じて既存のディスクのディスク・デバイスとパーティションの状態をチェックするには、`hwmgr` コマンドを使用します。詳細は、`hwmgr(8)` を参照してください。デバイスの管理については、『ハードウェア管理ガイド』を参照してください。

通常、ディスク・パーティションのデバイス特殊ファイル (たとえば、`/dev/disk/dsk5g`) は、システムが自動的に作成し管理します。ただし、デバイス特殊ファイルがない場合は、作成する必要があります。

`dsfmgr` コマンドについては、『ハードウェア管理ガイド』を参照してください。各デバイス (Dev ノード) のデバイス特殊ファイルをリストする `dsfmgr -s` などのコマンド・オプションについては、`dsfmgr(8)` を参照してください。

ファイル・システムをマウントするオプションは、SysMan Menu の [ストレージ] オプションで使用できます。メニューを展開して [一般的なファイルシステム・ユーティリティ - ファイルシステムのマウント] を選択し、「マウント操作」ウィンドウを表示します。このインタフェースは、`mount(8)` に説明されている `mount` コマンドの代わりになるものです。このユーティリティの操作対象は、`/etc/fstab` ファイルに現在リストされているファイル・システムだけです。マウントされているファイル・システムについての情報は、6.4.3 項に説明されている [現在マウントされているファイルシステムの表示] SysMan Menu オプションを使用して取得できます。

「マウント操作」ウィンドウには、次の 4 つの排他的な選択可能オプションがあります。

1. 指定したファイル・システムのマウント

特定のファイル・システムを 1 つマウントするには、このオプションを選択します。「ファイル・システム名とマウント・ポイント」ウィンドウが表示され、次のどちらかのフィールドへ入力を行うように要求されます。

マウント・ポイント `/etc/fstab` ファイルに指定されているマウント・ポイント・ディレクトリ (たとえば、`/cdrom`) を入力します。

ファイル・システム名 /dev/disk/cdrom0c などのデバイス特殊ファイル名を入力します。または、usr_domain#usr などの、AdvFS ドメイン名を入力します。

次に、「ファイル・システムのマウント・オプション」ウィンドウが表示されます。このウィンドウはさまざまなマウント操作に共通なので、このリストの最後で説明します。

2. /etc/fstab にリストされているすべてのファイル・システムのマウント

/etc/fstab ファイルに現在リストされているすべてのファイル・システムをマウントするには、このオプションを使用します。このオプションでは、指定されたすべてのパーティションまたはドメインがオンラインで、すべてのマウント・ポイントが作成されていることを前提としています。

次に、「ファイル・システムのマウント・オプション」ウィンドウが表示されます。このウィンドウはさまざまなマウント操作に共通なので、このリストの最後で説明します。

3. 指定したタイプのファイル・システムだけのマウント

/etc/fstab ファイルにリストされているファイル・システムのうち、指定したタイプをすべてマウントするには、このオプションを使用します。このオプションでは、指定されたすべてのパーティションまたはドメインがオンラインで、すべてのマウント・ポイントが作成されていることを前提としています。

次に表示される「ファイル・システムのマウント・オプション」ウィンドウで、ファイル・システムのタイプを指定します。このウィンドウはさまざまなマウント操作に共通なので、このリストの最後で説明します。たとえば、AdvFS ファイル・システムだけを選択することができます。

4. 選択したタイプ以外のすべてのファイル・システムのマウント

/etc/fstab ファイルにリストされているファイル・システムのうち、指定したタイプを除いてマウントするには、このオプションを使用します。このオプションでは、指定されたすべてのパーティションまたはドメインがオンラインで、すべてのマウント・ポイントが作成されていることを前提としています。

次に表示される「ファイル・システムのマウント・オプション」ウィンドウで、除外するファイル・システム・タイプを指定します。このウィンドウはさまざまなマウント操作に共通なので、このリストの最後で説明します。たとえば、UFS ファイル・システムだけを除外するように選択することができます。

「ファイル・システムのマウント・オプション」ウィンドウは、前述したさまざまなマウント操作に共通で、マウント操作の操作特性を追加指定することができます。実行するマウント操作のタイプによっては、使用できないオプションもあります。このウィンドウで利用できるオプションは、次のとおりです。

| | | |
|---------------|----------------------------|---|
| アクセス・モード | 使用可能にしたいアクセス・タイプを選択します。 | |
| | 読み/書き | 権限のあるユーザがファイル・システム内のファイルから読み取ったり、ファイルを書き込んだりできるようにするには、このオプションを選択します。 |
| | 読み取り専用 | 権限のあるユーザがファイル・システムからファイルの読み取りだけを行えるようにしたり、CD-ROM ボリュームなどの読み取り専用メディアをマウントするには、このオプションを選択します。 |
| ファイル・システム・タイプ | メニューから、次のいずれかのオプションを選択します。 | |
| | 指定なし | すべてのファイル・システムを指定するには、このオプションを選択します。 |
| | AdvFS | Advanced File System タイプを指定するには、このオプションを選択します。詳細 |

については、『*AdvFS* 管理ガイド』または `advfs(4)` を参照してください。

UFS

UNIX ファイル・システム・タイプを指定するには、このオプションを選択します。このファイル・システムの説明については、6.1.4 項を参照してください。

NFS

ネットワーク・ファイル・システムを指定するには、このオプションを選択します。詳細については、『ネットワーク管理ガイド：サービス編』と、`nfs(4)` を参照してください。

CDFS

コンパクト・ディスク読み取り専用ファイル・システムを指定するには、このオプションを選択します。詳細については、`cdfs(4)` を参照してください。

その他

次に説明する「その他のファイル・システム・タイプ:」フィールドに独自のファイル・システムを入力するには、このオプションを選択します。

その他のファイル・システム・タイプ

メモリ・ファイル・システム (RAM ディスク) の `mfs` など、指定するファイル・システムを入力します。サポートされているファイル・システムについて

の詳細は、`mount(8)` を参照してください。また、メモリ・ファイル・システムの `newfs(8)` など、個々のファイル・システムのリファレンス・ページも参照してください。

拡張マウント・オプション ファイル・システムに対する拡張マウント・オプションを入力します。たとえば、システム・クラッシュなどでファイル・システムが正常にディスマウントされなかった場合でもそのファイル・システムをマウントできるようにする、`dirty` オプションを入力できます。各種のオプションについての詳細は、`mount(8)` を参照してください。

オプションを指定したら [完了] ボタンを選択して、マウント操作を行い SysMan Menu オプションに戻ります。「マウント操作」ウィンドウに戻って新しいマウント操作を行うには、[戻る] ボタンを使用します。マウント操作を打ち切るには、[終了] ボタンを使用します。

フィールドのデータが不完全か誤っている場合は、マウント操作の開始前に、データを修正するように求められます。

6.4.5 SysMan によるローカル・ディレクトリの共用

ファイルの共用を行う場合は、ファイル・システムを `/etc/exports` ファイルに追加します。この追加により、他のホスト・システムのユーザは、NFS (ネットワーク・ファイル・システム) を介して共用ディレクトリをマウントすることができます。ASU (Advanced Server for UNIX) をインストールして実行している場合は、PC クライアントとファイル・システムを共用することもできます。『*Advanced Server for UNIX* コンセプトとプランニング・ガイド』を参照してください。

また、リモート・ホストが共用ディレクトリをマウントできるように、ご使用のシステムへのネットワーク・アクセスを使用可能にしなければならないこともあります。たとえば、ホストを `/etc/hosts` ファイルに追加して、NFS を設定し、`dxhosts` を実行します。共用ファイル・システムへの接続を許可するようにシステムを構成する方法については、『ネットワーク管理ガイド：サービス編』を参照してください。

dxfileshare グラフィカル・ユーザ・インタフェースを使用して、共用ファイル・システムを管理することもできます。このインタフェースは、コマンド行または CDE の「アプリケーション・マネージャ - 日常管理」フォルダから起動できます。このフォルダ内の「ファイル共有」オプションを参照してください。このインタフェースには、オンライン・ヘルプがあります。このインタフェースの起動についての詳細は、dxfileshare(8) を参照してください。

共用ファイル・システムの唯一の前提条件は、共用に適したディスク・ファイル・システムがすでに作成されていることです。作成方法は、6.3.1 項 (手動) または 6.4.7 項 (SysMan Menu オプションの使用) のとおりです。共用ファイル・システムは、ディレクトリのパス名 (たとえば、/usr/users/share) で指定します。

ファイル・システム共用オプションは、次の SysMan Menu [ストレージ] ブランチで使用できます。

- ストレージ [storage]
 - ファイルシステム管理ユーティリティ [filesystems]
 - 一般的なファイルシステム・ユーティリティ [generalfs]
 - | ローカル・ディレクトリ (/etc/exports) の共有 [export]

6.4.5.1 ファイル・システムの共用

既存のファイル・システムを共用するには、次の手順に従います。

1. 「ローカル・ディレクトリの共有: *hostname.xxx.yyy.xxx*」ウィンドウの最初のボックスに、既存の共用がディレクトリ・パス名で表示されます。
[追加...] ボタンを押して、ディレクトリをリストに追加します。「ローカル・ディレクトリの共有: ローカル・ディレクトリの追加」ウィンドウが開きます。
2. 「共有するディレクトリ:」フィールドに、/usr/users/share/tools などのディレクトリ・パス名を入力します。
3. ディレクトリを読み取り/書き込みアクセスで共用するか、読み取り専用アクセスで共用するかを選択します。省略時には、「読み書き両用」が選択されます。
4. 適切なすべてのホスト (リモート・システム) についてディレクトリを共用するか、次のように指定されたホストについてだけディレクトリを共用するかを選択します。

[すべて] すべてのホストの場合には、これを選択します。

[選択] 選択したホストだけの場合には、これを選択します。

5. 前の手順で [選択] を選択した場合は、ホスト名とアドレス (たとえば、`dplhst.xxx.yyy.com`) を入力します。このホストは、`/etc/hosts` ファイルまたはドメイン・ネーム・サーバ (DNS) によってローカル・ホストが認識できなければなりません。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。
6. [了解] を選択して、データを有効にしダイアログ・ボックスをクローズします。そして、「ローカル・ディレクトリの共有: *host name*」ウィンドウに戻ります。

すべての変更は、このウィンドウで [了解] を選択しない限り適用されません。

[了解] を選択すると、そのディレクトリの共有が可能になります。

6.4.5.2 共有ファイル・システムの削除

共有を外すには、次のように同じユーティリティを使用します。

1. アクセス・リストからホストを削除します。
2. 読み取り/書き込み許可を変更して共有ファイル・システムへのアクセスを変更するか、選択したホストをアクセス・リストから削除します。
3. すべてのアクセスを防止するには、共有リストから共有ファイル・システムを削除します。

6.4.6 SysMan によるネットワーク・ファイル・システムのマウント

ネットワーク・ファイル・システム (NFS) を使用すると、他のホストと共有 (エクスポート) される共有ファイル・システムをマウントすることができます。リモート・ホストへのネットワーク・アクセスを設定するなど、ローカル・システム (ホスト) を構成して NFS 共有ファイル・システムをインポートしなければなりません。リモート・システム (ホスト) を構成して、`/etc/exports` ファイルに自分のシステムを指定しファイル・システムを共有またはエクスポートしなければなりません。次のいくつかの方法で、NFS 共有ファイル・システムをマウントできます。

リブートすると解除される、一時的なマウント。

マウント・ポイントが作成され、現在のセッションの間だけファイル・システムが接続されます。何らかの理由でシステムがシャットダウンされると、マウント・ポイントはそのまま残りますが、ファイル・システムの接続は失われます。この接続は、システムのリブート時には再接続されません。

ローカル・システムの `/etc/fstab` ファイルに共用 NFS ファイル・システムを指定することによる、永続的なマウント。

たとえば、`/etc/fstab` ファイルには、次のような NFS ファイル・システムのエントリがすでにいくつか定義されている場合があります。

```
/usr/lib/toolbox@ntsv /usr/lib/toolbox nfs rw,bg,soft,nosuid 0 0
```

`/etc/fstab` ファイルの構造については、6.3.3 項を参照してください。

NFS automount ユーティリティによる、ユーザの要求に応じた自動マウント。

このオプションの使用については、『ネットワーク管理ガイド：サービス編』と、`automount(8)` を参照してください。`automount` を使用すると、ローカル・ユーザはローカル・システムにエクスポートされ共用されるファイル・システムを、透過的にマウントすることができます。ユーザからのマウント要求に対して絶えず応答する必要はありません。

この項では、NFS 共用を `/etc/fstab` ファイルに永続的に追加したり、共用ファイル・システムを一時的にインポートする方法を説明します。

SysMan Menu オプションを使用した、ファイル・システムの共用 (エクスポート) の処理については、6.4.5 項を参照してください。

「ファイル共有」グラフィカル・ユーザ・インタフェース (GUI) を使用して、共用ファイル・システムを管理することができます。このインタフェースは、コマンド行、または CDE の「アプリケーション・マネージャ - 日常管理」フォルダから起動することができます。このフォルダ内の「ファイル共有」オプションを参照してください。このインタフェースには、オンライン・ヘルプがあります。このインタフェースの起動については、`dxfileshare(8)` を参照してください。

6.4.6.1 共用ネットワーク・ファイル・システムのマウント

NFS ファイル・システムのマウント・オプションが SysMan Menu [ストレージ] オプションで使用できます。メニューを展開して、[一般的なファイルシステム・ユーティリティ - ネットワーク・ディレクトリ(/etc/fstab)のマウント]を選択します。共用ファイル・システムをマウントするには、次の手順に従ってください。

1. 「ネットワーク・ディレクトリのマウント: *hostname*」ウィンドウに、*/etc/fstab* ファイルにリストされている使用可能な既存の NFS 共用ファイル・システムが表示されます。このリストには次の情報があります。

| | | |
|------------|--|---|
| ディレクトリとホスト | ホストの名前と、ローカル・システムにエクスポートされるディレクトリの名前。 | |
| マウント先 | 共用ファイル・システムがマウントされるローカル・マウント・ポイント。このマウント・ポイントは、 <i>/tools/bin/imaging</i> のような、ディレクトリのパス名です。 | |
| オプション | ディレクトリのアクセス・オプション。アクセス・オプションには、次のものがあります。 | |
| | 読み書き両用 | 共用ファイル・システム上のデータの読み書きをユーザに許可します。ただし、共用ファイル・システムをエクスポートしているホストが設定したアクセス条件に依存します。 |
| | 読み取り専用 | 共用ファイル・システム上のデータの読み取りだけをユーザに許可します。 |

リブート

システムが何らかの理由でシャットダウンされた後に再度マウントが行われるかどうかを、次のように示します。

`true`

永続的なマウント。エントリがローカル・システムの `/etc/fstab` ファイルに存在し、リブート後もマウントされます。

`false`

一時的なマウント。ローカル・システムの `/etc/fstab` ファイルにエントリが存在せず、リブート後はマウントされません。

6.4.6.2 ネットワーク・ディレクトリの追加

ファイル・システムを NFS 共用ディレクトリのリストに追加するには、[追加...] ボタンを選択します。「ネットワーク・ディレクトリのマウント: ネットワーク・ディレクトリの追加」ウィンドウが表示されます。

このオプションを使用すると、ファイル・システムが、省略時のオプション `hard` (応答を受信するまでリトライ) と `bg` (バックグラウンドでのマウント) でマウントされます。これらのオプションについての詳細は、`mount(8)` を参照してください。

6.4.7 SysMan による UFS ファイル・システムの作成

`newfs` コマンドによる UFS ファイル・システムの手動作成については、6.3.1 項で説明しています。同じ前提条件とデータ・ソースが、SysMan Menu オプションによるファイル・システムの作成時にも適用されます。ただし、SysMan Menu オプションでは、標準ディスク・パーティションのみに制限されます。カスタム・パーティションを使用する場合は、『ハードウェア管理ガイド』に説明されている `diskconfig` ユーティリティを使用してください。

まず、次のデータ項目を取得します。

- ファイル・システムが格納される場所について。次のいずれかの方法で指定します。
 - ファイル・システムを作成するディスク・パーティションのデバイス特殊ファイル名。たとえば、ディスク 13 のパーティション h の場合には、`/dev/disk/dsk13h` です。
 - LSM アプリケーションを使用している場合は、LSM ボリューム名。詳細については、『*Logical Storage Manager*』を参照してください。
- RZ1DF-CB などの、ディスク・モデル。この情報は、次のように `hwmgr` コマンドを使用して取得できます。

```
# hwmgr -view devices
```

または、SysMan Station Hardware View を使用します。ディスクを選択し、MB3 を押して、ポップアップ・メニューで [Properties...] を選択すると、そのデバイスの詳細が表示されます。ディスク・モデルの情報ソースは、`/etc/disktab` ファイルです。`/etc/disktab` ファイルの構造については、`disktab(4)` を参照してください。

- ファイル・システムに、ブロック・サイズや最適化などの特定のオプションが必要かどうか判断します。オプションの完全なリストについては、`newfs(8)` を参照してください。SysMan Menu ユーティリティでオプションを表示することができます。

新しい UFS ファイル・システムを作成するオプションは、SysMan Menu の [ストレージ] オプションから使用できます。メニューを開いて、[UNIX File System (UFS) ユーティリティ – 新しい UFS ファイルシステムの作成] を選択します。次に、「新しい UFS ファイルシステムの作成」ウィンドウが表示されます。

ファイル・システムを作成するには、次の手順に従います。

1. 「パーティションか LSM ボリューム」フィールドに、ファイル・システムを格納するディスク・パーティションまたは LSM ボリュームの名前を入力します。
2. 「ディスク・タイプ」フィールドに、HB00931B93 などの、ディスク・モデルの名前を入力します。

3. 「拡張 `newfs` オプション」フィールドに、64 K バイトのブロック・サイズを示す `-b 64` など、オプション・フラグを入力します。

使用するオプションが不確かな場合は、すべてのフィールドをクリアしてから [適用] を選択してください。これにより、フラグ・オプションのリストを含む、`newfs` 情報ウィンドウが表示されます。

[了解] を選択して、ファイル・システムを作成し SysMan Menu を終了するか、[適用] を選択して、ファイル・システムを作成し他のファイル・システムの作成を続けます。操作を終了する場合は、[取消] を選択します。

新しく作成したファイル・システムをマウントするには、6.4.4 項で説明した SysMan Menu オプションの [ファイルシステムのマウント] を使用します。

6.5 クォータの管理

以降の項では、UFS のユーザ・クォータとグループ・クォータについて説明します。また、AdvFS も、ファイルセットのスペース使用量を制限するクォータをサポートしています。AdvFS ファイルセットのクォータについては、『*AdvFS 管理ガイド*』を参照してください。このドキュメントには、ユーザ・クォータおよびグループ・クォータについての、AdvFS 固有の情報も説明されています。

システム管理者は、ファイル・システムに対してクォータを設定することにより、ユーザ・アカウントおよびグループの単位で使用量の制限を設けることができます。このため、ユーザ・クォータおよびグループ・クォータは、ファイル・システム・クォータと呼ばれます。また、ファイル・システム・クォータは、ユーザ・アカウントとユーザ・グループの単位で、使用するディスク・ブロック数を制限することから、ディスク・クォータと呼ばれます。

ユーザ・アカウントおよびグループへのクォータの設定は、ファイル・システムごとに行います。たとえば、1 つのユーザ・アカウントがファイル・システム上の複数のグループのメンバになり、別のファイル・システム上の別のグループのメンバにもなることができます。この場合、このユーザ・アカウントに対するファイル・システム制限は、そのファイル・システム上にある、このユーザ・アカウントのファイルに対する制限となります。ユーザ・アカウントのクォータが超過するのは、そのファイル・システム上で使用しているブロック (または i ノード) 数が超過した場合です。

ユーザ・アカウントのクォータと同様、グループのクォータが超過するのは、特定のファイル・システム上で使用されているブロック (または i ノード) 数が超過した場合です。ただし、グループのブロックまたは i ノードを使用して、グループのクォータとしてカウントされるのは、作成されたファイルにグループの ID (GID) が割り当てられている場合だけです。グループのメンバが書き込みを行っても、グループの GID が割り当てられていないファイルは、グループのクォータとしてはカウントされません。

注意

クォータ・コマンドでのブロック・サイズは、一般的な 512 バイトではなく、1024 バイトで表示されます。

ユーザ・アカウントまたはユーザ・グループに割り当て可能なブロック数および i ノード数 (またはファイル数) の上限を設定するために、ファイル・システムにクォータを適用することができます。各ファイル・システム上で、ユーザまたはユーザ・グループごとに、異なるクォータ値を設定することができます。/usr/users などのホーム・ディレクトリを持つファイル・システムは他のファイル・システムと比較して使用量が増大する可能性が高いため、これらのファイル・システムにクォータを設定することが考えられます。/tmp ファイル・システムにクォータを設定することは避けてください。

6.5.1 物理クォータ制限と論理クォータ制限

ファイル・システム・クォータには、論理 (ソフト) クォータ制限と物理 (ハード) クォータ制限を設定することができます。物理制限に達した場合、制限を超えるようなディスク・スペースの割り当てまたはファイルの作成は許可されません。物理制限とは、クォータ制限が有効な場合に許可される値を 1 単位 (ブロック、ファイル、i ノードなど) 超えた値です。

クォータは、上限値まで (上限値は含まない) を指します。たとえば、ファイル・システムの各ユーザ・アカウントに対して 10,000 ディスク・ブロックの物理制限を設定した場合、9,999 ディスク・ブロックが割り当てられると、アカウントは物理制限に達します。ユーザ・アカウントに対して最大 10,000 ブロックを完全に割り当てたい場合は、物理制限として 10,001 を設定しなければなりません。

論理制限ではある一定の間 (猶予期間) 上限を超えることができます。猶予期間を過ぎても論理制限を越えている場合、十分なディスク・スペースを解放するかまたはファイルを削除して、論理制限に設定されたディスク・スペース使用量またはファイル数を下回るまで、ディスク・スペースの割り当てまたはファイルの作成は許可されません。

システム管理者は、ユーザが現在の作業を終了し、ファイルを削除して設定されている制限をクォータが下回るようにするための十分な猶予期間を設定しなければなりません。

注意

ファイルの書き込み中に、物理制限および論理制限の両方のクォータ制限に到達した場合は、ファイルが部分的に書き込まれることがあります。この結果、ファイルを別の場所に保存するか、処理を停止しない限り、データが失われます。

たとえば、ファイルの編集中にクォータ制限を超えた場合は、データが失われる可能性があるため、エディタを打ち切ったり、ファイルへの書き込みを行ったりしないでください。この場合、使用中のエディタから一度抜け出し、ファイルを削除してからエディタに戻ります。または別のファイル・システムにファイルを書き込んだ後、クォータに達したファイル・システムから不要なファイルを削除し、ファイルをもとのファイル・システムに移動して戻すことができます。

6.5.2 ファイル・システム・クォータの設定

UFS にファイル・システム・クォータを設定するには、次の手順を実行します。

1. システム構成にファイル・システム・クォータ・サブシステムが組み込まれるようにシステム構成ファイルを編集します。 `/sys/conf/NAME` システム構成ファイルに次の行を追加します。

```
options          QUOTA
```

2. `/etc/fstab` ファイルを編集してファイル・システムのエントリの 4 番目のフィールドを変更し、`rw` および `userquota` か `groupquota` を設定します。詳細については、`fstab(4)` を参照してください。

3. `quotacheck` コマンドを使用して、クォータ・ファイルを作成します。
クォータ・サブシステムは、現在の割り当てとクォータの上限値をクォータ・ファイルに格納します。 `quotacheck` コマンドについての詳細は、`quotacheck(8)` を参照してください。
4. `edquota` コマンドを使用してクォータ・エディタを起動し、各ユーザに対するクォータ・エントリを作成します。
`edquota` は、指定した各ユーザまたは各グループごとにテキスト・エディタで編集できる一時的な ASCII ファイルを作成します。このファイルを編集して、クォータを設定する各ファイル・システムのエントリ、および論理制限、物理制限、および猶予期間を記述します。
2 つ以上のユーザ名またはグループ名を `edquota` コマンド行に指定した場合、編集内容は各ユーザまたはグループに対して反映されます。またプロトタイプを使用すると、6.5.3 項に説明されているように、ユーザ・グループに対するクォータを迅速に設定することもできます。
5. `quotaon` コマンドを使用して、クォータ・システムを起動します。
設定したクォータの有効化についての詳細は、`quotaon(8)` を参照してください。
6. システム・スタートアップ時にファイル・システム・クォータを確認して有効にするには、次のコマンドを使用して `/etc/rc.config` ファイルのファイル・システム・クォータ構成変数を設定します。

```
# /usr/sbin/rcmgr set QUOTA_CONFIG yes
```

注意

`QUOTQ_CONFIG` に `yes` を設定すると、スタートアップ時に `quotacheck` コマンドが UFS ファイル・システムに対して実行されます。AdvFS では、このサービスは必要ありません。次のコマンドを使用すると、スタートアップ時に `quotacheck` コマンドを、UFS と AdvFS ファイル・システムの両方に対して強制的に実行することができます。ただし、この使用方法はお勧めしません。

```
# /usr/sbin/rcmgr set QUOTACHECK_CONFIG -a
```

省略時の、UFS だけの `quotacheck` 動作に戻すには、次のコマンドを使用します。

```
# /usr/sbin/rcmgr set QUOTACHECK_CONFIG ""
```

クォータ制限を解除する場合は、`quotaoff` コマンドを使用します。また、`umount` コマンドは、クォータ制限を解除してからファイル・システムをアンマウントします。詳しい説明については、`quotaoff(8)` を参照してください。

6.5.3 ユーザ・アカウントへのファイル・システム・クォータの設定

ファイル・システム・クォータをユーザに設定するには、クォータ・プロトタイプを作成するか、既存のクォータ・プロトタイプをそのユーザ用にコピーします。クォータ・プロトタイプは、既存のユーザ・クォータをプロトタイプ・ファイルにしたもので、他のユーザに対して同一のユーザ・クォータを生成する場合に使用できます。プロトタイプの作成には、`edquota` コマンドを使用します。クォータ・プロトタイプがない場合は、次の手順で作成してください。

1. `root` でログインし、次の構文で `edquota` コマンドを使用します。

```
edquota proto-user users
```

たとえば、ユーザ `eddie` に対して `large` という名前のクォータ・プロトタイプを設定するには、次のコマンドを入力します。

```
# edquota large eddie
```

これによって、ユーザ `eddie` に対して `large` クォータ・プロトタイプが作成されます。`users` 引数には、実際のログイン名を使用しなければなりません。

2. `edquota` プログラムでオープンされたクォータ・ファイルを編集して、ユーザ `eddie` からアクセスできる各ファイル・システムにクォータを設定します。

既存のクォータ・プロトタイプをユーザ用に使用するには、次の手順に従います。

1. `edquota` コマンドを次の構文で入力します。

```
edquota -p proto-user users
```

たとえば、large プロトタイプを使用してユーザ marcy に対するファイル・システム・クォータを設定するには、次のコマンドを入力します。

```
# edquota -p large marcy
```

2. ユーザ marcy に対して、期待したとおりにクォータが設定されているか確認します。

期待したとおりでない場合は、クォータ・ファイルを編集して、ユーザ marcy からアクセスできる各ファイル・システムに新しいクォータを設定します。

6.5.4 ファイル・システム・クォータの確認

ユーザ・ファイル・システム・クォータを設定している場合、クォータ・システムを定期的に確認する必要があります。quotacheck、quota、および repquota コマンドを使用して、設定された上限を実際の使用量と比較することができます。

quotacheck コマンドは、実際のブロック使用量が、設定された制限に合っているかどうかを確認します。quotacheck コマンドは、ファイル・システムでクォータが最初に有効になったとき (UFS と AdvFS) と、各リブート後 (UFS のみ) の 2 回実行してください。このコマンドは、システムが処理をまったく行っていない場合に、より正確な情報を示します。

quota コマンドは、ファイル・システムの各ユーザごとに、実際のブロック使用量を表示します。quota コマンドは、root ユーザのみが実行できます。

repquota コマンドは、指定されたファイル・システムに関する実際のディスク使用量とクォータを表示します。ユーザごとに、現在のファイル数とディスク使用量 (K バイト単位) をクォータとともに表示します。

設定済みのクォータを変更する場合は、edquota コマンドを使用します。このコマンドは各ユーザの上限の設定や変更を行うものです。

ファイル・システム・クォータについての詳細は、quotacheck(8)、quota(1)、および repquota(8) を参照してください。

6.6 ファイル・システムのバックアップとリストア

AdvFS および UFS の両方のファイル・システム用の主なバックアップおよびリストア・ユーティリティは、vdump ユーティリティと vrestore ユーティリティです。これらのユーティリティは、AdvFS ファイル・システ

ムと UFS ファイル・システムのローカル操作で使用されます。これらのユーティリティについては、`vdump(8)` と `vrestore(8)` を参照してください。AdvFS および UFS の両方のファイル・システム用のリモート・バックアップ操作とリモート・リストア操作を行うユーティリティは、`rvdump` と `rvrestore` です。

UFS ファイル・システムだけをバックアップする管理者は、`dump(8)` と `restore(8)` で、従来のユーティリティについて参照してください。

AdvFS のバックアップおよびリストア操作の例は、『*AdvFS 管理ガイド*』で説明しています。UFS のバックアップおよびリストア操作の例は、第 9 章で説明しています。この章では、ブート可能テープの作成処理についても説明しています。このテープは、厳密な意味ではバックアップではありませんが、ルート・ファイル・システムと重要なシステム・ファイルのブート可能テープのコピーを作成することができます。ルート・ディスクの破壊などの障害時に、このテープからシステムをブートして、システムを回復することができます。

この他にも、第 9 章で説明されている Networker Save and Restore というアーカイブ・サービスがあります。

6.7 ファイル・システムのモニタリングとチューニング

以降の項では、UFS ファイル・システムの情報表示および検査に使用するコマンドについて説明します。また、ファイル・システムのチューニングの基本的な情報についても説明します。チューニングについての詳細は、『システムの構成とチューニング』を参照してください。

6.7.1 UFS の一貫性の検査

`fsck` プログラムは、UFS ファイル・システムを検査して、ディスクへのファイルの保存が確実に行えるように修復を行います。`fsck` プログラムは、参照されていない i ノード、空きリスト内の存在しないブロック、スーパーブロック内のカウンターの誤りなどの、ファイル・システムの矛盾を修復できます。

ファイル・システムは、不適切なシャットダウン手順、ハードウェア障害、停電、電源のサージなど、いろいろな状況で壊れることがあります。また、マウントされているファイル・システムに物理的に書き込み保護をかけた後、マウントされているファイル・システムをオフラインにしたり、システ

ムをシャットダウンする前に `sync` コマンドを使用しなかった場合にも、ファイル・システムが壊れることがあります。

システムはブート時に `fsck` を非対話形式で実行し、安全に行える修復については自動的に実行します。システムが予期していなかった矛盾を発見すると `fsck` プログラムは終了し、システムはシングルユーザ・モードになります。この際、`fsck` プログラムを手動で実行するよう警告メッセージが表示されます。手動で実行すると、`fsck` はプロンプトを表示するため、ユーザは `yes` または `no` で応えることができます。

`fsck` プログラムを呼び出すコマンドの構文は、次のとおりです。

```
/usr/sbin/fsck [ options ...] [ file_system ...]
```

ファイル・システムの指定を省略すると、`/etc/fstab` ファイル内の全ファイル・システムが検査されます。ファイル・システムを指定する場合は、必ず `raw` デバイスを使用します。

コマンド・オプションについての説明は、`fsck(8)` を参照してください。

注意

ルート・ファイル・システムを検査するには、シングルユーザ・モードでなければなりません。また、ファイル・システムは読み取り専用でマウントされていなければなりません。システムをシャットダウンしてシングルユーザ・モードにするには、第2章に説明されている `shutdown` コマンドを使用します。

6.7.2 ファイル・システムのディスク使用量のモニタリング

適切な量のディスクの空きスペースを確保するには、次のいずれかの方法で、構成されているファイル・システムのディスク使用量を定期的にモニタリングする必要があります。

- `df` コマンドを使用して、使用可能な空きスペースを確認する。
- `du` コマンドまたは `quot` コマンドを使用して、ディスクの使用量を確認する。
- ファイル・システム・クォータを設定している場合は、`quota` コマンドを使用してファイル・システム・クォータを確認する。

quota コマンドは、root ユーザのみ使用できます。

6.7.2.1 使用可能な空きスペースの確認

構成されているファイル・システムに十分なスペースを確保するには、df コマンドを定期的に使用して、マウントされているすべてのファイル・システムのディスクの空きスペースの容量を確認しなければなりません。df コマンドは、指定されたファイル・システムの空きディスク・スペース、または指定されたファイルを含むファイル・システムの空きディスク・スペースについての統計値を表示します。

df コマンドの構文は次のとおりです。

```
df [- eiknPt] [- F fstype] [file] [file_system ...]
```

df コマンドの引数やオプションを省略すると、マウントされているすべてのファイル・システムの空きディスク・スペースが表示されます。df コマンドは、各ファイル・システムのサイズを 512 バイトのブロック単位で表示します。ただし、-k オプションを指定すると、サイズは K バイトのブロック単位で表示されます。このコマンドは、スペースの総容量、現在使用中の容量、現在使用可能な (未使用の) 容量、使用率、およびファイル・システムがマウントされているディレクトリを表示します。

AdvFS ファイル・ドメインに対しては、df は各ファイルセットのディスク使用量を表示します。

ファイル・システムがまったくマウントされていないデバイスを指定した場合、df はルート・ファイル・システムの情報を表示します。

ファイル・パス名を指定すると、そのファイルが含まれるファイル・システムの使用可能なディスク・スペース量を表示することができます。

df コマンドにブロック型特殊デバイス名または文字型特殊デバイス名を指定して、アンマウントされたファイル・システムの空きスペースを表示することはできません。この場合には、dumpefs コマンドを使用してください。

詳細については、df(1) を参照してください。

次の例では、マウントされているすべてのファイル・システムに関するディスク・スペース情報を表示しています。

```
# /sbin/df
Filesystem            512-blks   used  avail capacity Mounted on
/dev/disk/dsk2a        30686   21438   6178     77%   /
```

| | | | | | |
|---------------------|--------|--------|--------|-----|----------------|
| /dev/disk/dsk0g | 549328 | 378778 | 115616 | 76% | /usr |
| /dev/disk/dsk2 | 101372 | 5376 | 85858 | 5% | /var |
| /dev/disk/dsk3 | 394796 | 12 | 355304 | 0% | /usr/users |
| /usr/share/man@tsts | 557614 | 449234 | 52620 | 89% | /usr/share/man |
| domain#usr | 838432 | 680320 | 158112 | 81% | /usr |

注意

`newfs` コマンドは、割り当てやブロック・レイアウト用に、一定の割合のファイル・システム・ディスク・スペースを予約します。このため `df` コマンドでは、ファイル・システムの使用量が容量の 100 % を超えた値で報告されることがあります。 `tunefs` コマンドで `-minfree` フラグを指定することにより、この割合を変更することができます。

6.7.2.2 ディスク使用量の確認

ファイル・システムに使用可能スペースが十分でない場合は、`du` コマンドまたは `quot` コマンドを使用して、スペースがどのように使用されているかを確認します。

`du` コマンドは、ディスク・スペース割り当てを、ディレクトリごとに正確に表示します。この情報によって、ディスク・スペースを最も多く使用しているユーザや、ディスク・スペースを解放すべきユーザを判断することができます。

`du` コマンドの構文は次のとおりです。

```
/usr/bin/du [- aklrsx] [ directory... filename...]
```

`du` コマンドは、指定されたディレクトリ、指定されたファイル名、または現在の作業ディレクトリ (何も指定されなかった場合) 内の全ディレクトリに含まれるブロックの数を再帰的にリストして表示します。システムが使用するクラスタ・サイズにかかわらず、ブロックは、各ファイルの間接ブロックを 1 K バイト単位でカウントします。

オプションを省略すると、各ディレクトリに対するエントリのみが生成されます。コマンド・オプションについての詳細は、リファレンス・ページの `du(1)` を参照してください。

次の例では、`/usr/users` ディレクトリの各サブディレクトリで使用しているブロック数を表示しています。

```
# /usr/bin/du -s /usr/users/*
440    /usr/users/barnam
43     /usr/users/broland
747    /usr/users/frome
6804   /usr/users/norse
11183  /usr/users/rubin
2274   /usr/users/somer
```

この情報から、ユーザ `rubin` が最も多くのディスク・スペースを使用していることが分かります。

次の例では、`/usr/users/rubin/online` ディレクトリ内の各ファイルおよびサブディレクトリで使用しているスペースを表示しています。

```
# /usr/bin/du -a /usr/users/rubin/online
1      /usr/users/rubin/online/inof/license
2      /usr/users/rubin/online/inof
7      /usr/users/rubin/online/TOC_ft1
16     /usr/users/rubin/online/build
.
.
.
251    /usr/users/rubin/online
```

`du` コマンドの代わりに、`ls -s` コマンドを使用してファイルのサイズと使用量を取得することもできます。使用量を取得する際には、`ls -l` コマンドは使用しないでください。`ls -l` コマンドでは、ファイル・サイズだけが表示されます。

`quot` コマンドを使用すると、指定したファイル・システム内で現在各ユーザが所有しているブロック数をリストすることができます。`quot` コマンドの実行には、`root` ユーザ特権が必要です。

`quot` コマンドの構文は次のとおりです。

`/usr/sbin/quot [-c] [-f] [-n] [file_system]`

次の例では、`/dev/disk/dsk0h` ファイル・システム内で各ユーザが使用しているブロック数と、各ユーザが所有しているファイル数を表示します。

```
# /usr/sbin/quot -f /dev/disk/dsk0h
```

デバイスをマウントするとブロック型デバイス特殊ファイルはビジーになるため、UFS ファイルの情報を表示するためには文字型デバイス特殊ファイルを使用しなければなりません。

詳細は、`quot(8)` を参照してください。

6.7.3 UFS 読み取り効率の改善

UFS の読み取り効率を向上するには、`tunefs` コマンドを使用して、レイアウト・ポリシーに影響する、ファイル・システムの動的パラメータを変更します。

`tunefs` コマンドの構文は次のとおりです。

```
tunefs [-a maxc] [-d rotd] [-e maxb] [-m minf] [-o opt] [file_s]
```

`tunefs` コマンドは、マウントされたファイル・システムとアンマウントされたファイル・システムの両方に使用できます。ただし、変更は、アンマウントされたファイル・システムに対してコマンドを使用した場合のみ可能です。ルート・ファイル・システムを指定した場合は、変更内容を有効にするにはリブートしなければなりません。

コマンド・オプションを使用すると、ディスク・パーティション・レイアウト・ポリシーに影響する動的パラメータを指定することができます。コマンド・オプションについての詳細は、`tunefs(8)` を参照してください。UFS サブシステム属性についての詳細は、`sys_attrs_ufs(5)` を参照してください。

6.8 ファイル・システムのトラブルシューティング

次のツールを使用すると、UFS ファイル・システムに関連する問題の解決に役立ちます。

- UNIX シェル・オプションの使用

UNIX シェル・オプションは、経験豊かな管理者のためのインストレーション・オプションです。このオプションは、オペレーティング・システムのテキスト・インストレーションまたはグラフィカル・インストレーションのどちらかでも使用できます。たとえば、このオプションを使用すると、壊れたルート・ファイル・システムの回復を行うことができます。

このインストレーション・オプションの概要については、『インストレーション・ガイド』を参照してください。このオプションで実行できる、ファイル・システム関連の管理については、『インストレーショ

ン・ガイド — 上級ユーザ編』を参照してください。このインストール・オプションは、AdvFS と UFS の両ファイル・システムの問題に使用できます。

- /usr/field ディレクトリと fsx コマンドの使用

/usr/field ディレクトリには、オペレーティング・システムのフィールド管理に関連するプログラムが置かれています。このディレクトリ内のプログラムを使用して、オペレーティング・システムの構成要素およびシステム・ハードウェアをモニタしたり操作することができます。

fsx ユーティリティはファイル・システムを操作します。このユーティリティについては、fsx(8) を参照してください。テープ・エクササイザ (tapex) およびディスク・エクササイザ (diskx) などの、このディレクトリ内の他のプログラムは、ファイル・システムの問題を調査するときに役立ちます。

- UFS ファイル・システムの情報を表示するには、dumpfs ユーティリティを使用します。詳細については、dumpfs(8) を参照してください。
- ファイル・システムの問題に関連したイベントをフィルタ処理して表示するには、EVM (イベント・マネージャ) を使用します。このユーティリティは、ファイル・システムおよびストレージ・デバイスの予防保守の設定とモニタに役立ちます。詳細については、第 13 章を参照してください。
- ファイル・システムのグラフィカル・ビューを表示し、ディスク・スペース不足などのファイル・システムの問題をモニタリングしたり、トラブルシューティングを行うには、SysMan Station と Insight Manager を使用します。詳細については、第 1 章を参照してください。



ユーザ・アカウントおよびグループの管理

ユーザ・アカウントの割り当てとグループ化は、システム・リソースをユーザに提供する最も一般的な方法です。この章では、ユーザ・アカウントおよびグループの管理について以下の項目を説明します。

- アカウントおよびグループの管理に使用するユーティリティと、それらを使用するユーザ環境 (7.1 節)
- ユーティリティの概要と、作業を進めるためのオンライン・ヘルプの使用法 (7.2 節)
- アカウントおよびグループの一般的な概念 (LDAP や NIS) を理解するための情報、アカウントおよびグループに割り当てられる一意の識別子などの重要なデータ項目、パスワードおよびグループのシステム・データ・ファイルの内容、アカウントやグループの省略時の特性の設定方法 (7.3 節)
- ユーザ・アカウントおよび関連するシステム・リソースの追加、変更および削除など、それぞれのユーティリティを使用した、ユーザ・アカウントの管理手順 (7.4 節)
- それぞれのユーティリティを使用したユーザ・グループの管理手順 (7.5 節)
- Windows NT ドメイン・アカウントと UNIX アカウントの関連付け (同期化) (7.6 節)

7.1 アカウント管理のオプションと制限

ローカル・システムの構成、ユーザ環境およびユーザの好みに応じて、ユーザ・アカウントの管理に使用できるいくつかの方法とさまざまなユーティリティがあります。以降の各項では、管理方法の選択肢を示し、これらを使用する場合の制限および必要条件について説明します。

7.1.1 管理ユーティリティ

オペレーティング・システムには、アカウントを管理するさまざまなユーティリティがあります。この章では、これらのユーティリティすべてについて詳しく説明しませんが、その使用に関する基本方針はすべてのユーティリティにおいて同じです。各ユーティリティのオプションについての詳細は、個々のオンライン・ヘルプとリファレンス・ページを参照してください。

ユーティリティの一覧は、表 7-1 にリストされています。これらのユーティリティを使用するには、UNIX システムの場合 root ユーザ、Windows NT の場合ドメイン管理者でなければなりません。

表 7-1: アカウントおよびグループの管理ユーティリティ

| ユーティリティ | ユーザ環境の説明 |
|--|---|
| SysMan Menu の「アカウント」オプション 「ローカル・ユーザとグループの管理」 「NIS ユーザとグループの管理」 「LDAP ユーザとグループの管理」 | さまざまなユーザ環境で SysMan Menu を使用できます (第 1 章を参照)。このユーティリティでは、アカウントとグループの追加および削除など、限られた管理機能が使用できます。ASU (Advanced Server for UNIX) をインストールしてある場合は、このユーティリティを使用して、UNIX アカウントおよびグループの省略時の特性を管理することはできません。関連する (同期化された) Windows NT ドメイン・アカウントの作成または削除を選択することはできません。アカウントの省略時の構成 (useradd または usermod による) によって、これらの選択は自動的に行われます。 SysMan Menu の「アカウント」オプションのフィルタ (検索) 機能を使用すると、大量のユーザ・アカウントを管理する方法をカスタマイズできます。 |
| アカウント・マネージャ (dxaccounts) | これは、UNIX および Windows NT ドメインの両方のアカウントに関しての、ほとんどのユーザおよびグループの管理オプションを提供するグラフィカル・ユーザ・インタフェースです。これは、SysMan Menu ツールではなく、X11 ベースのツールです。CDE (省略時の UNIX 環境) は、X11 に準拠しています。 |

表 7-1: アカウントおよびグループの管理ユーティリティ (続き)

| ユーティリティ | ユーザ環境の説明 |
|--|--|
| useradd usermod userdel | これらのコマンドは、文字セル環境の UNIX システムで実行するコマンド行ツールであり、すべてのユーザの管理作業ができます。これらのコマンドを使用して、UNIX および関連する (同期化された) Windows NT ドメインの両方のアカウントを管理することができます。また、アカウントの省略時の環境を構成することもできます。 |
| groupadd groupmod groupdel | これらのコマンドは、文字セル環境の UNIX システムで実行するコマンド行ツールであり、すべてのユーザ・グループの管理作業ができます。これらのコマンドを使用して、UNIX グループの省略時の環境を構成することができます。 |
| ASU ドメイン・ユーザ・マネージャ User Manager for Domains | Microsoft Windows NT ベースの PC 用アプリケーション。このユーティリティを使用して、Windows NT ドメインのアカウントを管理することができます。このユーティリティと他の ASU ユーティリティを使用して、ポリシー管理オプションを使ってアカウントの省略時の特性を設定することができます。UNIX アカウントの省略時の環境は構成できません。 |
| ASU の net コマンド | UNIX システムの端末または Windows NT サーバを実行しているシステム上の DOS プロンプトで入力できるコマンド。これらのコマンドの動作は、ASU User Manager for Domains ユーティリティと同じです。 |

Microsoft Windows ベースのユーティリティを使用するには、ASU (Advanced Server for UNIX) をインストールして構成する必要があります。この章では、ASU ユーティリティの使用の詳細は説明していません。ASU ユーティリティについては、UNIX サーバ上での ASU ソフトウェアの実行についてのみ説明しています。ASU のインストレーションとその使用については、『*Advanced Server for UNIX* インストレーション/管理ガイド』を参照してください。

7.1.2 ユーティリティの使用についての説明と制限

以下の制限は、アカウント管理ユーティリティを使用する際に、または特定のシステム機能を使用可能にする際に適用されます。

- UNIX アカウントの省略時の特性 (プロパティ) の構成

省略時の UNIX アカウントおよびグループの省略時の特性を構成する際には、UNIX コマンド行ユーティリティまたはアカウント・マネージャ `dxaccounts` だけを使用できます。

ASU を使用している場合の PC アカウントの省略時の設定については、『*Advanced Server for UNIX* インストレーション/管理ガイド』を参照してください。

- エンハンスト (C2) セキュリティ

エンハンスト・セキュリティ機能が使用可能な場合は、アカウントの作成が制限され、次のような追加機能が使用できます。

1. エンハンスト・パスワード制御
2. アカウントの使用可能化および使用不能化 (またはロック) のオプション
3. アカウントの削除および廃止のオプション

詳細については、『セキュリティ管理ガイド』を参照してください。

- ネットワーク情報サービス (NIS)

NIS を使用すると、ユーザはローカル・ネットワーク内で NIS を実行しているすべてのシステムにログインすることができます。アカウント名およびパスワードなどのユーザ・データは、すべての NIS を実行しているシステム間で共用されます。この場合ユーザは、`passwd` コマンドではなく、`yppasswd` などの別のコマンドを使用してパスワードを変更します。

NIS を構成した場合は、2 つのユーザのクラスを管理することになります。

ローカル・ユーザとグループ
NIS ユーザとグループ

ユーザ・アカウント管理ユーティリティの NIS をサポートする機能は、NIS が実行されている場合だけ使用できます。NIS 環境の設定については、『ネットワーク管理ガイド：サービス編』を参照してください。

- LDAP (Lightweight Directory Access Protocol)

LDAP は、概念的には、NIS に似ています。認証対象に対し、1 つのレポジトリ、LDAP データベースを持ちます。

LDAP を使用すると、ユーザはローカル・ネットワーク内で LDAP を実行しているすべてのシステムにログインすることができます。アカウント名およびパスワードなどのユーザ・データは、すべての LDAP を実行しているシステム間で共有されます。この場合ユーザは、異なるコマンドを使用します。たとえば、passwd コマンドではなく、yppasswd を使用してパスワードを変更します。

LDAP を構成した場合は、3 つのユーザのクラスを管理することになります。

- ローカル・ユーザとグループ
- NIS ユーザとグループ
- LDAP ユーザとグループ

ユーザ・アカウント管理ユーティリティの LDAP をサポートする機能は、LDAP が実行されている場合だけ使用できます。

LDAP の詳細は、www.OpenLDAP.org にある OpenLDAP のドキュメントを参照してください。

- アカウント管理ユーティリティの同時起動

アカウント管理ユーティリティは、起動されるとロック・ファイルを作成して、他のアカウント管理ユーティリティ (または、同じユーティリティの 2 つのインスタンス) が `/etc/passwd` などのファイル・システムにアクセスしないようにします。このロック・ファイルは、`/etc/.AM_is_running` にあります。ロック・ファイルの作成により、システム・ファイルのアカウント・データの破壊を防止できます。場合によっては、このロック・ファイルが正しくクリアされず、手作業で削除しなければならないことがあります。ロック・ファイルを削除するときには、`ps -ef` コマンドを実行して、`AM_is_running` のインスタンスを検査し、アカウント管理ツールの実行中のインスタンスに関連していないことを確認してください。

LDAP をサポートしているのは、コマンド行ユーティリティと SysMan Menu ツールだけです。アカウント管理ユーティリティ (dxaccounts) ではサポートしていません。

SysMan Menu の「アカウント」オプションは、遅延完了を使用するように設計されています。つまり、入力したデータは保持され、ユーザが確認を行うまでファイルに書き込まれません。このため、別のインスタンスのアカウント管理ユーティリティが動作している最中に SysMan Menu

の「アカウント」オプションを起動することはできますが、[適用] または [了解] を選択してシステム・ファイルをアップデートすることはできません。別のインスタンスのアカウント管理ユーティリティがクローズされると、ロック・ファイルが削除されるため、処理を完了できます。

DOP (Division of Privileges) と分散管理機能を使用して、root ユーザはアカウント管理の特権を一般のユーザに容易に割り当てることができます。ただし、同時には 1 人の特権を持ったユーザが 1 つのアカウント管理ユーティリティだけを使用できます。

7.1.3 関連ドキュメント

アカウント管理に関するその他のドキュメントには、マニュアル、リファレンス・ページ、およびオンライン・ヘルプがあります。

7.1.3.1 マニュアル

Tru64 UNIX オペレーティング・システムのドキュメント・セットの中で、アカウント管理について説明しているドキュメントは次のとおりです。

- 第 6 章 では、ファイル・システムとユーザのファイル・スペースについて説明しています。
- 『ネットワーク管理ガイド：サービス編』では、NIS ユーザ・アカウントについて説明しています。
- 『セキュリティ管理ガイド』では、リソースをユーザに割り当てるときのセキュリティ上の重要な考慮事項について説明しています。また、エンハンスド・セキュリティ機能およびシステム監査のためのアカウントの必要条件についても説明しています。
- 『*Common Desktop Environment: 上級ユーザ及びシステム管理者ガイド*』では、CDE 環境の構成とプリンタなどの省略時のシステム・リソースの設定について説明しています。
- ASU ドキュメンテーション・キットには、『*Advanced Server for UNIX* コンセプトとプランニング・ガイド』、『*Advanced Server for UNIX* インストレーション/管理ガイド』、および『*Advanced Server for UNIX* リリース・ノート』があります。

7.1.3.2 リファレンス・ページ

リファレンス・ページには、各種コマンドがサポートするすべてのオプションとスイッチの完全なリストが記載されています。この章で参照するリファレンス・ページは次のとおりです。

- コマンド行ユーティリティは、`useradd(8)`、`usermod(8)`、`userdel(8)`、`groupadd(8)`、`groupmod(8)`、および `groupdel(8)` で説明されています。
- SysMan ユーティリティは、`sysman(8)` および `sysman_cli(8)` で説明されています。
- アカウント・マネージャ (`dxaccounts`) の起動については、`dxaccounts(8)` で説明されています。
- システム・ファイルは、`passwd(4)`、`group(4)`、`shells(4)`、および `default(4)` で説明されています。
- 個々のコマンドは、`passwd(1)`、`vipw(8)`、`grpck(8)`、および `pwck(8)` で説明されています。

7.1.3.3 オンライン・ヘルプ

SysMan Menu の [アカウント] オプションおよびアカウント・マネージャ `dxaccounts` には、すべてのオプションを説明し、適切なデータの入力方法を示すオンライン・ヘルプがあります。

コマンド行ルーチンの中には、コマンド構文のヘルプ・テキストを表示するものがあります。このヘルプ・テキストは、`-h` または `-help` コマンド・フラグを指定して表示します。

7.1.4 関連ユーティリティ

次に示すリソースも、アカウントの管理に役立ちます。これらのコマンドおよびユーティリティは、システム・クラッシュ後にグラフィカル・ユーザ環境が利用できない場合や、基本的な文字セル端末しかアクセスできない場合に、システムの問題を修正するのに役立つことがあります。

`vipw`

`vipw` を使用すると、テキスト・エディタを起動してパスワード・ファイルを手動で編集することができます。ファイルの編集はできれば手動で行わずに、ユーティリティを使用してください。 `vipw` ユーティリティを使用してローカルのパスワード・データ

ベースを編集することはできますが、NIS データベースを編集することはできません。また、エンハンスド・セキュリティ機能を備えたシステム上でこのユーティリティを使用することはできません。

`vipw` ユーティリティでは、`passwd` ファイルを編集するだけでなく、他から使用できないように、ファイルがロックされます。このコマンドは、`root` のパスワード・エントリの整合性をチェックして、`passwd` ファイルに間違ったルート・パスワードが入力されないようにします。スタンドアロン・モードでは、`vipw` ユーティリティを使用して、破損した `passwd` ファイルにパッチをあてることもできます。

詳細は、`vipw(8)` を参照してください。

`who`

現在ログインしているユーザを表示します。詳細は、`who(1)` を参照してください。

`finger`

パスワード・ファイルに記述されているユーザ情報を表示します。詳細は、`finger(1)` を参照してください。

`cs``sh`、`k``sh`、および `s``h`

`cs``sh`、`k``sh`、および `s``h` コマンドは、C シェル、Korn シェルおよび POSIX シェルを起動しコマンドを処理します。

`grpck`

`grpck` コマンドを使用すると、`group` ファイルの整合性を検査できます。

`pwck`

`pwck` ユーティリティを使用すると、`group` ファイルと `passwd` ファイルの整合性を検査することができます。

`quotaon`

`quotaon` コマンドを使用すると、クォータ (制限) 情報を有効または無効にできます。

passwd, chfn, および chsh

passwd, chfn, および chsh コマンドを使用すると、ユーザはパスワード・ファイル情報を変更できます。passwd では、パスワードを変更できます。chfn では、フル・ネームを変更できます。chsh では、ログイン・シェルを変更できます。

7.2 アカウント管理 - クイック・スタート

以降の項では、アカウント管理ユーティリティの起動手順について簡単に説明します。この手順により、基本的なアカウントを簡単に作成することができます。たとえば、root として、システムをインストールして構成した後に、とりあえず、アカウントの省略時の設定を使用して一般ユーザ・アカウントを設定することができます。後に、7.3 節 および他の節を読むことで、システムの省略値の構成方法を理解し、アカウントおよびグループ管理ユーティリティの高度な機能を使用することができます。

7.2.1 システム・セットアップの際の初期アカウントの作成

オペレーティング・システムのフル・インストール後に、root ユーザとして最初にログインすると、順を追ってシステムの構成オプションを示す「システム・セットアップ」ユーティリティが表示されます。「システム・セットアップ」の「アカウント・マネージャ」(dxaccounts) アイコンを使用して、初期アカウントを構成することができます。このアイコンでは、共通デスクトップ環境 (CDE) または他の X ウィンドウ環境で動作する、X11 準拠の GUI が起動されます。アカウント・マネージャの使用についての詳細は、7.5.2 項を参照してください。ASU がインストールされて構成されている場合は、アカウント・マネージャ (dxaccounts) GUI を使用して、Windows NT ドメイン・アカウントを管理することができます。これについては、7.6 節で説明しています。

7.2.2 アカウント・マネージャ (dxaccounts) GUI の使用

アカウント・マネージャ (dxaccounts) には、ドラッグ・アンド・ドロップやカット・アンド・ペーストなど、CDE 環境でサポートされている機能が用意されており、既存のアカウントから新しいアカウントを素早くクローニングできます。この GUI は、次の方法で起動できます。

- 次のコマンドを端末から実行して、X11 準拠のウィンドウ環境で GUI を起動します。

dxaccounts

- CDE では、フロント・パネルで、[アプリケーション・マネージャ] または [SysMan アプリケーション] ポップアップ・メニューをオープンします。[日常管理] を選択し、「アカウント・マネージャ」アイコンをクリックします。

アカウント・マネージャ GUI (dxaccounts) には、ASU がインストールされている場合の Windows NT ドメイン・ユーザを管理するオプションも用意されています。ASU がインストール/構成されていない場合、これらのオプションはウィンドウ上で淡色表示になります。

アカウント・マネージャ GUI (dxaccounts) を使用して、シェルや親ディレクトリなど、ユーザ・アカウントの省略時のオプションを構成することもできます。詳細は、7.4.2.6 項を参照してください。

7.2.3 SysMan Menu の [アカウント] オプションの使用

SysMan Menu の [アカウント] オプションでは、dxaccounts と同じ機能を使用できますが、次の機能のサポートに制限があります。

- PC クライアントの Windows NT ドメイン・アカウントの管理
- エンハンスド (C2) セキュリティ下のアカウントの管理

SysMan Menu の [アカウント] オプションは、CDE の「アプリケーション・マネージャ」や、CDE のフロント・パネル ([SysMan アプリケーション] メニュー) で、または次のようにコマンド行で実行することができます。

sysman accounts

[アカウント] オプションを使用して、ネットワーク情報サービス (NIS) 環境と LDAP (Lightweight Directory Access Protocol) 環境でアカウントを追加したり、変更したりすることもできます。ローカル・ユーザを NIS 環境に追加することなく、システムに追加することができます。NIS については、『ネットワーク管理ガイド：サービス編』を参照してください。

SysMan Menu で [アカウント] オプションを使用するには、第 1 章に説明されているように SysMan Menu を表示して、次のようにオプションを展開します。

1. [アカウント] オプションを選択して、メニューを展開します。次のメニュー・オプションが表示されます。

- ローカル・ユーザの管理
 - ローカル・グループの管理
 - NIS ユーザの管理
 - NIS グループの管理
 - LDAP ユーザの管理
 - LDAP グループの管理
2. マウス・ポインタ (または Tab キー) を使用してオプションを選択します。マウスの第一ボタン (MB1) をクリックするか、または Enter キーを押してユーティリティを起動します。
 3. ユーティリティの初期ウィンドウ (画面) が開きます。次のオプションがあります。

[追加...]

このオプションを使用して、新しいユーザ・アカウントを作成します。

[修正...]

このオプションを使用して、既存のユーザ・アカウントのアカウント詳細を変更します。

[削除...]

このオプションを使用して、ユーザ・アカウントと、そのシステム・リソースすべて (省略可能) を削除します。

[フィルタ...]

このオプションを使用して、特定のユーザまたはユーザ群をフィルタリング (検索) します。ユーザの UID やアカウントのコメントなど、さまざまな検索条件を指定できます。

[オプション...]

このオプションを使用して、フィルタリングを自動的に開始するアカウントの数を定義します。また、ユーザ・アカウントのリストにどのユーザ・データを含めるかも選択できます。

これらのユーティリティの使用についての詳細は、7.5.1 項およびオンライン・ヘルプで説明しています。

7.2.4 コマンド行ユーティリティの使用

アカウントおよびグループの管理に使用できるコマンド行ユーティリティは次のとおりです。

| | |
|---|--|
| <code>useradd</code> , <code>usermod</code> , および <code>userdel</code> | これらのコマンドを使用してユーザ・アカウントを追加、変更または削除します。 |
| <code>groupadd</code> , <code>groupmod</code> , および <code>groupdel</code> | これらのコマンドを使用して、グループを追加、変更および削除します。 |
| <code>adduser</code> および <code>addgroup</code> | <code>adduser(8)</code> および <code>addgroup(8)</code> に説明されているこれらのユーティリティは、旧バージョンとの互換性のためだけに提供されている古い対話式のスクリプトです。これらのスクリプトをまだ使用している場合は、文字セル端末や Windows NT などを含む作業環境をサポートする新しいユーティリティに移行する必要があります。 |

また、コマンド行ユーティリティは、ASU がインストールされている場合に Windows NT ドメインの管理オプションを提供します。

7.2.5 Advanced Server for UNIX

ASU (Advanced Server for UNIX) は、UNIX オペレーティング・システムを実行しているサーバに Windows NT バージョン 4.0 サーバのサービスおよび機能を実装するレイヤード・アプリケーションです。これを使用すると、Windows を実行しているシステムにとって、UNIX システムは Windows NT バージョン 4.0 サーバのように見えます。ASU を使用することで、UNIX ファイル・システムおよびプリンタを共通資源として共用することができます。クライアント Windows ユーザが、UNIX リソースを共用するためには、省略時構成では Windows NT ドメイン・アカウントと UNIX アカウントの両方が必要です。ASU が動作している場合、この章で説明する UNIX アカウント管理ユーティリティを使用して、アカウントの新規作成など、アカウントの特定の管理作業を行うことができます。

ASU ソフトウェアは「*Associated Products Volume 2*」CD-ROM で配布され、オペレーティング・システムのライセンスで 2 台まで自由に接続できます。それ以上の台数で使用する場合は、別途ライセンスが必要です。ソフトウェア・キットに添付されている『*Advanced Server for UNIX* インストールेशन/管理ガイド』を参照してください。

7.3 ユーザ・アカウントおよびグループの概要

ユーザ・アカウントおよびグループの管理には、システムのパスワード・ファイルおよびグループ・ファイルの内容の管理が伴います。スタンドアロン・システムでは、管理するファイルは `/etc/passwd` (`passwd(1)` を参照) および `/etc/group` (`group(4)` を参照) です。

ネットワーク・システムでは、通常、ネットワーク情報サービス (NIS) または LDAP (Lightweight Directory Access Protocol) を使用して、アカウントおよびグループが中央で管理されます。NIS または LDAP を使用すると、ネットワーク上のシステムは、共通のパスワード・ファイルとグループ・ファイルを使用することができます。詳細については、『ネットワーク管理ガイド：サービス編』と www.OpenLDAP.org を参照してください。

エンハンスド (C2) セキュリティ機能がシステムで利用できる場合は、セキュリティのために `/etc/passwd` ファイル以外のファイルも管理する必要があります。たとえば、セキュリティ関連情報 (最小パスワード長やパスワードの満了日など) のために、保護パスワード・データベースが使用されます。これらのタスクについては、『セキュリティ管理ガイド』に説明があります。

7.3.1 システム・ファイル

次のシステム・ファイルは、アカウント管理作業を実行するときにアップデートされます。これらのファイル・システムは定期的にバックアップする必要があります。

`/etc/group`

`/etc/group` ファイルには、グループ・データを記述します。各行ではそれぞれ、グループ名、オプションの暗号化されたパスワード、数字のグループ ID、セカンダリ・グループ内のすべてのユーザのリストを、たとえば次のように指定します。

```
system:*:0:root luis
daemon:*:1:daemon
uucp:*:2:uucp
```

```
mem:*:3:
kmem:*:3:root
bin:*:4:bin,adm
sec:*:5:
cron:*:14:
.
.
.
users:*:15:billP carsonK raviL annieO
sysadmin:*:16:
tape:*:17:
.
.
.
```

/etc/passwd

/etc/passwd ファイルには、各ユーザごとに 1 行のレコードが含まれています。各レコードは、7 個のフィールドから構成されています。詳細については、7.3.3 項を参照してください。このファイルの例を次に示します。

```
carsonK:6xl6duyF4JaEI:200:15:Kit Carson,3x192,1-6942,
:/usr/users/carsonK:/bin/sh
annieO:.murv3nlpG2Dg:200:15:Annie Olsen,3x782,1-6982,
:/usr/users/annieO:/bin/sh
```

例ではページに入るように改行されていますが、ファイル内ではそれぞれ 1 行です。

/usr/skel

/usr/skel ディレクトリには、.login ファイルなど、新規アカウント用のスケルトン・ファイルが置かれています。各ユーザは、これらのファイルを編集して、ローカル環境に合わせてアカウントをカスタマイズすることができます。たとえば、環境変数を定義したり、プログラムまたはプロジェクト・ファイルの省略時のパスを定義します。/etc/shells ファイルには、システムで利用できるコマンド・シェルのリストが含まれています。

ログ・ファイル

ログ・ファイル `/var/adm/wtmp` と `/var/adm/utmp` および `/usr/var/adm/syslog.dated` ディレクトリ内のログ・ファイルには、アカウントの使用状況に関する情報が記録されます。

エンハンスド・セキュリティ機能を使用している場合は、以下のセキュリティ・ファイルが関係してきます。

- `/etc/auth/system/default`
- `/tcb/files/auth.db`
- `/var/tcb/files/auth.db`

ネットワーク情報サービス (NIS) を使用している場合は、以下のNIS ファイルが関係してきます。NIS マスタ・データベースにあるこれらのファイルをバックアップすることを忘れないでください。

- `/var/yp/src/group`
- `/var/yp/src/passwd`
- `/var/yp/src/prpasswd`

LDAP 情報は、LDAP データベースに格納されています。このデータベースは、バックアップする必要があります。

以下のログ・ファイルには、アカウントの使用状況に関する情報が記録されます。

- `/var/adm/wtmp`
- `/var/adm/utmp`
- `/usr/var/adm/syslog.dated` ディレクトリ内のログ・ファイル

7.3.2 識別子 (UID と GID) の概要

各ユーザ・アカウントは、ユーザ識別子 (UID) と呼ばれる一意の番号によって認識されます。また、各ユーザはグループ識別子 (GID) と呼ばれる一意の番号によっても認識されます。システムはこれらの番号を使用して、ユーザ・ファイルのアクセス許可およびグループの特権を追跡し、ユーザ課金統計情報を収集します。

UID および GID に指定できる最大数は、4,294,967,294 (32 ビット、このうち 2 つの値が予約済) です。ログオンできるユーザの最大数は、使用できるシステム資源によって決まります。ただし、実際の値はその最大数よりもはるかに小さくなります。最大数の UID と GID を使用する場合、古いユーティリティおよびアプリケーションの中にはこの最大値をサポートしていないものがあるため、次の注意が必要です。

- 最新バージョンのエンド・ユーザ・アプリケーションを実行していない場合は、それらがサポートしている UID と GID の最大数を確認します。たとえば、一般に広く使用されている Kerberos バージョン 4.0 は、特定の範囲を超える UID および GID をサポートしていません。Kerberos バージョン 4.0 を現在使用している場合は、バージョン 5.0 にアップグレードすることを考慮してください。同様に、PATHWORKS を使用している場合は、ASU バージョン 4.0 以上にアップグレードすることを考慮してください。
- System V File System (S5FS) では、UID および GID の最大範囲をサポートしていません。UID および GID に 65,535 を超える値を指定したファイル・システム `syscall` では、`EINVAL` エラーが返されます。UID または GID として 65,535 を超える値が割り当てられたユーザは、System V File System 上にファイルを作成することができません。解決策としては、UFS、または AdvFS の使用を検討してください。
- UID と GID の最大範囲を増やした場合は、特定のコマンドとユーティリティの動作が変わっている可能性があります。以下の変化がコマンドの使用に影響しないか、シェル・スクリプトなどを確認してください。
 - `ls -l` コマンドで、クォータ・ファイルまたはスパース・ファイルのディスク・ブロック使用量が表示されなくなります。ファイルの実際のディスク・ブロック使用量を表示するには、`ls -s` コマンドを使用します。
 - `cp` コマンドで、クォータ・ファイルやその他のスパース・ファイルを正しくコピーできなくなります。クォータ・ファイルや他のスパース・ファイルを正しくコピーするには、次のように `conv=sparse` パラメータを指定して `dd` コマンドを使用します。

```
# dd conv=sparse if=inputfile of=outputfile
```
 - クォータ・ファイルや他のスパース・ファイルを含む UFS ファイル・システムを、`vdump` ユーティリティを使用してバックアップ

プし、vrestore ユーティリティを使用してリストアした場合、クォータ・ファイルや他のスパーズ・ファイルは次のようにリストアされます。

- ディスク上のファイルの最初のページは、すべて占有されているページとしてリストアされます。つまり、割り当てられていない空のディスク・ブロックは 0 で埋められます。
- ディスク上のこれ以降のページはスパーズとして (きっちり埋まっていないページとして) リストアされます。

7.3.3 パスワード・ファイル

スタンドアロン・システムでは、passwd ファイルによって、root も含めシステムの各ユーザが識別されます。passwd ファイルの各エントリは、1 行に記述し、7 個のフィールドがあります。各フィールドはコロンで区切られ、最後のフィールドは復帰改行文字で終わります。各エントリの構文および各フィールドの意味は次のとおりです。

```
username:password:user_id:group_id:user_info:login_directory:login_shell
```

| | |
|-----------------------|---|
| <code>username</code> | ユーザ・アカウントの名前。username は一意であって、1 ~ 8 文字の英数字でなければなりません。 |
| <code>password</code> | パスワードは、直接入力することはできません。アカウントへのログインを不可にするには、* (アスタリスク) をその passwd フィールドに記述します。パスワード・フィールドが空の場合は、そのログイン名を知っていれば誰でもそのユーザとしてシステムにログインすることができます。 |
| <code>user_id</code> | このアカウントの UID (整数で指定)。この値は、システムの各ユーザに対応しており、一意でなければなりません。UID 0 は、root 用です。各 UID は、100 から昇順に割り当てます。100 未満の番号は、bin や daemon などの擬似ユーザに使用されます。(/usr/include/limits.h ファイルも参照してください。) |

| | |
|--|--|
| <code>group_id</code> | このアカウントの GID (整数で指定)。この上限値については、『 <i>Tru64 UNIX 概要</i> 』を参照してください。GID 0 は、 <code>system</code> グループ用です。この GID は、 <code>group</code> ファイルに必ず定義しておきます。 |
| <code>user_info</code> (または GECOS データ) | このフィールドには、ユーザのフルネーム、オフィスの住所、電話の内線番号、および自宅の電話番号などの補足のユーザ情報を記述します。 <code>user_info</code> フィールドの情報は <code>finger</code> コマンドが読み取ります。ユーザは、 <code>chfn</code> コマンドで <code>user_info</code> フィールドの内容を変更することができます。詳細は、リファレンス・ページの <code>finger(1)</code> および <code>chfn(1)</code> を参照してください。 |
| <code>login_directory</code> | ログイン直後にユーザ・アカウントが置かれるディレクトリの絶対パス名。このパス名は、 <code>login</code> プログラムにより <code>HOME</code> 環境変数に割り当てられます。 <code>HOME</code> 変数の値はユーザが変更することができますが、値を変更すると、ホーム・ディレクトリとログイン・ディレクトリは別のディレクトリとなります。 <code>passwd</code> ファイルにユーザ・アカウントを追加した後、ログイン・ディレクトリを作成します。通常、ユーザの名前をログイン・ディレクトリ名として使用します。ログイン・ディレクトリの作成についての詳細は、 <code>chown(1)</code> 、 <code>mkdir(1)</code> 、 <code>chmod(1)</code> 、および <code>chgrp(1)</code> を参照してください。 |
| <code>login_shell</code> | ユーザがログインした後に起動するプログラムの絶対パス名。このフィールドに何も指定しないと、 <code>Bourne</code> シェル <code>/bin/sh</code> が起動されます。 <code>Bourne</code> シェルについての詳細は、 <code>sh(1b)</code> を参照してください。ユーザは、 <code>chsh</code> コマンドを使用してログイン・シェルを変更することができます。詳細は、 <code>chsh(1)</code> を参照してください。 |

ウィンドウ (グラフィカル) ユーザ環境では、アカウント・マネージャ (dxaccounts) などのユーティリティを使用して、passwd および mkdir などのコマンドによる操作のすべてを実行することができます。

コマンド行ユーティリティでは、省略時の特性の設定および変更を、制限なく行うことができますが、グラフィカル・ユーティリティの中には、省略時の特性を新規アカウントに対してだけに限って、設定できるものがあります。アカウント・マネージャ (dxaccounts) を使用した操作については、7.4.2.6 項を参照してください。

/etc/passwd ファイルが非常に大きい場合は、性能が低下することがあります。passwd のエントリ数が 30,000 を超える場合、mkpasswd はハッシュ (ndbm) データベースの作成に失敗することがあります。このデータベースの目的は、パスワード・ファイル情報の効率的な (速い) 探索であるため、このデータベースの構築に失敗すると、このデータベースに依存するコマンドは、/etc/passwd を順次探索するようになります。その結果、これらのコマンドで大幅な性能低下が起こります。

mkpasswd -s オプションを使用して、この種の障害を回避することができますが、潜在的なデータベースまたはバイナリ互換の問題が生ずる可能性があります。mkpasswd で作成されたパスワード・データベースにアクセスするアプリケーションが静的に (非共用で) 構築されている場合、そのアプリケーションはパスワード・データベースに対して、読み取りまたは書き込みを正しく行うことができなくなります。このため、アプリケーションは、誤った結果を生成して失敗するか、またはコアをダンプを出力して失敗します。

静的にリンクされたアプリケーションは、ndbm(3) リファレンス・ページに記載されている libc ndbm ルーチンのいずれかを直接あるいは間接的に呼び出したのち、パスワード・データベースにアクセスすると、影響を受けることがあります。この状態を解決するには、アプリケーションを再リンクする必要があります。この互換性の問題を避けるには、mkpasswd -s オプションを使用しないでください。

注意

NIS 環境では、ローカルの passwd ファイルあるいは NIS 分散 passwd ファイルのいずれかにユーザ・アカウントを追加することができます。ローカルの passwd ファイルにアカウントを追加した場合、そのアカウントはローカル・システムでのみ認識され

ます。NIS 分散 `passwd` ファイルにアカウントを追加した場合、追加したアカウントは、その分散 `passwd` ファイルにアクセスできるすべての NIS クライアントで認識されます。分散環境におけるユーザの追加についての詳細は、`nis_manual_setup(7)` を参照してください。

同様に、LDAP ユーザもグローバルです。

7.3.4 グループ・ファイル

すべてのユーザは、少なくとも 1 つのグループのメンバです。group ファイルでは、ユーザのグループ名を定義します。ユーザは必ずいずれかのグループに属します。ユーザ・アカウントをグループ化する理由には、主として次の 2 つがあります。

- 複数のユーザが同一のファイルおよびディレクトリで作業する場合、これらのユーザをグループ化することによって、ファイルおよびディレクトリへのアクセスが簡略化される。
- システム・ファイルまたはディレクトリへのアクセスが許可されている特権ユーザをグループ化することにより、特権ユーザの識別が容易になる。

group ファイルは次の目的で使用されます。

- `passwd` ファイルで使用されているグループ識別番号に名前を割り当てる。
- ユーザ・アカウントを複数のグループ・エントリに追加することによって、ユーザが複数のグループに所属できるようにする。

group ファイルの各エントリは、1 行で記述され、4 個のフィールドがあります。各フィールドはコロンで区切られ、最後のフィールドは改行文字で終わります。エントリの構文および各フィールドの意味は、次のとおりです。

```
groupname: password: group_id: user1 [user2,..., userN]
```

groupname

このエントリによって定義されるグループの名前。
groupname は、1 ~ 8 文字の英数字からなり、一意でなければなりません。

| | |
|-----------------|---|
| <i>password</i> | このフィールドには * (アスタリスク) を記述します。 このフィールドのエントリは現在は無視されます。 |
| <i>group_id</i> | このグループのグループ識別番号 (GID) (整数で指定)。この範囲の上限については、『 <i>Tru64 UNIX 概要</i> 』を参照してください。GID 0 は、システム用です。GID は一意でなければなりません。 |
| <i>user</i> | <code>passwd</code> ファイルで定義されたユーザ名で識別される、このグループに属するユーザ・アカウント。2 人以上のユーザがグループに属している場合、各ユーザ・アカウントはコンマで区切られます。最後のユーザ・アカウントは、改行文字で終わります。ユーザは複数のグループのメンバになることができます。 |

リファレンス・ページの `group(4)` で説明しているように、1 人のユーザが属することができるグループ数には制限があります。行の長さの上限は、`/usr/include/limits.h` ファイルで `LINE_MAX` として定義されています。ユーザ・アカウントを複数の管理上のグループに分割することをお勧めします。

特定の GID の省略値を、グラフィカル・ユーティリティまたはコマンド行ユーティリティを使用して設定することもできます。アカウント・マネージャ GUI (`dxaccounts`) による省略値の設定方法については、7.4.2.6 項を参照してください。

7.4 ユーザ・アカウントの管理

以降の各項で次の管理方法について説明します。

- SysMan Menu オプションによるユーザ・アカウントの管理。この方法を使用して、ユーザをネットワーク情報サービス (NIS) 環境と LDAP (Lightweight Directory Access Protocol) 環境に追加することもできます。SysMan Menu の起動と [ローカル・ユーザの管理] オプションの選択については、7.2.3 項で説明しています。
- アカウント・マネージャ GUI (`dxaccounts`) によるローカル・ユーザと NIS ユーザ、および関連する Windows NT ドメイン・アカウント

の管理。アカウント・マネージャ GUI の起動については、7.2.2 項で説明しています。

useradd コマンド行ユーティリティを使用する方法もあり、リファレンス・ページに説明されていますが、このユーティリティでは、NIS アカウントはサポートしていません。NIS については『ネットワーク管理ガイド：サービス編』を参照してください。SysMan Menu の [アカウント] オプションは、端末、X11、または Java クライアントから使用することができます。

注意

adduser はすべてのオプションを提供していないため、またそのセキュリティを詳細に設定できないため、このユーティリティは使用しないでください。システム・ファイルの整合性を維持するために、ユーザ・アカウントを手動で追加しないでください。

7.4.1 SysMan Menu の [アカウント] オプションの使用

以降の各項では、SysMan Menu オプションを使用して新規アカウントを作成する方法について説明します。次の作業について説明します。

- アカウント情報の収集 (7.4.1.1 項)
- ローカル・アカウント、NIS アカウント、および LDAP アカウントに適用されるアカウント・オプションの設定 (7.4.1.2 項)
- ローカル・アカウント、NIS アカウント、および LDAP アカウントでのアカウント検索に適用されるフィルタ・オプションの使用 (7.4.1.3 項)
- ローカル・ユーザ・アカウントの作成または変更 (7.4.1.4 項)
- ローカル・ユーザ・アカウントの削除 (7.4.1.5 項)
- LDAP および NIS ユーザ・アカウントの作成または変更 (7.4.1.6 項)
- LDAP および NIS ユーザ・アカウントの削除 (7.4.1.7 項)

SysMan Menu ユーティリティのフィールドにデータをキーボードを使って入力する方法については、オンライン・ヘルプを参照してください。

7.4.1.1 アカウント情報の収集

アカウントの管理を始める前に、表 7-2 のワークシートを使って情報を収集します。エンハンスド・セキュリティ機能を使用している場合、データ項目

は最低限の必要条件 (パスワードの長さなど) を満たす必要があります。詳細については、『セキュリティ管理ガイド』を参照してください。

passwd ファイルのデータ項目の説明については、7.3.3 項を参照してください。

表 7-2: アカウント管理ワークシート

| フィールド | 説明 | データ項目 |
|---------------|---------------------|-------|
| ユーザ名* | | |
| コメント (gecos) | フルネーム | |
| | 場所 | |
| | 電話番号 | |
| ユーザ ID (UID)* | 自動的に割り当てられる | |
| パスワード* | 大文字と小文字または英数字を混在させる | |
| プライマリ・グループ* | 自動的に割り当てられる | |
| セカンダリ・グループ | | |
| シェル | 選択可能 | |
| ホーム・ディレクトリ* | 自動作成可能 | |
| アカウントのロック | | |
| ローカル・ユーザ | | |
| NIS ユーザ | | |
| Windows ユーザ | 共用が必要 | |

* は必須フィールドであることを示しています。

一般的なユーザ・データの例を、表 7-3 に示します。

表 7-3: データの例を記入したアカウント管理ワークシート

| フィールド | 説明 | データ項目 |
|---------------|---------------------|--------------------------|
| ユーザ名* | | carsonK |
| コメント (gecos) | フルネーム | Kit Carson |
| | 場所 | オフィス 3T-34 |
| | 電話番号 | 4-5132 |
| ユーザ ID (UID)* | 自動的に割り当てられる | 次の使用可能な UID を使用 |
| パスワード* | 大文字と小文字または英数字を混在させる | サイト固有の初期パスワードを使用 |
| プライマリ・グループ | 自動的に割り当てられる | users |
| セカンダリ・グループ | | marsx , 25 |
| シェル | 選択可能 | ksh |
| ホーム・ディレクトリ* | 自動作成 | /usr/marsx/carsonK |
| アカウントのロック | | no |
| ローカル・ユーザ | | no |
| NIS ユーザ | | yes |
| Windows ユーザ | | yes , \\maul\astools を共有 |

* は必須フィールドであることを示しています。

7.4.1.2 フィルタ・オプションと表示オプションの設定

フィルタ・オプション (7.4.1.3 項を参照) と表示オプションを構成するには、SysMan の [ローカル・ユーザの管理] の [オプション...] を使用します。オプションを設定するには、SysMan Menu を起動し、7.2.3 項の説明のように [ローカル・ユーザの管理] オプションを選択します。

[オプション...] ボタンを選択すると、「SysMan アカウント管理: プログラム・オプション」ウィンドウがオープンされ、次の設定を構成できます。こ

ここでは、一部のオプション名を短縮しています。短縮されている名前は、ウィンドウ内では内容を示す行として表示されます。

「開始時に合計ユーザ数が右記の数字を越えている場合に警告」

このオプションは、フィルタ機能のトリガ値の設定に使用します。省略時の設定は、200 ユーザ・アカウントです。

この機能は、ユーザ・アカウントが多数 (数百または数千) ある場合に便利です。システム上のアカウントが多いほど、SysMan アカウント・タスクがすべてのアカウントを見つけて表示するのに時間がかかります。トリガ値を設定すると、省略時の設定で、SysMan アカウント・タスクがスタートアップ時にフィルタ (検索) モードになります。特定のアカウントやアカウント群を選択できるようになるため、検索および表示の時間が大幅に短縮されます。

たとえば、300 ユーザ・アカウントを設定した場合、アカウント数が 300 を超えた場合だけ、SysMan 「アカウント」が省略時の設定でフィルタ・モードになります。

「ユーザ名」

このチェックボックスは、すべてのアカウント・リスト内にユーザ・アカウント名が表示されるようにします。

「ユーザ ID (UID)」

このチェックボックスは、すべてのアカウント・リスト内にユーザ識別子 (UID) が表示されるようにします。

「コメント」

このチェックボックスは、すべてのアカウント・リスト内にアカウントのコメント (場所や電話番号など) が表示されるようにします。

チェックボックスを選択すると、フィルタ・オプションに影響します。アカウントのフィルタは、表示されたデータだけをベースに、アカウントをフィルタリングします。

7.4.1.3 フィルタ・オプションの使用

アカウントが多数ある場合は、[フィルタ...] オプションを使用すると、特定のアカウントやアカウントのグループを素早く見つけることができます。

フィルタは、[オプション...] (7.4.1.2 項を参照) を設定することで自動的に起動できます。自動起動を使用すると、アカウント・マネージャがすべてのユーザ・アカウント・データを見つけてロードする際の遅延を回避できます。この機能を使用して、ローカル・アカウントと NIS アカウントの両方をフィルタリングできます。

検索およびフィルタ・オプションを使用するには、SysMan Menu を起動し、[ローカル・ユーザの管理] オプションを選択します (7.2.3 項を参照)。[フィルタ...] を選択して、「ローカル・ユーザの管理: 表示」というタイトルのダイアログ・ボックスをオープンします。このウィンドウを使用すると、簡単な検索や高度な検索ができます。

簡単な検索を行うには、次の手順を実行します。

フィルタ (検索文字列) を 1 つ入力するか、フィルタのセットを入力します。簡単な検索はすべて、次のように入力されたアカウント名をベースに行われます。

- s_kahn などの、個々のユーザ名
- *khan や ?_khan などの、ワイルドカード・パターン
- *khan, kim, donny_w, tom* のような、コンマで区切られたユーザ名またはワイルドカード・パターンのリスト

フィルタ指定と一致したアカウントは、「ローカル・ユーザの管理」ウィンドウにリストされます。このウィンドウの上部には、オリジナルのフィルタ文字列が表示されます。

高度な検索を実行するには、次の手順に従います。

[拡張] を選択し、追加のフィルタ・オプションを表示します。検索オプションを起動するには、チェックボックスを選択します。

次のフィルタ・オプションがあります。

「検索するユーザ名 (フィルタ)」

簡単な検索オプションで説明したのと同様にフィルタを入力します。

「検索するユーザ ID の範囲」

1-100 のように UID の制限範囲を入力するか、終端なしの範囲 (UID が 100 以上のアカウントをすべて見つけるためには 100-、

UID が 100 以下のアカウントをすべてを見つけるために -100 など) を入力します。

「検索するコメント・フィールドのパターン」

ユーザ・アカウントを作成する際に「コメント」(GECOS データ) フィールドに入力されたデータで検索する検索パターンを入力します。

このデータは、電話番号や、物理的位置、その他のユーザ固有情報です。このパターンの定義には、アスタリスク (*) や疑問符 (?) のワイルドカードが使用できます。たとえば、**string** や、**Sub** などです。

「LOCKED または UNLOCKED 検索基準」

このオプションを使用すると、ロックされたアカウントやロック解除されたアカウントを含める (または除外する) ことができます。このオプションを使用して、現在ロックされているすべてのアカウントを見つけることができます。

簡易検索の内容をクリアしないで拡張検索を起動すると、警告ダイアログがオープンします。この警告ダイアログが表示された場合、[了解] を選択して高度な検索を確定します。これにより、簡単な検索で指定した検索方法が無効になります。

7.4.1.4 ローカル・アカウントの作成と変更

新しいアカウントを作成するには、7.2.3 項に説明されているように、SysMan Menu を表示して [ローカル・ユーザの管理] オプションを選択します。既存のローカル・ユーザ・アカウントのすべてのリストが表示されます。

オンライン・ヘルプでは、これらのフィールドとその有効なデータについて説明しています。

次の手順に従ってローカル・ユーザを追加します。

1. [追加...] オプションを選択して「ローカル・ユーザの管理: ユーザの追加」ダイアログ・ボックスをオープンします。
2. 表 7-2 のワークシートの情報を使用して、データ・フィールドを入力します。

3. ほかに、NIS オプションが必要な場合は、[オプション...] を選択します。「オプション」ダイアログ・ボックスがオープンします。適切な NIS オプションを選択して、[了解] ボタンを選択し、「ユーザの追加」ダイアログボックスに戻ります。
4. [了解] ボタンを選択して、新規ユーザを追加します。パスワードの入力ミスの確認など、エラーがある場合は、ユーティリティからその修正を促すプロンプトが表示されます。
5. 「ローカル・ユーザ」ウィンドウがオープンし、確認メッセージが示されます。[了解] を選択して、SysMan Menu に戻ります。

既存のアカウントを変更するには、7.2.3 項 に説明されているように、SysMan Menu を表示して [ローカル・ユーザの管理] オプションを選択します。ローカル・ユーザ・テーブルが表示され、既存のローカル・ユーザ・アカウントがリストされます。

オンライン・ヘルプでは、これらのフィールドとその有効なデータについて説明しています。

次の手順に従ってユーザのエントリを変更します。

1. ユーザのリストをスクロールして、変更するエントリを選択します。
2. [修正...] を選択して「ローカル・ユーザ管理: ユーザの修正」ウィンドウをオープンします。
3. 必要に応じて、データ・フィールドの値を変更します。
4. NIS オプションの追加が必要な場合は、[オプション...] を選択します。「オプション」ダイアログ・ボックスがオープンします。適切な NIS オプションを選択して、[了解] を選択し、「ユーザの変更」ウィンドウに戻ります。

複数のアカウントを追加または変更するには、[了解] の代わりに [適用] を選択します。加えたすべての変更は、[了解] を選択して作業を終了するまで有効になりません。

5. [了解] を選択して変更を確認します。パスワードの入力ミスの確認など、エラーがある場合は、ユーティリティからその修正を促すプロンプトが表示されます。
6. 「ローカル・ユーザ管理」ウィンドウがオープンし、確認メッセージが示されます。[了解] を選択して、SysMan Menu に戻ります。

7.4.1.5 ローカル・アカウントの削除

アカウントを削除する前に、次の事項を検討してください。

- 削除する代わりに、[修正...] を使用してアカウントをロックできます。また、[修正...] を使用してこのアカウントを他の新しいユーザに転送し、そのアカウントの詳細を変更することができます。
- アカウントを削除する前に、dxarchiver ユーティリティを起動してユーザのディレクトリおよびファイルの圧縮したアーカイブ・ファイルを作成することもできます。詳細については、dxarchiver(8) を参照してください。

アカウントを削除するには、7.2.3 項 に説明されているように、[ローカル・ユーザ管理] オプションを選択します。ローカル・ユーザ・テーブルが表示され、既存のアカウントがすべてリストされます。次の手順に従ってユーザを削除します。

1. ユーザのリストをスクロールして、削除するユーザ・アカウントを選択します。
2. [削除...] を選択して、「ローカル・ユーザ管理: ユーザの削除」ダイアログ・ボックスをオープンします。
3. ユーザのリソースを削除してディスク・スペースを解放する場合は、「ユーザとディレクトリとファイルを削除」を選択します。
4. [了解] を選択してアカウントを削除します。ローカル・ユーザのリストが即座にアップデートされます。

7.4.1.6 LDAP アカウントと NIS アカウントの作成と変更

新しい LDAP アカウントまたは NIS アカウントを作成するには、SysMan Menu を表示して、7.2.3 項 に説明されているように、[NIS ユーザの管理] オプションまたは [LDAP ユーザの管理] オプションを選択します。「LDAP Users」テーブルまたは「NIS Users」テーブルが表示され、既存のローカル・ユーザ・アカウントがすべてリストされます。次の手順に従ってローカル・ユーザのアカウントを作成します。

1. [追加...] オプションを選択して、「LDAP ユーザの管理: ユーザの追加」ウィンドウまたは「NIS ユーザの管理: ユーザの追加」ウィンドウをオープンします。

2. 表 7-2 に示されているワークシートの情報を使って、データ・フィールドを入力します。
3. [了解] を選択して新しいユーザを追加します。パスワードの入力ミスの確認など、エラーがある場合は、ユーティリティからその修正を促すプロンプトが表示されます。
4. 「LDAP ユーザ管理」ウィンドウまたは「NIS ユーザ管理」ウィンドウがオープンし、追加が正常に行われたことを示す確認メッセージが表示されます。[了解] を選択して、SysMan Menu に戻ります。

既存のアカウントを変更するには、SysMan Menu を表示して、7.2.3 項に説明されているように、[LDAP ユーザの管理] オプションまたは [NIS ユーザの管理] オプションを選択します。LDAP ユーザ・テーブルまたは NIS ユーザ・テーブルに、既存のローカル・ユーザ・アカウントがすべてリストされます。次の手順に従ってユーザのエントリを変更します。

1. LDAP ユーザまたは NIS ユーザのリストをスクロールして、変更するユーザ・アカウントを選択します。
2. [修正...] を選択して、「LDAP ユーザの管理: ユーザの修正」ウィンドウまたは「NIS ユーザの管理: ユーザの修正」ダイアログ・ボックスを表示します。
3. 必要に応じて、データ・フィールドの値を変更します。
4. [了解] を選択して変更を確認します。パスワードの入力ミスの確認など、エラーがある場合は、ユーティリティからその修正を促すプロンプトが表示されます。

複数のアカウントを追加するには、[了解] の代わりに [適用] を選択します。加えたすべての変更は、[了解] を選択して作業を終了するまで有効になりません。

5. 「ローカル・ユーザ」ウィンドウがオープンし、変更が正常に行われたことを示す確認メッセージが表示されます。[了解] を選択して、SysMan Menu に戻ります。

オンライン・ヘルプは、フィールドとその有効なデータについて説明しています。

7.4.1.7 LDAP アカウントと NIS アカウントの削除

LDAP アカウントまたは NIS アカウントを削除するには、7.2.3 項の説明に従って、[LDAP ユーザの管理] オプション または [NIS ユーザの管理] オプションを選択します。LDAP ユーザ・テーブルまたは NIS ユーザ・テーブルに、既存のアカウントがすべてリストされます。

次の手順に従ってユーザを削除します。

1. ユーザのリストをスクロールして、削除するアカウントを選択します。
2. [削除...] オプションを選択して、「LDAP ユーザの管理: ユーザの削除」ダイアログボックスまたは「NIS ユーザの管理: ユーザの削除」ダイアログボックスをオープンします。
3. ユーザのリソースを削除してディスク・スペースを解放する場合は、「ユーザのディレクトリとファイルを削除」を選択します。
4. [了解] を選択してアカウントを削除します。LDAP ユーザまたは NIS ユーザのリストが即座にアップデートされます。

7.4.2 アカウント・マネージャ (dxaccounts) の使用

7.2.2 項のクイック・スタートの説明に従って、アカウント・マネージャ GUI (dxaccounts) を起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。表 7-2 に示したワークシートで収集したデータを使用して、以下の手順に従ってアカウントを管理します。

アカウント・マネージャ GUI を使用する際は、以下の手順に従ってアカウントを追加、変更、または削除します。これらの手順は、NIS データベースを変更する権限が必要になること以外は、NIS ユーザの管理手順と同じです。(NIS についての詳細は、『ネットワーク管理ガイド: サービス編』を参照してください。)

大半のオプションは、ユーザ・アカウント・データベースに影響を及ぼすため、root の権限が必要です。データベースに影響を及ぼさないオプションは、すべてのユーザが使用できます。このようなオプションの例としては、アカウントを検索するための Find オプションがあります。

ASU がインストールされている場合は、追加のオプションが「dxaccounts」ウィンドウに表示されます。このウィンドウを使用して、Windows NT ドメインのアカウントを管理すると同時に、対応する UNIX アカウントを作成す

ることができます。ASU についての詳細は、『*Advanced Server for UNIX* インストレーション/管理ガイド』を参照してください。

エンハnst (C2) セキュリティが使用可能な場合は、追加のオプションによって、有効なセキュリティの設定に従いアカウントを廃止し使用不能にできます。詳細は、『セキュリティ管理ガイド』を参照してください。

7.4.2.1 アカウントの追加と変更

「アカウント・マネージャ: <host>」ウィンドウを使用すると、次の手順でユーザ・アカウントの追加や変更ができます。

- 新しいアカウントを作成するには、[追加] を選択します。
- 既存のアカウントを変更するには、ユーザのアイコンをダブルクリックします。アカウントが多数ある場合、7.4.2.3 項に説明されているオプションを使用してアカウントを見つけることができます。
- また、既存のアカウントから新しいアカウントをコピー (クローニング) することもできます (7.4.2.4 項を参照)。

アカウントの追加や変更を行うには、次の手順を使用します。

1. 現在の表示が「ローカル・ユーザ」でない場合は、[表示] プルダウン・メニューを表示して [ローカル・ユーザ] オプションを選択します。
2. [追加] を選択して「ローカル・ユーザの追加/修正」ダイアログボックスをオープンし、[追加] を選択します。

既存のアカウントを変更する場合は、ユーザのアイコンをダブルクリックします。

3. 「ユーザ名」フィールドに新しいユーザ名を入力します。
4. 使用可能な次の UID を選択するか、新しい UID を入力します。

ユーザの UID をアカウント・マネージャで変更しても、ユーザのファイルおよびサブディレクトリの所有権は変更されません。また、場合によっては、ホーム・ディレクトリの所有権も変更されないことがあります。たとえば、ユーザ johndoe の UID を 200 から 201 に変更しても、そのホーム・ディレクトリ下のファイルおよびサブディレクトリは、UID 200 に属したままです。さらに、johndoe が自分のホーム・ディレクトリを所有していない場合は、そのディレクトリの所有権も変更され

ません。この問題を回避するには、`chown` コマンドを使用して、該当するディレクトリおよびファイルの所有権を変更します。

5. プルダウン・メニューを使用してプライマリ (1 次) グループを選択するか、またはテキスト・フィールドをクリアしてグループ名を入力します。
セカンダリ (2 次) グループを指定する場合は、[セカンダリ・グループ...] を選択します。「セカンダリ・グループ」ウィンドウで、目的のローカル・グループまたは NIS グループ (使用可能な場合) をダブルクリックします。
6. プルダウン・メニューで使用するシェルを選択します。
7. ホーム・ディレクトリが省略時のディレクトリ `/usr/users/<username>` に作成されます。必要に応じて、代替パスを入力します。
8. [パスワード...] を選択して初期パスワードを入力します。ローカル・セキュリティの設定で定義された長さの文字列を大文字と小文字または英数字を混在させて入力します。
9. コメント・フィールド (GECOS フィールド・データ) のユーザ情報を入力します。
10. 次のボックスをチェックできます。
「ホームディレクトリを作成する」
正しい所有者と保護モードでディレクトリを作成します。
「アカウントをロックする」
このボックスをクリアするまで、ユーザはログインできません。
11. [了解] を選択してアカウントを作成して、アカウント・マネージャのメイン・ウィンドウに戻ります。パスワードの入力ミスの確認など、エラーがある場合は、ユーティリティからその修正を促すプロンプトが表示されます。

現在のビューは、新規ユーザのアイコンでアップデートされます。

7.4.2.2 アカountの削除

7.2.2 項の説明に従って、`dxaccounts` ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初に表示されます。次の手順でアカウントを削除します。

1. 削除するアカウントのアイコンをダブルクリックします。多数のアカウントがある場合は、7.4.2.3 項の説明のオプションを使用して、アカウントを見つけます。
2. [削除] を選択します。「ローカル UNIX ユーザの削除」ウィンドウがオープンします。このウィンドウで、ユーザのファイルおよびディレクトリを削除できます。ファイルおよびディレクトリをアーカイブする場合は、`dxarchiver` オプションを参照してください。
3. [了解] を選択して削除を確認し、「アカウント・マネージャ: <host>」ウィンドウに戻ります。ユーザ・アカウントの削除が即座にこのウィンドウに反映されます。

7.4.2.3 アカountの検索と選択

`dxaccounts` ユーティリティには、ユーザ・アカウントを探すのに役立つ検索機能があります。この機能を使用して、ユーザ・シェルまたはパスワードなどの変更をグローバルに適用するユーザのグループを選択することもできます。

7.2.2 項の説明に従って、`dxaccounts` ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。次の手順に従って検索します。

1. [検索] を選択します。
2. いずれかのフィールドに検索文字列 (テキスト文字列) を入力して、[了解] を選択します。

[検索] オプションは、検索フィールドに検索文字列が含まれているアカウントをすべて見つけて表示します。次に例を示します。

1. 「ユーザ名」フィールドに文字列 `ad` を入力して、[了解] を選択します。
2. 「選択されたユーザ」ウィンドウがオープンし、検索条件に一致したユーザがリストされます。

3. 一致したユーザには、`adm`、`admin`、`adamK` および `wadmanB` が含まれます。これらのユーザ・アカウントは、現在のビューで強調表示されます。

ユーザ・アカウントのグループを選択してから変更または削除オプションを選択すると、選択したユーザに対してグローバル操作を実行できます。

7.4.2.4 アカウントのコピー

既存のアカウントをテンプレートとして使用して新しいアカウントを作成し、アカウントのプロパティをクローニングすることができます。1 つ以上のアカウントの正確な複製を、次の手順で作成できます。

1. 既存のユーザ・アカウントのアイコンを選択して強調表示するか、マウスを使用してアカウントのグループを選択します。
2. [了解] を選択して、アカウントをコピーします。
3. [貼付け] を選択して、クローン・アカウントを作成します。元の名前に文字列 `_copy n` が付加されて、新しいアイコン・ラベルがつけられます。ここで、 n は、コピーの順序番号です。必要な数だけコピーを作成できます。
4. 複製されたアカウントを順に選択し、名前とプロパティを変更します (7.4.2.1 項を参照)。
5. 次のような必要最小限の変更をアカウントに行います。
 - a. 新しいユーザ名を入力する
 - b. UID を変更するか、次に利用できる UID を選択する
 - c. パスワードを変更する
6. [了解] を選択して、変更されたアカウントを追加し、「アカウント・マネージャ: <host>」ウィンドウに戻ります。新規アカウントのアイコンが即座にこのウィンドウに反映されます。

同じ手順で、グループのクローニングもできます。

カット・アンド・ペーストまたはドラッグ・アンド・ドロップを使用してユーザ・アカウントをコピーする場合、「一般オプション」ダイアログ・ボックスの「重複したユーザ ID を許可する」オプションが動作に関係します。たとえば、UID が 200 のユーザ・アカウントのコピーを作成する場合、

「重複したユーザ ID を許可する」チェックボックスがオフ (省略時の値) であれば、一意の UID がコピー結果に自動的に設定されます。「重複したユーザ ID を許可する」チェックボックスがオンであれば、コピーは同じ UID を持ちます。同じ規則が、グループのコピーにも適用されます。

MB1 を使用してユーザ・アカウント、グループ、またはテンプレートのドラッグ・アンド・ドロップを行うと、移動操作ではなく、コピー操作になります。この動作は、MB1 でドラッグ・アンド・ドロップの移動操作が行われ、Shift-MB1 でコピー操作が行われる、CDE の省略時の動作とは異なります。たとえば、MB1 を使用してユーザ・アカウントを「Local Users」ビューからドラッグし、「NIS Users」ビューにドロップすると、ユーザ・アカウントのコピーが NIS 内に作成されます。この問題を回避するには、コピーの完了後に、オリジナルのアイコンを削除してください。

7.4.2.5 password オプションの使用

`dxaccounts` には、単一ユーザまたは複数のユーザのパスワードを変更または削除できるオプションがあります。このオプションは、次のように使用します。

1. パスワードを変更する 1 つのユーザまたは複数のユーザを選択します。ユーザのグループの選択には、検索オプションが役立ちます。
2. [編集] メニューで [パスワード] を選択します。
3. 「新しいパスワード」ウィンドウで、新しいパスワードを入力してそれを確認します。

[パスワード無し] を選択して、現在のパスワードを削除します。このオプションを選択した場合、システムのセキュリティに重要な影響があります。

4. [了解] を選択して変更を確認し、アカウント・マネージャのメイン・ウィンドウに戻ります。

7.4.2.6 アカウント・マネージャ (`dxaccounts`) の一般オプション

アカウント・マネージャ GUI (`dxaccounts`) を使用して、新たに作成されたアカウントの省略値を容易に設定することができます。また、コマンド行 (`useradd`) を使用してアカウントの省略時の値を設定することもできます

が、SysMan Menuの [アカウント] オプションを使用して省略時の値を設定することはできません。次の手順に従って省略値を追加または変更します。

1. [オプション] メニューで、[一般...] を選択します。「一般オプション」ウィンドウがオープンします。このウィンドウで、次の省略値を設定することができます。

「重複した ID の方針」

ユーザ識別子 (UID) およびグループ識別子 (GID) の重複が可能になります。

「ID 範囲の方針」

UID および GID の最小値、次に割り当てられる値、および最大値が設定できます。

「デフォルトのプライマリ・グループ」

省略時の 1 次グループを `users` 以外のグループに設定することができます。

「デフォルトのホームディレクトリ」

省略時のホーム・ディレクトリを `/usr/users` 以外のディレクトリに設定することができます。

「ユーザのデフォルトのシェル」

省略時のログイン・シェルを設定できます。

「デフォルトのスケルトン・ディレクトリ」

省略時のスケルトン・ディレクトリ・パスを `/usr/skel` 以外のディレクトリに設定することができます。

「ハッシュされたパスワード・データベースを使用する」

ハッシュ (暗号化された) パスワード・データベースの作成を強制します。

「新規アカウントにはパスワードを要求する」

新しく作成したすべてのアカウントにパスワードの入力を強制します。

「UNIX と PC のアカウントを一致させる」

UNIX アカウントの作成時にその関連アカウントの自動作成を強制します。

2. 必要な変更を行った後に、[了解] をクリックし、省略値をアップデートして「アカウント・マネージャ」のメイン・ウィンドウに戻ります。

7.5 グループの管理

以降の各項では、次のグループ管理の方法について説明します。

- SysMan Menu の次の [アカウント] オプションを使用したグループの管理
 - ローカル・グループの管理
 - NIS グループの管理
 - LDAP グループの管理
- アカウント・マネージャ GUI (dxaccounts) を使用した、グループの管理

また、`groupadd`、`groupmod`、および `groupdel` コマンドを使用してグループを管理することもできます。 コマンド行オプションについての詳細は、7.1.3 項で説明されているドキュメントを参照してください。

注意

`addgroup` ユーティリティは、指定できないオプションがあるため、またセキュリティを詳細に設定できないため、使用しないでください。

システム・ファイルの整合性を維持するために、ユーザ・アカウントは手動で追加しないでください。

7.5.1 SysMan Menu のグループ管理オプションの使用

以降の各項では、SysMan Menu のオプションを使用してグループを管理する方法について説明します。 次の作業について説明しています。

- 新しいローカル・グループ、LDAP グループ、または NIS グループの作成

- 既存のローカル・グループ、LDAP グループ、または NIS グループの変更
- ローカル・グループ、LDAP グループ、または NIS グループの削除

SysMan Menu 画面のフィールドにデータをキーボードから入力する方法については、オンライン・ヘルプを参照してください。

7.5.1.1 グループ情報の収集

グループの管理を始める前に、表 7-4 に示したワークシートを使って情報を収集します。エンハンスド・セキュリティ機能を使用している場合は、各データ項目が、最低限の必要条件を満たしている必要があるので注意してください。詳細については、『セキュリティ管理ガイド』を参照してください。

group ファイルのデータ項目の説明については、7.3.4 項を参照してください。SysMan Menu のオプションでは、NIS グループの省略時の値を指定することもできます。NIS の構成については、『ネットワーク管理ガイド：サービス編』を参照してください。

表 7-4 で、O と記されたデータ項目はオプションです。少なくとも 1 つのユーザ・アカウントを指定しなければなりません。

表 7-4: グループ管理ワークシート

| フィールド | 説明 | データ項目 |
|----------------|------------------------------|-------|
| グループ名* | | |
| パスワード* | 現在は使用されていない | |
| グループ識別子 (GID)* | 使用されていない場合は、次の番号が自動的に割り当てられる | |
| ユーザ* | | |
| ユーザ | | |
| ユーザ | | |
| ユーザ | | |
| ユーザ | | |
| ユーザ | | |

* は必須フィールドであることを示しています。

7.5.1.2 グループの作成と変更

新たにグループを作成するには、7.2.3 項の説明に従って、SysMan Menu を表示して [ローカル・グループの管理] オプションを選択します。ローカル・グループ・テーブルが表示され、既存のローカル・グループがすべてリストされます。NIS グループを追加する手順は、[NIS グループの管理] オプションを選択すること以外は同じです。

次の手順に従ってグループを作成します。

1. [追加...] を選択して、「グループの追加」ダイアログ・ボックスをオープンします。
2. 表 7-4 のワークシートの情報を使用して、データ・フィールドを入力します。
3. 必要に応じて、「メンバ」パネルで、新規グループの初期メンバとなるユーザの名前を強調表示します。
4. [了解] を選択して新規グループを追加します。エラーがある場合には、ユーティリティによってその修正を促すプロンプトが表示されます。

5. ローカル・グループ・テーブルのダイアログ・ボックスがオープンし、追加が正常に行われたことを示す確認メッセージが表示されます。[了解] を選択して SysMan Menu に戻ります。

既存のグループを変更するには、7.2.3 項に説明されているように、SysMan Menu を表示して [ローカル・グループの管理] オプションを選択します。ローカル・グループ・テーブルが表示され、既存のローカル・グループがすべて表示されます。次の手順に従ってグループ・エントリを変更します。

1. グループのリストをスクロールして、変更するグループを選択します。
2. [修正...] を選択して、「ローカル・グループの管理: グループの修正」ウィンドウをオープンします。
3. 必要に応じて、データ・フィールドの値を変更します。たとえば、ユーザのリストをスクロールして、新しいユーザをそのグループに追加することができます。
4. [了解] を選択して変更を確認します。

複数のグループを変更するには、[了解] の代わりに、[適用] を選択します。加えたすべての変更は、[了解] を選択して作業を終了するまで有効になりません。

5. 「ローカル・グループ」ウィンドウがオープンし、変更が正常に行われたことを示す確認メッセージが表示されます。[了解] を選択して SysMan Menu に戻ります。

オンライン・ヘルプには、フィールドとその有効なデータについての説明があります。

7.5.2 アカウント・マネージャ (dxaccounts) の使用

7.2.2 項に説明されているように、アカウント・マネージャ (dxaccounts) ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。表 7-4 のワークシートのデータを使用して、以降の各項の手順に従って dxaccounts を使用してグループを追加、変更または削除します。NIS グループの管理手順は、NIS データベースを変更する権限が必要になること以外は、ローカル・グループの管理手順と同じです。[検索] オプションなど、データベースに影響を及ぼさないオプションについては、すべてのユーザが使用できます。

システムに多数のグループがある場合は、7.5.2.4 項で説明されている [検索] オプションを使用して、変更または削除するグループを見つけてください。

LDAP グループは、アカウント・マネージャでは管理できません。

7.5.2.1 グループの追加

次の手順に従ってグループを追加します。

1. [表示] プルダウン・メニューで [ローカル・グループ] オプションを選択します。
2. [追加] ボタンを選択して、「ローカル UNIX グループの追加/修正」ウィンドウをオープンします。
3. 「ユーザ名」フィールドに新しいグループ名を入力します。
4. 使用可能な次の GID を選択するか、新しい GID を入力します。
5. ユーザ名をダブルクリックして、そのユーザをグループに追加します。この動作は、オプションです。
6. [了解] を選択してグループを追加して、「アカウント・マネージャ: <host>」ウィンドウに戻ります。新規グループのアイコンが即座にこのウィンドウに反映されます。

新しいグループを追加する別の方法としては、次のようにして既存のグループからクローンを作成する方法があります。

1. 既存のグループ・アイコンを選択して、そのグループを強調表示します。
2. [複写] を選択して、グループをコピーします。
3. [貼付け] を選択して、新しいバージョンのグループを作成します。元の名前に文字列 `_copy n` が付加されて、新しいアイコン・ラベルがつけられます。ここで、 n は、コピーの順序番号です。必要な数だけコピーを作成できます。
4. 新たにコピーされたアイコンをダブルクリックしてそれを強調表示し、「ローカル UNIX グループの追加/修正」ウィンドウを表示します。[修正] が自動的に選択されます。
5. グループへの次のような必要な変更を行います。
 - 新しいグループ名を入力する

- GID を変更するか、次に使用可能な GID を選択する
 - メンバを追加または削除する
6. [了解] を選択してグループを追加して、「アカウント・マネージャ: <host>」ウィンドウに戻ります。新規グループのアイコンが即座にこのウィンドウに反映されます。

7.5.2.2 グループの変更

7.2.2 項に説明されているように `dxaccounts` ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。次の手順を使用して、グループに変更を加えます。

1. 変更するグループをダブルクリックします。「ローカル UNIX グループの追加/修正」ウィンドウがオープンします。
2. グループへの次のような必要な変更を行います。
 - グループ名を変更する
 - GID を変更する
 - メンバを追加または削除する
3. [了解] を選択して変更を確認して「アカウント・マネージャ: <host>」ウィンドウに戻ります。グループへの変更が即座にこのウィンドウに反映されます。

7.5.2.3 グループの削除

7.2.2 項に説明されているように `dxaccounts` ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。次の手順に従ってグループを削除します。

1. 削除するグループを選択します。
2. [削除] を選択します。ユーティリティによって、そのグループの削除を確認するプロンプトが表示されます。
3. [了解] を選択して削除を確認して「アカウント・マネージャ: <host>」ウィンドウに戻ります。グループの削除が即座にこのウィンドウに反映されます。

7.5.2.4 グループの検索

アカウント・マネージャ・ユーティリティ (dxaccounts) を使用すると、グループおよびグループのメンバであるユーザを探することができます。

7.2.2 項に説明されているように dxaccounts ユーティリティを起動します。「アカウント・マネージャ: <host>」ウィンドウが最初にオープンします。グループを検索するには、次の手順に従います。

1. [検索] を選択します。
2. 次のいずれかの検索文字列を入力します。

グループ名または名前の一部 (テキスト文字列)

[検索] オプションは、指定された文字列を含むグループ名のグループをすべて選択して表示します。たとえば、文字列 mem はグループ mem および kmem に一致します。

GID (整数)

入力した数値は文字列として扱われます。[検索] オプションは、この文字列を含む GID のグループをすべて選択して表示します。たとえば、文字列 20 は、グループ 20 および 220 に一致します。

ユーザ名 (テキスト文字列)

[検索] オプションは、この文字列を含むユーザ名のユーザをすべて選択して表示します。たとえば、文字列 wal は、wallyB および cadwalz という名前のユーザを含むグループと一致します。

7.6 Windows ドメイン・アカウントとグループの管理

ASU (Advanced Server for UNIX) が実行されている場合は、アカウント管理ユーティリティを構成して、Windows ドメイン・アカウントの作成および管理をサポートすることができます。ASU のインストールおよび構成については、ASU の『*Advanced Server for UNIX* インストール/管理ガイド』を参照してください。ASU がインストールされている場合、アカウント管理ユーティリティを使用して関連 (同期化された) アカウントに対して特定の操作を行うことができます。これらのアカウントは、Windows ドメインと UNIX 環境の両方に存在する同じユーザのアカウントです。これらのアカ

ントを、UNIX ユーティリティでは、同期化されたアカウントと呼びます。Windows 2000 に特有の情報については、7.6.2 項を参照してください。

UNIX システムを構成して、関連する Windows NT ドメイン・アカウントと UNIX アカウントを作成したり、アカウントの省略時の作成オプションを設定するには、例 7-1 に説明されているように `usermod` (または `useradd`) コマンドを実行して、アカウント環境変数を設定する必要があります。

注意

ASU がインストールされて構成されている場合は、関連する Windows NT ドメイン・アカウントおよび UNIX アカウントの作成が省略時の設定で有効になります。すべてのアカウント管理ユーティリティの PC サポート機能は有効になります。Synchronized UNIX/PC Accts 環境変数の値は、設定がオンであることを示す 1 になります。

例 7-1: `usermod` による省略時の環境変数の変更

```
# usermod -D [1]

Local                = 1
Distributed          = 0
Minimum User ID      = 12
Next User ID         = 200
Maximum User ID      = 4294967293
Duplicate User ID    = 0
Use Hashed Database  = 0
Max Groups Per User  = 32
Base Home Directory  = /usr/users [2]
Administrative Lock   = 1
Primary Group        = users
Skeleton Directory   = /usr/skel
Shell                = /bin/sh
Synchronized UNIX/PC Accts = 0
PC Minimum Password Length = 0
PC Minimum Password Age = 0
PC Maximum Password Age = 42
PC Password Uniqueness = 0
PC Force Logoff After = Never

# usermod -D -x pc_synchronize=1 pc_passwd_uniqueness=1 pc_max_passwd_age=60 [3]

# usermod -D
.
.
.
Synchronized UNIX/PC Accts = 1
PC Minimum Password Length = 0
PC Minimum Password Age = 0 [4]
PC Maximum Password Age = 60
PC Password Uniqueness = 1
```

例 7-1: usermod による省略時の環境変数の変更 (続き)

PC Force Logoff After = Never

- ❶ このコマンドは、現在の省略時の環境変数を表示します。
- ❷ usermod コマンドの出力結果には、環境変数の省略値がリストされます。アカウントの作成時にこれらの省略値が新しいアカウントに割り当てられます。たとえば、新しいアカウントはすべて、ベース・ホーム・ディレクトリ /usr/users に作成されます。
- ❸ このコマンドでは、Windows NT ドメイン・アカウントにのみ適用される 3 つの環境変数に、新しい省略値を指定しています。
- ❹ このリストは途中で省略されていますが、次のような、環境変数の新しい省略値を表示しています。

pc_synchronize=1

ASU が実行されている場合に関連する Windows NT ドメイン・アカウントと UNIX アカウントを作成します。

pc_passwd_uniqueness=1

パスワードの独自性チェックを強制します。

pc_max_passwd_age=60

パスワードの最大有効日数。この日数を超える前に、ユーザはパスワードを変更しなければなりません。

groupmod -D コマンドを使用して、新しいグループを作成する際の省略時の環境変数を設定することができます。新しいアカウントを作成するときに、代替りの値を環境変数に指定して、省略値より優先させることもできます。詳細については、useradd(8)、usermod(8)、userdel(8) を参照してください。

コマンド行のプロンプトで、コマンドの最後に -h を入力して、さまざまなコマンド・オプションを示すヘルプ画面を表示することもできます。ASU User Manager for Domains では、省略時のポリシーを編集するときと同じような作業を行います。これにより、新たに作成されるアカウントに同じ省略時の環境変数を設定することができます。

ASU アカウント管理ユーティリティを使用して UNIX アカウントに対してのみ操作を実行したり、UNIX ユーティリティを使用して Windows NT ドメイン・アカウントに対してのみ操作を実行することはできません。以降の各項では、ASU が実行されている場合と同期化されたアカウントを管理

する場合の UNIX アカウント管理機能および ASU アカウント管理機能の動作について説明します。

7.6.1 同期化されたアカウントの管理

ASU をセットアップして同期化されたアカウントの作成を構成した場合は、アカウント管理ユーティリティの特定の機能が自動的に有効になります。以降の各項では、これらの機能がさまざまなアカウント管理ユーティリティでどのように使用されるかについて説明します。

ロック・ファイルによって、異なる 2 つのユーティリティ (または、同じユーティリティの 2 つのインスタンス) が同時に使用されるのを防止します。多数の管理者が多数のアカウントを管理する大規模な環境では、このような状況がよく起こります。このロック・ファイルは、`/etc/.AM_is_running` に作成されます。ロック・ファイルが存在する場合は、1 つのプロセスだけがユーザ・データおよびグループ・データに関連するシステム・ファイルにアクセスできます。UNIX アカウント管理マネージャの別のインスタンスを実行しようとする、データ・ファイルがロックされていることを示すメッセージが表示されます。

ASU ユーティリティを使用してアカウントを追加する場合は、ASU はロック・ファイルの存在を検出します。ロック・ファイルがあると、関連する UNIX アカウントを作成することはできません。Windows NT ドメイン・アカウントだけを作成します。ロック・ファイルに関するエラー・メッセージは表示されず、関連アカウントが作成されなかったという確認は表示されません。ASU ツールを使用する場合は、`/etc/passwd` ファイルを検査して関連する UNIX アカウントが作成されたか確認する必要があります。

7.6.1.1 SysMan Menu のアカウント・オプションとグループ・オプションの使用

SysMan Menu [アカウント] ユーティリティのユーザ・インタフェースは、ASU が実行されている場合と見た目に違いはありません。同期化されたアカウントが使用可能になっている場合、ウィンドウと画面には違いが生じません。ただし、次の動作上の相違点に注意してください。

| | |
|--------|---|
| ユーザの追加 | アカウント作成の一部として、アカウント所有者をグループに割り当てるときに、いくつかの DOS----グループから選択できます (Primary Group オプション)。 |
|--------|---|

例 7-1 の説明に従って、関連する Windows NT ドメイン・アカウントの作成を有効にした場合は、その関連アカウントが自動的に作成されます。この自動作成を無効にすることはできません。

ユーザの削除 関連する Windows NT ドメイン・アカウントが自動的に削除されます。この削除を無効にすることはできません。ユーザの Windows NT ドメイン・アカウントを保持したい場合は、この操作を実行しないでください。

グループの追加/削除 いくつかの DOS---- グループが選択リストに含まれ、lanman および lmxadmin などの、省略時の Windows NT ドメイン・アカウントが示されます。

SysMan Menu の使用については、第 1 章 を参照してください。

7.6.1.2 アカウント・マネージャ (dxaccounts) の使用

アカウント・マネージャ・ユーティリティ (dxaccounts) は、X11 準拠の GUI であり、CDE などの X-window ユーザ環境でのみ使用できます。dxaccounts のメイン・ウィンドウには、**PC** (Windows NT ドメイン) アカウントを作成するオプションがあります。このオプションは、ASU が実行されていない場合は、淡色表示になり使用することはできません。ASU が実行されている場合は、次の機能が使用できます。

- Windows NT ドメインなどの 1 つのユーザ環境でアカウントを作成する場合、UNIX 環境などの別のユーザ環境での同期化されたアカウントの作成を選択できます。
- 例 7-1 で示されているように、関連する Windows NT ドメイン・アカウントまたは UNIX アカウントの作成が省略時で有効になっている場合でも、この関連アカウントを作成しないように選択することができます。
- [表示] メニューに追加のオプションが表示され、すべての Windows NT ドメイン・アカウントおよびグループのオープンが可能になります。これらの追加オプションを選択した場合は、PC (Windows NT ドメイン) ユーザ・アカウントおよびグループ・アカウントのアイコンがそれぞれ

表示されます。これにより、PC アカウントまたはグループを UNIX アカウントと同じ方法で、追加、変更または削除できます。

- [オプション] メニューで [PC デフォルト] オプションを使用して特性を設定することができます。この特性は、新たに作成されるすべてのアカウントに継承されます。[一般オプション] メニュー項目を使用して、アカウントの同期化を設定し、また、UNIX アカウントの特性を設定することができます。
- [削除] でアカウントを削除する場合は削除するアカウントを、UNIX アカウント、PC アカウント、あるいはその両方から選択するように促されます。
- [表示] メニューで [ローカル・グループ] オプションを使用する場合は、PC グループ (DOS----) が表示され、それらのグループに対して管理作業を行うことができます。
- [表示] メニューで [PC グループ] オプションを使用する場合は、PC ドメイン・グループが表示され、それらのグループに対して管理作業を行うことができます。

7.5.2 項 に説明されている手順を使用して、PC アカウントおよびグループに対して管理作業を行うことができます。

`dxaccounts` を使用する利点は、本来の X11 アプリケーションであるため、ドラッグ・アンド・ドロップやカット・アンド・ペーストのアイコン操作などのウィンドウ環境機能で、新しいユーザ・アカウントやグループを既存のものから簡単に作れることです。ただしこれらは、ポータビリティのある SysMan Menu のアカウント・ユーティリティと違って、CDE などの X-window ユーザ環境だけで実行できます。

LDAP グループは、アカウント・マネージャ・ユーティリティでは管理できません。

7.6.1.3 コマンド行ユーティリティの使用

ユーザ・アカウントおよびグループ・アカウントを管理するコマンド行ユーティリティを使用して、例 7-1 に示したように、省略時のアカウント特性を設定することもできます。これらの特性は新たに作成されたすべてのアカウントに適用されるもので、ASU ユーティリティではアカウント・ポリシと呼ばれます。グラフィカルなユーティリティとは異なり、コマンドを使用する

と、省略時の環境変数を無効にして、カスタマイズした値を新しいアカウントに指定することができます。

ASU がインストールされている場合は、次のアカウントおよびグループ作成オプションが使用できます。

- `useradd`, `usermod -D` オプションを使用して、省略時の Windows NT ドメイン・アカウントの特性を設定できます。この拡張オプションとその省略値を次に示します。

`pc_synchronize=` (値: 1, オン)

新しいアカウントを、Windows NT ドメイン・アカウントまたは UNIX システム・アカウントのどちらかとして作成するときに、特に指定しなくても同期化されたアカウントを作成するかどうかを指定します。この値が 0 (ゼロ) の場合は、同期化されたアカウントは作成されません。

`pc_min_password_age=` (値: 0, オフ)

パスワードの変更が可能になるまでの日数を指定します。ユーザは、この日数よりも頻繁にパスワードを変更することはできません。

`pc_max_password_age=` (値: 42 日)

このオプションでは、パスワードの有効日数を指定します。ユーザは、少なくともこの日数を超える時点でパスワードを変更する必要があります。

`pc_passwd_uniqueness=` (値: 0, オフ)

ユーザが指定したパスワードを検査して、同じパスワードを使用しないようにします。

`pc_force_logoff=` (値: Never, オフ)

アカウントの有効期限が過ぎた場合に、自動的にアカウント所有者をログアウトさせる一時的なアカウントを設定します。

例 7-1 に示したように、`-D -x` オプションを使用してこれらの拡張オプションを設定します。省略時の特性を無効にするには、アカウントの

作成などのアカウント管理操作で次のように `-x` フラグを使用して、拡張オプションを指定します。

```
# useradd -x pc_passwd_uniqueness=1 guest9
```

次のコマンド・オプションは拡張オプションではないので、省略時のアカウントの特性は設定しません。これらのアカウントの特性は、ASU ユーティリティを使用して設定することもできます。これらのコマンド・オプションは、新しいアカウントを追加する際に使用します。

- `pc_username=name_string`

Windows NT ドメインでのユーザ・アカウント名。このアカウント名は、ユーザの UNIX アカウントの名前と同じにすることも、別の名前にすることもできます。

- `pc_unix_username=login_name`

同期化された UNIX アカウント名。名前が指定されない場合は、Windows NT ドメインのアカウント名と同じ名前になります。

- `pc_fullname=text_string`

ユーザのフルネームまたはアカウントの説明。

- `pc_comment=text_string`

アカウントの簡単な説明。この説明は、管理者だけが変更できます。

- `pc_usercomment=text_string`

アカウントの簡単な説明。この文字列はユーザが変更できます。

- `pc_homedir=pathname`

Windows NT と共通の形式で指定する、ユーザのホーム・ディレクトリのパス。

- `pc_primary_group=group`

ユーザが属する 1 次グループ (Windows NT ドメイン)。

- `pc_secondary_groups=group,group....`

ユーザが属する 2 次 Windows NT ドメイン。この値は、コンマで区切ったリストで指定します。

- `pc_logon_workstations=client_name`

ユーザがログオンできるクライアント・ホスト・システムのリスト。この値はコンマで区切ったリストで指定します。ヌル値 (" ") は、ユーザがすべてのワークステーションからログオンできることを示します。

- `pc_logon_script=pathname`
省略時のログオン・スクリプトが存在するディレクトリ。このディレクトリは、ASU の構成時に作成されます。
- `pc_account_type=local/global`
Windows NT ドメインでアカウントがローカルかグローバルかを指定します。
- `pc_account_expiration=date_string`
アカウントの有効期限が過ぎ、ログインができなくなる日付。
- `pc_logon_hours=Dd0000-0000,Dd0000-0000....`
ログインの有効期間を曜日と時刻で指定し、ログインを許可するかどうかを指定します。
- `pc_user_profile_path=pathname`
省略時のユーザ・プロファイル・ディレクトリのパス名。
- `pc_disable_account=0|1`
初期状態で、アカウントをロックして、ログインできなくするかどうかを指定します。
- `pc_passwd0|1`
アカウントの初期パスワードとして使用されるテキスト文字列。このオプションは、`-x` フラグを使って指定する必要があります。このオプションを指定すると、パスワードの入力を促すプロンプトが表示されるので、パスワードを入力して、この入力を確認する必要があります。入力したパスワードは表示されません。
- `pc_passwd_choose_own=0|1`
ユーザが自分のパスワードを設定できるようにするかどうかを指定します。
- `pc_passwd_change_required=0|1`
最初のログイン時にユーザにパスワードを変更するように強制します。

- `userdel` – このコマンドで利用できる PC (Windows NT ドメイン) オプションは、`Synchronized UNIX/PC Accts` だけです。

このオプションを使用して、次のように同期化されたアカウントを削除します。

```
# userdel -r -x pc_synchronize=1 studentB
```

- `groupadd`, `groupmod`

次の拡張オプションを `-x` フラグを使用して、Windows NT ドメインのグループを管理することができます。

```
pc_group_description=string
```

グループを説明するテキスト文字列。

```
pc_group_members=user,user....
```

コマンドで区切られたグループ・メンバのリスト。

コマンド行を使用する利点は、これを用いて管理作業を完全に制御することができることです。すべてのコマンド・オプションを指定でき、また、アカウントの省略時の環境変数より優先させることができます。

コマンドを、シェル・スクリプトで使用してアカウントの作成をカスタマイズし、自動化することができますが、コマンド・オプションが長くなる場合は、かえって、グラフィカル・ユーティリティを使用してアカウントを設定する方が便利です。

`useradd(8)` と `groupadd(8)` およびここで示したそれらに関連するリファレンス・ページを参照してください。

7.6.1.4 ASU ドメイン・ユーザ・マネージャの使用

ASU には、Windows NT ドメイン、ドメイン・ユーザ・アカウントおよびグループを管理するユーティリティがあります。このアプリケーションは、Windows NT を実行しているシステムにインストールされていなければなりません。また、このアプリケーションは、Windows NT を実行しているシステムでのみ使用でき、`net` コマンド行オプションと同じ機能を提供します。

新たに作成されるすべてのアカウントに適用される、省略時の環境変数を指定できます。これらの環境変数は、Windows NT ドメインではアカウント・ポリシーと呼ばれます。ドメイン・ユーザ・マネージャ (`usrmgr.exe`) を使用

している場合は、同期化された UNIX アカウントに対して省略時の環境変数を設定することはできません。

詳細については、ASU の『*Advanced Server for UNIX* インストレーション/管理ガイド』、およびドメイン・ユーザ・マネージャのオンライン・ヘルプを参照してください。

7.6.1.5 ASU net コマンドの使用

ASU はさまざまな net コマンドのセットを提供しています。このコマンドは、UNIX コマンド行または Windows NT サーバの DOS ウィンドウで入力します。

たとえば、次のコマンドはユーザ・アカウントの追加、変更または削除に使用する net user コマンドのヘルプを表示します。

```
# net help user | more
```

The syntax of this command is:

```
NET USER [username [password | \*] [options]]
          username [password | \*] /ADD [options]
          username [/DELETE]
.
.
.
# net user josef /add
```

net コマンド・オプションのリストを表示するには、次のコマンドを実行します。

```
# net help view
```

net コマンドの使用方法の詳細は、『*Advanced Server for UNIX* インストレーション/管理ガイド』を参照してください。

7.6.2 Windows 2000 シングル・サイン・オン

UNIX サーバと Windows 2000 クライアント・システムからなるローカル・コンピューティング環境では、環境内に 1 台以上のドメイン・コントローラがある場合、オプションの Windows 2000 シングル・サイン・オン (SSO) ソフトウェアを構成できます。SSO ソフトウェアを使用すると、Windows 2000 ドメインのアカウント保持者は、UNIX アカウントなしで、UNIX サーバのコンピューティング・リソースにアクセスすることができます。

SSO ソフトウェアは、Windows Active Directory と、関連する Windows アカウント管理ユーティリティを変更します。この変更により、Windows 2000 ドメインの管理者は、UNIX のアカウント情報をユーザの Windows 2000 のアカウント・レコードに記録できるようになります。UNIX サーバ・システムは、アカウント保持者のデータに安全にアクセスでき、アカウント保持者の UNIX ログイン情報 (パスワードや GID など) を読み取ることができます。

また、同じソフトウェアや管理ツールを使用する SSO ユーザ・グループを作成することもできます。

7.6.2.1 シングル・サイン・オンのインストール要件

この機能を構成して使用するには、次のインストール要件を満たす必要があります。

- UNIX システムヘルート・アクセスができ、SSO ソフトウェアをインストールするすべての Windows 2000 ドメイン・コントローラの管理者でなければなりません。UNIX システム上と、少なくとも 1 つのドメイン・コントローラ上で、インストール・プロシージャを実行しなければなりません。
- UNIX システムでは、C2 レベルのセキュリティは実行できません。セキュリティ・レベルについての詳細は、『セキュリティ管理ガイド』を参照してください。
- 「*Associated Products Volume 2*」CD-ROM が必要です。この CD-ROM には、SSO ソフトウェア・キットが収められています。このキットの /doc ディレクトリには、『*Windows 2000 Single Sign On Installation and Administration Guide*』があります。
- 次の情報が必要です。
 - ドメイン名 (sso.w2k.com など)
 - ドメイン・コントローラのホスト名 (w2kserv.sso.w2k.com など)
 - 特権ドメイン・アカウントのアカウント名とパスワード。このアカウントは、Administrators グループに属していて管理者の特権を持っていなければなりません。メインの Administrator アカウント以外でなければなりません。このようなアカウントがない場合は、作成してからインストールを開始してください。

7.6.2.2 シングル・サイン・オン・ソフトウェアのインストール

ソフトウェアを次のようにインストールします。

1. CD-ROM を CD-ROM ドライブにセットします。
2. 次のようなコマンドを使用して、マウント・ポイントを作成し CD-ROM をマウントします。

```
# mkdir /apcd  
# mount -r /dev/disk/cdrom4c /apcd
```

3. 次のコマンドで、インストレーション・キットとドキュメントを見つけてみます。

```
# ls /apcd/Windows2000_SSO
```

4. `setld` コマンドを使用して、`W2KSS0100` という名前のソフトウェア・サブセットをインストールします。インストレーションの完了時に、構成スクリプト `/usr/sbin/w2ksetup` が自動的に実行されます。『*Windows 2000 Single Sign-On Installation and Administration Guide*』に説明されているように、構成を完了させます。

7.6.2.3 シングル・サイン・オン・アカウントの作成についての UNIX の要件

UNIX アカウントの特性に対する次の要件が、SSO アカウントに適用されます。

- SSO ユーザ・アカウントは、標準の Windows 2000 ユーザ管理ツールの変更されたバージョンを使用した場合のみ Windows 2000 ユーザ環境に作成できます。SSO アカウントは、`dxaccounts` や `useradd` などの UNIX ツールを使用して作成することはできません。

既存の Windows 2000 アカウントをアップグレードして、アカウント保持者に UNIX リソースへの SSO 特権を付加することもできます。

- UNIX アカウントと Windows 2000 アカウントには、用語の違いがあります。たとえば、アカウントの特性を示すユーザ・アカウント・データは、Windows 2000 ではプロパティ、UNIX オペレーティング・システムでは属性 (attribute) と呼ばれます。この情報は、UNIX 環境では GECOS データと呼ばれます。一部の UNIX コマンドやユーティリティは、このデータを使用して、アカウントの操作や、ユーザの識別を行います。UNIX のアカウント属性については、7.3.3 項と以降の節を参照してください。

ユーザやグループごとに、次のアカウント・データを用意します。必要であれば、この章で説明している UNIX アカウント管理ツールを使用して、各ユーザのアカウント・データが正しいフォーマットであり重複していないことを確認します。

ユーザ名

Windows 2000 では、ユーザ名はユーザ・ログオン名です。SSO では、この名前が 2 つの要件(長さ、重複しないこと)を満たさなければなりません。この条件は、グループ名にも適用されます。

Windows 2000 は非常に長いユーザ名をサポートしていますが、実際には大半のユーザは、覚えやすく入力しやすい、名前の短縮形やイニシャルを使用します。ユーザ名の最大長は、UNIX 環境での一般的な制限により、8 文字となります。

実際の名前はユーザのイニシャルのように短くすることができますが、どちらのシステム上でも重複しない名前であればなりません。ユーザ名が `chs` という UNIX アカウントのみを持つユーザがいる場合は、この名前を SSO アカウントとして割り当てることはできません。

パスワード

各ユーザにはパスワードが必要です。パスワードの長さは、UNIX システムの現在の設定により決まります。この設定は、使用されているセキュリティ・メカニズムにより異なることがあります。詳細については、『セキュリティ管理ガイド』を参照してください。

UID および GID

各アカウントには、UID という識別用の重複しない整数が必要です。また、各グループには、GID があります。これらの識別子については、7.3.2 項および 7.3.4 項を参照してください。

ユーザ・コメント

このフィールドを使用すると、後で参照するときに役立つように、テキストで GECOS データの説明を入力することができます。

ホーム・ディレクトリ

UNIX 環境でのユーザのホーム・ディレクトリは、Windows 2000 システムでのディスク・シェアと同じ意味です。ホーム・ディレクトリは、UNIX ファイル・システム `/usr` の一部をユーザ・アカウント用に予約したものです。一般的に、ホーム・ディレクトリのパス名には、ユーザのアカウント名を使用します。たとえば、`/usr/staff/songch` や `/usr/users/chs` です。

シェル

ユーザがログオンしたときに起動される、ユーザの省略時の UNIX コマンド環境です。UNIX コマンド環境には、Bourne シェル (sh) や Korn シェル (ksh) があります。詳細については、`shells(4)` のリファレンス・ページおよび 7.3.1 項を参照してください。

7.6.2.4 シングル・サイン・オンのアカウントおよびグループの作成

7.6.2.3 項で用意した情報を使用して、次のように SSO アカウントを作成します。

1. Windows 2000 ドメイン・コントローラに、管理者のアカウントでログインします。
2. Microsoft Management Console (MMC) インタフェースを起動し、「Active Directory ユーザーとコンピュータ」ウィンドウをオープンします。
3. 「Users」フォルダをオープンし、既存のユーザを選択するか、アクション・メニューをオープンし、[新規作成] オプションを選択してから [ユーザー] オプションを選択します。
4. 3つのダイアログ・ボックスが、連続してオープンします。新規ユーザ・アカウントごとに、次の情報の入力が必要です。
 - 名前などの、ユーザ・アカウントの詳細
 - アカウントの初期パスワードと、パスワードの特性
 - UNIX アカウント・プロパティ。UID や GID など、7.6.2.3 項で説明している情報を使用します。

SSO グループを作成するには、同じ手順を使用し、手順 3 で [新規作成] と [グループ] メニュー・オプションを選択します。

7.6.2.5 シングル・サイン・オンのシステム・ファイル

ソフトウェアのインストールおよび構成時には、次のシステム・ファイルが作成されます。

- ドメイン・サーバ上のアカウント情報のレジストリへのコネクションである、`ldapcd` デーモン。このデーモンが誤って抹消されたり停止された場合は、次のコマンドを使用して再起動してください。

```
# /sbin/init/dldapw2k restart
```

- `ldapcd` デーモンの設定が格納されている `/etc/ldapcd.conf` 構成ファイル
- 特定のユーザにだけ UNIX 認証を強制する `/etc/w2kusers.deny` 構成ファイル

これらのファイルについての詳細は、ファイル・ヘッダおよび『*Windows 2000 Single Sign-On Installation and Administration Guide*』を参照してください。



プリント・サービスの管理

この章では、プリント・サービス用の情報取得と、その設定方法および管理方法について説明します。プリント・サービスの設定および管理は、オペレーティング・システムの新規インストールまたは新バージョンへのアップグレードの直後に行うことも、後で行うこともできます。プリント・サービスを管理するために使用するファイルとユーティリティについては、この章で説明します。

インストールがすべて終わり、システムの初期構成を行っている間に、「システム・セットアップ」というタイトルのチェック・リストが表示されます。このメニューには、[プリンタの設定] というオプションがあります。このオプションを選択すると、`printconfig` グラフィカル・ユーザ・インタフェースを実行します。詳細については、8.3.2 項を参照してください。

この章では、次の項目について説明します。

- 異なる構成方法および設定ユーティリティの説明を含む、管理タスクの概要 (8.1 節)
- プリンタ構成に必要な情報取得の概要 (8.2 節)
- `printconfig` および `lprsetup` ユーティリティの説明を含む、物理的に接続されたプリンタ、リモート・プリンタ、およびネットワーク・プリンタの設定用のさまざまなユーティリティの使用方法 (8.3 節)
- プリンタの追加や削除、また、プリント・ジョブの制御といったプリント・システムの日常保守 (8.4 節)
- システム・ファイルの構造、スプーリング、デーモン、エラー報告、およびプリント・フィルタなどの高度なトピックに関する参照情報 (8.5 節)
- 一部のプリント・フィルタを使う上での現在の制限事項 (8.6 節)
- プリンタの試験や問題解決に関する情報 (8.7 節)

8.1 プリント管理タスク

プリント管理タスクには、プリント・システムの設定と、プリント・システムの管理という2つのカテゴリがあります。

プリント・システムを最初に設定するためには、次の処理を行います。

- プリンタをシステムに接続する。物理的に接続するか、ネットワークを通してアクセスします。
- そのプリンタについての情報を `/etc/printcap` ファイルに追加する。
- 必要なデバイス・ファイルおよびスプール・ディレクトリを作成する。
- プリント・デーモン `lpd` を起動する。
- 必要に応じて、プリンタの課金機能を管理する。
- プリンタのインストレーションを確認し、テスト・プリントを実行する。

プリンタを設定した後、システムで動作させるには、以下のことを行う必要があります。

- 新規プリンタの追加や既存プリンタの特性変更といった日常管理を実行する。
- 必要に応じて、プリント・キューおよびファイルを管理する。
- プリント・ジョブの日常操作やスルーブットを管理する。

以上の操作を行う際に使用するツールについては、8.1.6 項を参照してください。

8.1.1 プリンタの接続方法

プリンタのインストールと接続には、ローカル・システムの構成に応じて、以下のようないくつかの方法があります。

| | |
|----------|--|
| ネットワーク接続 | ローカル・エリア・ネットワーク (LAN)、ローカル・エリア・トランスポート (LAT)、または TCP/IP を介しての共用ネットワーク・プリンタへの接続。 |
| 直接接続 | 1 台の物理的に接続されたプリンタをローカルにインストールする最も簡単なオプション。システムの後面には、1 台のプリンタをケーブルで接続する、シリアルまたはパラレルのハードウェア・ポートがあり |

8-2 プリント・サービスの管理

ます。ハードウェアのインストールについては、プリンタの説明書に書かれています。

ローカル・システム上のユーザは誰でもプリンタにアクセスすることができます。

リモート接続

ネットワーク上の別のシステムに直接接続されたプリンタへの接続。

リモート・オプションを選択したシステムが、プリンタの接続されているシステムにリモートからアクセスしそのサービスを受けられるようになっている必要があります。

PC ネットワーク接続

Advanced Server for UNIX (ASU) を使用している場合の、パーソナル・コンピュータ (PC) のプリント・キューへの接続。

PC と UNIX システム環境が混在している場合に、このアプリケーションを使用します。ASU がインストールされている場合は、PC プリント・キューを構成して PC クライアント間でプリンタを共有するためのオプションがさらにあります。

8.1.2 プリンタの管理方法

プリンタの管理には、いくつかの方法があります。それぞれに利点があります。

8.1.2.1 Printer Configuration ユーティリティ (`printconfig`) の使用

このユーティリティは、グラフィカル・ユーザ・インタフェースを備えており、初めてのユーザやクイック・セットアップにお勧めします。このユーティリティはシステム管理ツールの標準セットに含まれています。これらのツールの全般的な情報については、第 1 章を参照してください。

ASU がインストールされている場合は、`printconfig` を使って PC プリンタを管理することもできます。

オペレーティング・システムには、非常に多くの他社製プリンタをサポートするドライバおよび構成ファイルが用意されています。`printconfig`

ユーティリティを使用すると、サポートしているプリンタの全リストが自動的に表示されるので、設定が迅速にできます。

注意

旧版の `/etc/printcap` ファイルと一緒に `printconfig` を使用する場合は、制約があります。詳細は、`printcap(4)` を参照してください。

8.1.2.2 `lprsetup` ユーティリティの使用

端末から実行するこのコマンド行ユーティリティは、以前のリリースとの互換性があります。

`lprsetup` ユーティリティは `printconfig` と同様のタスクを実行しますが、ASU がインストールされていても PC プリンタはサポートしません。`printconfig` のコマンド行オプションを呼び出すと、`lprsetup` ユーティリティが起動されます。

`lprsetup` ユーティリティは、ASU (Advanced Server for UNIX) のもとでは、ASU 自身に PC キューの管理機能があるにもかかわらず、PC プリンタの管理をサポートしません。

8.1.2.3 手動によるシステム・ファイルの編集

経験を積んだシステム管理者の場合、`/etc/printcap` ファイルを直接編集してプリンタを管理することがあります。たとえば、特定のプリンタ構成を多くのシステムにクローニングする場合や、あるシステムの `printcap` ファイルの一部を別のシステムのファイルとマージするような場合です。この章にある参照情報は、このような作業に役に立ちます。個々のプリント・キューのエントリを管理するには、`lprsetup` ユーティリティまたは `printconfig` ユーティリティの使用をお勧めします。

注意

`/var/spool/lpd` ファイルは、TruCluster Server 環境で使用される特殊リンクになっています。スプール・ディレクトリとしては使用できません。ファイルを手動で編集する場合は、この

リンクを壊さないようにしてください。第 6 章の CDSL に関する項を参照してください。

8.1.3 Advanced Printing Software

Advanced Printing Software は、「Associated Products」CD-ROM で提供される、オプションのサブセットです。このソフトウェアについては、『インストレーション・ガイド』を参照してください。Advanced Printing Software ソフトウェアをインストールしてある場合は、Advanced Printing Software およびプリント・デーモン `lpd` を同じシステム上で実行するために、8.3.2 項の説明に従ってゲートウェイを構成する必要があります。Advanced Printing Software は、`lpd` を無効にして実行させることもできますが、その場合は LPD の代わりにリモートからの入力プリント要求 (ソケット 515 上) をすべて受けます。ただし、この構成では `lpr` のようなローカルの `lpd` コマンドは動作しません。

8.1.4 関連ドキュメント

プリンタ構成ツールの使用に関する追加ドキュメントが、マニュアル、リファレンス・ページ、およびオンライン・ヘルプにあります。

8.1.4.1 マニュアル

Tru64 UNIX オペレーティング・システムのドキュメント・セットにある、プリンタ構成ツールの使用についての情報を、次の一覧に示します。

- 『ネットワーク管理ガイド：接続編』には、ネットワーク・プリンタが使用する接続についての情報が記載されています。
- 『国際化ソフトウェア・プログラミング・ガイド』には、アジア地域の言語のサポートのように、ローカル言語機能を持つプリンタを対象とした国際化サポートの情報が記載されています。
- 『*Common Desktop Environment*: ユーザーズ・ガイド』および『*Common Desktop Environment*: 上級ユーザ及びシステム管理者ガイド』には、共通デスクトップ環境でのプリンタ・サービスの設定に関する情報が記載されています。

他の Tru64 UNIX ドキュメントとしては、PC プリント・キューを管理する ASU 機能について説明している『*Advanced Server for UNIX* インストール／管理ガイド』があります。

Advanced Printing のドキュメントには、Tru64 UNIX オペレーティング・システムのオプションの構成要素である Advanced Printing Software を使用するための情報があります。『*Advanced Printing Software User Guide*』には、コマンド行インタフェースを使用して、ジョブをプリンタにサブミットする方法が説明されています。また、ローカル・プリント環境の設定方法と、サブミットしたジョブの監視方法も説明されています。このマニュアルは、以下のドキュメントとともに、ソフトウェア・キットに含まれています。

- 『*Advanced Printing Software System Administration and Operation Guide*』
- 『*Advanced Printing Software Command Reference Guide*』
- 『*Advanced Printing Software Release Notes*』
- 『*Advanced Printing Software Installation Guide*』

プリンタのインストールやデータ伝送(ポー)レートなどの、ソフトウェアに設定する必要がある情報については、プリンタのマニュアルも参照してください。構成しようとしているプリンタが、サポート対象デバイスのリストに載っていない場合は、必ずそのプリンタのマニュアルを参照してください。プリンタのマニュアルには、トレイの選択など、プリンタの特殊機能を使用するために構成ユーティリティに指定しなければならない情報が説明されています。通常、インストール・ユーティリティを使用した場合(あるいは、`/usr/sbin/lprsetup` ディレクトリを見ると)、プリンタはサポートされているデバイスのリストに入っています。プリンタがこのディレクトリ内のファイルで定義されておらず、製造元もプリンタを使用するのに必要な情報を提供していない場合は、構成ツールが提供している汎用設定を使用してください。ただし、汎用設定を使用すると、プリンタ機能の使用が制限される場合があります。

8.1.4.2 リファレンス・ページ

ここに示したリファレンス・ページは、ユーティリティ、ファイル、およびデーモンに関する詳細情報を説明しています。

| | |
|---------------------|--|
| <code>lpc(8)</code> | ライン・プリンタ・システムの動作の制御。 |
| | <code>/etc/printcap</code> ファイルで構成されている各ライ |

| | |
|---|--|
| | <p>ン・プリンタに対し、プリンタを使用可能または使用不能にする、プリンタ・スプーリング・キューを有効または無効にする、スプーリング・キュー内のジョブの順序を整理する、またはプリンタや、対応するスプーリング・キュー、プリンタ・デーモンのステータスを調べるために、<code>lpc</code> コマンドを使用することができます。</p> |
| <code>ports(7)</code> | <p>システムにプリンタを接続する際に使用するプリンタ・ポートの情報、および、それらプリンタ・ポートと <code>/dev</code> 内のプリンタ・デバイス特殊ファイル名との対応。</p> |
| <code>printconfig(8)</code> および <code>lprsetup(8)</code> | <p>構成ツールとそのコマンド行オプションに関する情報。</p> |
| <code>printcap(4)</code> および <code>lprsetup.dat(4)</code> | <p>プリンタの構成情報が入っているシステム・ファイルに関する情報。</p> |
| <code>wwpsmf(8)</code> , <code>pcfof(8)</code> | <p>汎用プリント・フィルタに関する情報。</p> <p><code>wwpsmf(8)</code> のリファレンス・ページは、ローカル言語での PostScript のプリントをサポートする PostScript プリンタ用の汎用国際化プリント・フィルタについて説明しています。</p> <p><code>pcfof(8)</code> のリファレンス・ページは、ANSI, PCL (Printer Control Language), および PostScript プリンタ用の汎用プリント・フィルタについて説明しています。</p> |
| <code>lpd(8)</code> | <p>プリント・デーモンに関する情報。</p> |
| <code>latcp(8)</code> | <p>ローカル・エリア・トランスポート (LAT) 制御ユーティリティに関する情報。このユーティリティは、ネットワーク・プリンタを使っている場合にだけ使用するもので、プリント・サービスのようなサービスをホストに追加するのに使用します。</p> |

| | |
|--|--|
| lpr(1), pr(1), lprm(1), lpq(1), お よび lpstat(1) | ファイルのプリントに使用するコマンドの説明。これらのコマンドの使用については、『 <i>Tru64 UNIX ユーザーズ・ガイド</i> 』を参照してください。 |
| lpctest(8) | プリンタのテスト・パターン・ユーティリティの説明。 |
| ppdof(8) | PostScript プリント・フィルタ用のテキストの説明。 |
| services(4) | /etc/services ファイルのフォーマットの説明。 TCP/IP を使うプリントに対して定義するサービス・ポートをこのファイルで定義します。 |

8.1.4.3 オンライン・ヘルプ

Printer Configuration アプリケーション (printconfig) には、オンライン・ヘルプのボリュームがあって使用方法を調べることができます。

lprsetup ユーティリティには、コマンド行ヘルプがあります。

8.1.5 システム・ファイル

次のシステム・ファイルには、プリンタの構成情報が入っています。ファイルの中には、/var/spool のように省略時設定つまり最初から UNIX で決まっているものがあります。ファイル名とロケーションは、好きなように決めることができます。

/etc/printcap 構成済プリンタのデータが入っています。

/usr/sbin/lprsetup /usr/sbin/lprsetup ディレクトリには、サポートされている (既知の) プリンタに対する一連の構成データ (*.lpd) が入っています。各ファイルの名前は、プリンタ名に対応しています。この情報は、使用条件に合わせてプリンタをインストールして構成すると、/etc/lprsetup.dat ファイルとしてコンパイルされ、その後 /etc/printcap ファイルに転送されます。

| | |
|--------------------------------|--|
| <code>/etc/lprsetup.dat</code> | <code>/etc/lprsetup.dat</code> ファイルは、 <code>/usr/sbin/lprsetup</code> ディレクトリ内のファイルを コンパイルした結果です。 |
| <code>/usr/spool</code> | このファイルは、 <code>/var/spool</code> (後述参照) を指す シンボリック・リンクです。このシンボリック・ リンクは、 <code>mail</code> または <code>uucp</code> などの古いプログラ ムのために用意されています。 |
| <code>/var/spool</code> | このディレクトリは、省略時ディレクトリで、プ リント・ジョブは、プリントが終わるまで一時的 にここに保存されます。スプール・キューは、 <code>/etc/printcap</code> に定義されている各デバイスの <code>sd</code> エントリで識別されます。 新規作成するスプール・ディレクトリは、 <code>/var/spool</code> 下に置いてください。 <code>/var</code> は、可変 データ用に定義されています。 |
| <code>/var/spool/lpd</code> | このファイルは、 <code>/var/spool/cluster/mem- bers/{memb}/spool/lpd</code> を指すシンボリック・リ ンクです。これはクラスタ内のノード上のメンバ固 有のスプール・ディレクトリです。スタンドアロ ン・システムは、単一ノード・クラスタ (<code>member0</code>) と見なされます。ロック・ファイル <code>lpd.lock</code> は、 このディレクトリに格納されます。 |
| <code>/usr/adm/lpd*err</code> | <code>/usr/adm/lpd*err</code> ファイルは、インストールされ ている各プリンタのエラー・ログ・ファイルです。 エラー・ログ・ファイルは、エラー・ロギングが有 効になっている場合にだけ作成されます。 |
| <code>/var/adm</code> | <code>/var/adm</code> ディレクトリには、課金機能が有効 になっている場合、プリンタ課金ファイルが置 かれます。課金ファイル名のフォーマットは、 <code>/var/adm/<printer>acct_sum</code> です。ここで、 |

<printer> は、インストール時にプリンタへ割り当てた名前です。

/usr/sbin/lpd

/usr/sbin/lpd ファイルはライン・プリンタ・デーモンです。構成ファイルは、ディレクトリ /var/spool/* (または /usr/spool/*) にあります。

/dev

/dev ディレクトリには、ローカルの UNIX ソケット /dev/printer があります。このソケットは、lpd デーモンが作成したもので、親 lpd が動いている限り存在します。

8.1.6 関連ユーティリティ

次のユーティリティもプリンタ環境で使用できます。

lpc

このライン・プリンタ制御ユーティリティにより、プリント・キューの管理およびプリンタへのアクセス制御ができます。また、このユーティリティを使用して、構成エラーがないかプリンタ記述ファイルを調べることができます。詳細は、lpc(8) を参照してください。

pac

このプリンタ/プロッタ課金ユーティリティは、プリンタ課金ログ・ファイルのデータをフォーマットして表示したり、テキスト・ファイルとして保存したりします。詳細は、pac(8) を参照してください。課金機能については、acct(8) を参照してください。

Print Manager GUI

CDE の「アプリケーション・マネージャ — デスクトップアプリケーション」には、印刷マネージャのグラフィカル・ユーザ・インタフェースが含まれており、以下の操作が実行できます。プリント・キューの管理、プリンタへのアクセス制御、およびプリンタ・データのビューのカスタマイズを行うために、グラフィカル・ユーザ・インタフェース

を使用する方法については、オンライン・ヘルプを参照してください。

これらの機能は、ターミナルからコマンド行で実行する `lpc` ユーティリティおよび `lpq` コマンドまたは `lpstat` コマンドの機能とほぼ同じです。

Print Screen GUI

CDE の「アプリケーション・マネージャ — デスクトップアプリケーション」には、画面印刷グラフィカル・ユーザ・インタフェースが含まれており、画面の全体または一部を印刷したり、ファイルに保存することができます。詳細は、オンライン・ヘルプを参照してください。

CDE フロント・パネルのプリンタ・アイコン

CDE のフロント・パネルにあるプリンタ・アイコンにより、プリンタの選択とプリント・キューの管理をローカルで行うことができます。「プリンタの構成」(`printconfig`) は、CDE の「アプリケーション・マネージャ — システム設定」から実行することも、SysMan Menu または SysMan Station から起動することもできます。さらに後者のユーザ・インタフェースでは、グラフィカル・ツールをリモートから使用することも、PC のような異なるワークステーションまたは他の UNIX システムから使用することもできます。システムの省略時プリンタを CDE 環境に割り当てるために、`LPDEST` などの環境変数を設定する方法については、CDE のドキュメントを参照してください。

8.2 情報収集

プリンタを追加する前に、`lprsetup` または `printconfig` ユーティリティを使用するときに入力する、追加プリンタに関する情報を集める必要があります。必要な情報は、プリンタがリモートか、直接接続か、または LAT や TCP/IP を介したネットワーク接続かによって変わってきます。

システムがネットワークに接続されている場合は、ネットワーク・プリンタを追加またはアクセスするときに必要な情報について、ローカルのネッ

トワーク管理者に問い合わせるか『ネットワーク管理ガイド：接続編』と『ネットワーク管理ガイド：サービス編』を参照してください。

8.2.1 直接接続とネットワーク接続のプリンタ

直接接続または LAT や TCP/IP を介したネットワーク接続を行う場合は次の情報が必要になります。

使用可能なプリンタのタイプ (サポートされているプリンタ)

他社製プリンタを追加していた場合には、プリンタのタイプは `lprsetup.dat(4)` を参照するか、または `/usr/sbin/lprsetup` ディレクトリを参照して調べることができます。普通は、たとえば LN03 といったように、プリンタの表面に表示してあるプリンタの名前とほぼ同じですが、DECLaser 5100 のプリンタ・タイプが `ln09` であるように、プリンタ名とタイプが異なる場合があります。`printconfig` ユーティリティは、サポートされているデバイスをリストにして表示します。このとき、ASU がインストールされていれば、PC プリンタのオプションも一緒に表示します。

プリンタの別名 (代替名)

プリンタの別名 (代替名) — 1 台のプリンタに対して 1 つ以上の別名を割り当てることができます。別名はプリンタ・コマンドで使用することができます。たとえば、ローカル・システムを `alfie2` と名付けた場合、その名前を別名として割り当て、ファイルのプリントで次のようにその別名を使うことができます。

```
# lpr -Palfie2 prt_accounting.txt
```

接続タイプ

この情報は、プリンタをシステムに接続する方法によって以下のように変わります。

TCP ネットワーク接続されたプリント・サーバ・デバイス

直接接続 システム・ボックスの後面にあるポートに接続

LAT ローカル・エリア・トランスポート・ポートまたはサービスとして接続 (詳細は『ネットワーク管理ガイド：接続編』を参照)

デバイスのパス名

これは、接続タイプによって次のように変わります。

TCP 接続タイプが TCP/IP の場合は、TCP アドレスが必要です。次に示すように 2 つの形式がありますが、どちらでも構いません。

`@node/port`

プリンタ・ホスト (またはノード) 名の後ろに、ポート番号、またはサービス名を続けます。ポート番号には LN32 に対する 9100 のような番号を指定し、サービス名には `/etc/services` ファイルでこのポート用に定義されている名前を指定します。サービス名が定義されていない場合には、ポート番号を使用する必要があります。たとえば、`@alfie.nic.ccc.com/ln32port` は、`/etc/services` 内のエントリ `ln32port 9100/tcp` にマップされます。エントリ `@alfie.nic.ccc.com/9100` はポート番号を直接指定するので、`/etc/services` の中にサービス・エントリがなくても構いません。

`@tcp_address/port`

`nnn.nnn.nnn.nnn` 形式の TCP/IP アドレスの後ろに (LN32 に対する 9100 のような) ポート番号または `/etc/services` に定義されているサービス名を続けます。たとえば、`@123.321.123.321/9100` のように指定します。

直接接続 接続タイプが直接接続の場合は、このエントリには `/dev` ディレクトリ内のデバイス・ファイル名を指定します。たとえば、プリンタ・ケーブルを 1 または COMM1 のラベルが付いた 9-ピンのソケットに接続する場合、対応するデバイス特殊ファイルは、`/dev/tty00` です。

LAT 接続タイプが LAT の場合は、`printconfig` が省略時の `/dev/lat` ポート名またはサービス名、LAT サーバ・ノー

ド名，または LAT ポート名を指定してくれます (詳細は『ネットワーク管理ガイド：接続編』を参照)。

拡張オプション

これらのオプションのほとんどは基本的なインストレーションでは指定する必要がありません。省略時の値を受け入れれば済みます。しかし，プリンタによっては，オーナーズ・マニュアルで伝送ボー・レートのような特定の設定が要求されている場合もあります。伝送レートは，インストール時に選択したプリンタ・ケーブルの長さやタイプなどの特性に依存することがあります。

拡張オプションは，ファイル `/etc/printcap` ではシンボルとして設定されています。サポートされているシンボルおよび値は，`printcap(4)` でリストになって定義されています。`printconfig` を使用するときには，オンライン・ヘルプでシンボルの説明を見ることができます。

この画面では，スクロール・ダウンすることによって，使用可能なすべてのオプションと省略時のエントリを見ることができます。省略時のエントリは書き換えることができます。

よく使用されるのは次のオプションです。

| | |
|--------------------------------|--|
| 課金ファイル名 | プリンタ課金機能を使って印刷消耗品の消費状況を調べる場合，省略時の値を選択 (または希望の値を入力) してください。 |
| stty ボー・レート (ハード・ワイヤ・ポートのみ) | プリンタに対して使用する伝送レートが指定されている場合はここへ入力します。プリンタのスループットを上げるために省略時のレートを高くしていることがよくあります。 |
| 使用制限 | プリント・ジョブのボリュームと量を制限したい場合は，ここで制限値を指定します。 |
| 省略時のページ・レイアウト | ページ・レイアウト特性に対する省略時の値を変更する場合は，ここで特性値を指定します。サポートされているレイアウトは，プリンタの制限と特性に左右されることがあります。 |

す。プリンタのオーナーズ・マニュアルを参照してください。

デスティネーション・ディレクトリおよびファイル

プリント・スプールまたはエラー出力用に独自のディレクトリとファイルを使いたい場合は、そのロケーションをここで指定します。

8.2.2 リモート・プリンタ

リモート・プリンタをインストールする際に必要な情報を、以下に示します。

プリンタの別名 (代替名)

1 台のプリンタに対して 1 つ以上の別名を割り当てることができます。別名はプリンタ・コマンドで使用することができます。たとえば、リモート・システムを `alfabet` と名付けた場合、その名前を別名として割り当て、ファイルのプリントで次のようにその別名を使うことができます。

```
# lpr -Palfabet prt_accounting.txt
```

リモート・システムの名前

`alfabet.ccc.nic.com` のようなリモート・システムのホスト名。

リモート・プリンタの名前

`lp0` のようなリモート・システム上のプリンタ名、または有効な別名。

拡張オプション

これらのオプションは基本的なインストレーションでは指定する必要がありません。リモート・プリントで構成 (またはエラー・ログなどを構成から除外) できる拡張オプションは限られています。詳細については、8.2.1 項を参照してください。ほとんどの拡張オプションは、リモート・システムには適用されません。

リモート・プリントの設定で選択するプリンタ・タイプは `Generic_Remote_LPD` です。

8.3 プリンタの構成

以降の項では、`printconfig` ユーティリティを使用してコンピュータにプリンタを接続する際に必要となる情報について説明します。プリンタの構成に `lprsetup` ユーティリティを使用する方法を説明している項もあります。

先に進む前に、プリンタがシステムに物理的に接続されていること、リモート・プリントの場合はネットワークからアクセスできること、そしてオーナーズ・マニュアルの説明に従って機能していることを確認してください。インストール時の問題を避けるには、構成ユーティリティが示す省略時データをそのまま受け入れるのが得策です。プリンタが動作するようになれば、拡張オプションを調べ、必要なら、同じユーティリティを使って構成をチューニングすることができます。

Printer Support Environment サブセットは、事前にインストールされている必要があります。このサブセットがインストールされているかどうかを確認するには、次のコマンドを実行します。

```
# setld -i | grep OSFPRINT
```

OSFPRINT がインストールされている場合は、次のような情報が表示されます。

```
OSFPRINT540      installed      Local Printer Support (Printing Environment)
```

OSFPRINT サブセットがインストールされていない場合は、『インストール・ガイド』を参照して `setld` ユーティリティを使って追加してください。

この節で説明するプリンタ構成の項目は、以下のとおりです。

- `printconfig` を使用した TCP/IP プリントの構成 (8.3.1 項)
- `printconfig` を使用した直接接続のプリンタのインストール (8.3.2 項)
- `printconfig` を使用したリモート・プリンタの設定 (8.3.3 項)
- `printconfig` を使用した PC プリント・キューの構成 (8.3.4 項)
- `lprsetup` を使用したプリンタのインストール (8.3.5 項)

`printconfig` は CDE またはコマンド行から起動できます。詳細は、`printconfig(8)` を参照してください。図 1-1 に示すように、「プリンタ設定」メイン・ウィンドウが表示されます。最初に表示されるのは、「プリンタ設定: *host name*」というタイトルの付いたメイン・ウィンドウです。

図 8-1 は、Printer Configuration (printconfig) のメイン・ウィンドウを示しています。

図 8-1: Printconfig のメイン・ウィンドウ



8.3.1 printconfig を使用した TCP/IP プリントの構成

TCP/IP プリントを使用すると、ネットワークに接続されているリモート・プリンタにプリント・ジョブをサブミットすることができます。ただし、この機能を使用するためには、プリンタにネットワーク・インタフェース・カードが搭載されているか、プリント・サーバまたはターミナル・サーバに接続されている必要があります。また、そのカードを TCP/IP ノード名とノード・アドレスで登録する必要があります。

TCP/IP プリントでは、ローカル・ホストは、直接接続されたプリンタのプリント・ジョブを管理するのと同様に、プリント・ジョブを管理します。唯一の違いは、TCP/IP プリントでは、ローカル・プリント・デーモン (lpd) が

TCP/IP を介してリモート・プリンタと通信を行うことです (LAT プリントと同様)。各プリンタは、ネットワーク・インタフェース・ハードウェアに指定されているソケット番号、またはプリンタ・コンソールからユーザが定義したソケット番号上で接続要求をリスンします。

ネットワークに接続されている 1 台のプリンタに対して、複数のホストがこのように通信できますが、ホストからの要求は先着順にサービスされます。したがって、TCP/IP プリントはリモート・プリントとは異なります。リモート・プリントでは、リモート・ホストまたはリモート・プリンタはリモート・サイトのプリント・キューを管理して、ソケット 515 (/etc/services の `printer` 用のエントリで定義) でネットワーク接続をリスンします。

8.3.1.1 `printconfig` を使用した TCP/IP プリンタの構成

TCP/IP プリントを使うと、ホスト・デバイスとしてネットワークに接続されているリモート・プリンタにジョブを送ることができます。TCP/IP プリントについては、8.3.1 項を参照してください。また、8.3 節で説明されているように、情報を収集してください。

これまでの項で説明したように `printconfig` を起動します。

このウィンドウでは、次のように、収集したデータを使用してプリンタ・タイプを選択します。

- Compaq LN32 を選択します。
- [設定] オプションを選択します。

このプリンタは物理的に接続されたプリンタとして扱われるので、次に「プリンタ設定: ローカル・プリンタ設定」ウィンドウが表示されます。lp4 のような、次に使用可能なプリンタ名が表示されます。以下のデータをフィールドに入力します。

- プリンタの別名。たとえば、ローカル・ホスト名、または LN32_office23 のようにプリンタのタイプや物理的なロケーションを識別しやすい名前にします。

この値は `@nodename/servicename` のように指定することもできます。ここで、`servicename` は、`/etc/services` で定義されている名前であり、プリンタの TCP/IP ポートと結び付けられています。

- 接続タイプ。TCP です。

- 最後に、デバイスのパス名をポート番号またはサービスとして指定する必要があります。この例では、プリンタのネットワーク・アドレスで、`@alfie.nic.ccc.com/9100` または `@123.321.123.321/9100` のようになります。
- [コミット] を選択してこのオプションを `/etc/printcap` ファイルに書き込みます。

基本的な TCP/IP プリンタの構成は以上で終わりです。プリント・ジョブを制限するような拡張オプションを使う場合は、[詳細] オプションを選択して「プリンタ設定: ローカル・プリンタ設定: 詳細」ウィンドウを開きます。

構成を確認すると、「プリンタ設定: *host name*」ウィンドウに戻ります。そこには、設定したプリンタが構成済みプリンタとしてリストに追加されているのが見られます。テスト・ページをプリンタにプリントさせるには、[テスト] オプションを使います。プリントできない場合は、データをよくチェックしトラブルシューティングの節を参照してください。

プリンタには TCP/IP アドレスの構成が必要です。そのためには、ハードウェア・コンソール・パネルでエントリを作成するか、`telnet` または Web ブラウザを使用して、プリンタと通信する必要があります。詳細は、プリンタのマニュアルを参照してください。

「プリンタ設定: *host name*」ウィンドウにある残りのオプションは [標準に設定] で、これを使うと、システムの省略時プリンタとして、任意の構成済みプリンタを指定できます。プリント・キューの指定されていないプリント・ジョブは、すべてこの省略時プリンタに出力されます。

必要なプリンタを選択して [標準に設定] を選択します。現在の省略時プリンタが「標準プリンタ」フィールドに表示されます。

8.3.1.2 TCP/IP プrint の設定に必要な手動による追加手順

今までの項で説明した手順の他に、ローカル・ホストで TCP/IP プrint を設定するために、以下の手順を実行してください。

1. プリンタを設定します。ネットワーク・カードを持つ各プリンタに、TCP/IP アドレスおよびノード名を割り当てます。また、プリンタが接続要求をリッスンする TCP/IP ソケット番号も決定します。
`/etc/services` ファイルで定義されている名前を指定しても、また、プリンタに割り当てられているポート番号を直接指定しても構いません。

ん。サービス名を作成する場合は、ステップ 2b で `/etc/services` ファイルを編集する際にソケット番号が必要になります。

表 8-1 に 5 つの Compaq 製プリンタ、1 つの Lexmark 製プリンタ、そして 1 つの Hewlett-Packard 製プリンタ用のソケット番号を示します。

表 8-1: TCP/IP ソケット番号

| プリンタ | ソケット番号 |
|----------------------|--------|
| DEClaser 3500 (LN14) | 10001 |
| DEClaser 5100 (LN09) | 10001 |
| HP プリンタ | 9100 |
| Digital_LN17 | 2501 |
| Lexmark プリンタ | 9100 |
| Compaq LN16 | 9100 |
| Compaq LN32 | 9100 |

その他のプリンタのソケット番号を調べるには、プリンタのマニュアルでネットワーク・カードについての情報を参照してください。この番号をユーザが指定できるプリンタもあります。

2. ローカル・ホストを構成します。このステップでは、実行する必要があるユーティリティ、および TCP/IP プリントを構成するためにローカル・ホスト上で変更する必要があるファイルについて説明します。以下のタスクを実行するには、スーパーユーザ特権が必要です。
 - a. 次の値を `ct` および `lp` 変数に割り当てます。

```
ct=tcp
lp=@nodename/servicename
```

`nodename` には、ネットワークで使用するために登録したプリンタのノード名を指定します。`servicename` には、次のステップで `/etc/services` データベースに入力するために選択した名前、またはポート番号 (たとえば、`lp=myHPLaserjet/9100`) のいずれかを指定します。既存の `/etc/printcap` プリンタ・エントリを変更して TCP/IP プリントを使用したい場合は、`/etc/printcap` ファイルを編集し、`ct` および `lp` 変数の値を変更します。

制御変数 `xs`、`xc`、`fs`、および `fc` の値を削除することもできます。これらの変数は、シリアル・ポート・ドライバに関連した設

定を確立する制御変数ですが、ネットワーク・ソケット・ドライバでは無視されます。

- b. サービス・データベースを構成します。 `/etc/services` データベース・ファイルにサービス名と `tcp` ポート番号 (ソケット番号) を登録する必要があります。ステップ 1 でプリンタを構成したときに決定したソケット番号を入力し、その番号にユーザが選択したサービス名を関連付けてください。たとえば、DEC Laser 3500 用にサービス・データベースを構成する場合は、`/etc/services` ファイルに次の行を追加します。

```
declaser3500    10001/tcp
```

ユーザが定義した `declaser3500` という文字列がサービスを表しています。つまり、これは、ステップ 2a で `/etc/printcap` ファイルに `servicename` として入力した文字列と同じものです。変更内容を `/etc/printcap` ファイルに保存してください。

ステップ 2a でポート番号を指定した場合は、`/etc/services` ファイルは変更しないでください。

- c. リモート・ホスト・データベースを構成します。 `/etc/printcap` ファイルで `lp` 変数値の一部として指定した `nodename` 値をローカル・ホストのネットワーク管理サービスに認識させる必要があります。したがって、`nodename` およびそのネットワーク・アドレスを `/etc/hosts` データベース・ファイルに入力しなければなりません。リモート・ホスト名用に BIND サーバを実行していれば、必ずしも `/etc/hosts` ファイルにプリンタのノード名を追加する必要はありませんが、BIND サーバに問題が起こった場合、`/etc/hosts` にエントリがあればフォールバックとして役立ちます。

TCP/IP プリントは、一度構成すると、ローカルおよびリモート・プリントと同じように使用できます。コマンド行からは、プリンタのノード名、コマンド・オプション、およびファイル名を指定して、`lpr` コマンドを実行します。CDE のプリント・ユーティリティでも、プリンタの状態を見たりプリント・ジョブを登録したりできます。

8.3.2 `printconfig` を使用した直接接続のプリンタのインストール

この項では、`printconfig` ユーティリティを使用してプリンタをインストールする方法について説明します。例として、グラフィカル・ユーザ・イ

ンタフェースを使用して、DEClaser 5100 プリンタをローカルにインストールする方法を説明します。物理的な接続はすべて済んでおり、また必要な情報も収集してあるものとします。プリンタ構成の変更またはプリンタの削除にも、`printconfig` を使用することができます。このようなその他の処理については、8.4 節 に説明してあります。

注意

Version 3.2 以前が動作しているシステムで `/etc/printcap` ファイルを変更する場合は、`printconfig` コマンドを使用しないでください。古い `/etc/printcap` ファイルには互換性のない部分があり、`printconfig` コマンドがそのファイルを壊す可能性があります。代わりに `lprsetup` を使用してください。

(ハードウェアのインストール後に行う) プリンタの通常のインストールには、データ収集の時間も含めて 10 分 ~ 15 分 程度かかります。

プリンタを初めてインストールする場合には、省略時の値を使用することをお勧めします。必要なら、その後で `printconfig` を使用して構成を変更することもできます。

以前の項で説明したように、`printconfig` を起動します。

`printconfig` ユーティリティを実行するにはスーパーユーザの特権が必要です。追加するプリンタのタイプおよび指定する情報に従って、ユーティリティは以下のような処理を行います。

- ファイルの作成または既存ファイル `/etc/printcap` の編集
- スプール・ディレクトリの作成
- エラー・ログ・ファイルの作成
- 課金ファイルの作成
- デバイス特殊ファイルの作成
- 以前に選択した記号の変更の要求

`printconfig` ユーティリティを実行すると、「プリンタ設定: *host name*」というタイトルの付いたメイン・ウィンドウが最初に表示されます。このウィンドウで、8.2 節 の説明にあるように事前に収集しておいたデータを使って、プリンタのタイプを選択します。

- [Digital_DEClaser_5100] を選択します。

- [設定] オプションを選択します。

このプリンタはリモート・プリンタではないので、次に「プリンタ設定: ローカル・プリンタ設定」ウィンドウが表示されます。そして、次に使用可能なプリンタの名前が表示されます (これがこのシステムで最初に設定しようとするプリンタである場合は lp0)。このウィンドウで次のデータを入力します。

- プリンタの別名。たとえば、ローカル・ホストの名前、または local_DL5100 のようにプリンタ・タイプを識別しやすい名前にします。
- 接続タイプ。ケーブルを COMM1 のようなローカルのシリアル・ポートに接続しているので、直接接続のオプションを選択する必要があります。
- 最後にデバイスのパス名を指定します。これはシリアル・ポートに対応するデバイス特殊ファイルです。デバイス・ファイル /dev/tty00 は COMM1 ポートに対応します。
- [コミット] を選択してオプションを /etc/printcap ファイルに書き込みます。

基本的な構成の場合は、以上で終わりです。プリント・ジョブの制限を設定するような拡張オプションを使用する場合は、[詳細] オプションを選択して「プリンタ設定: ローカル・プリンタ設定: 詳細」ウィンドウを開きます。詳細については、8.5 節を参照してください。

構成を確認すると、「プリンタ設定: *host name*」ウィンドウに戻ります。そこには、設定したプリンタが構成済みプリンタとしてリストに追加されているのが見られます。テスト・ページをプリンタにプリントさせるには、[テスト] オプションを使います。プリントができない場合は、データをよくチェックするとともに、8.7 節を参照してください。

「プリンタ設定: *host name*」ウィンドウにある残りのオプションは [標準に設定] で、これを使うと、システムの省略時プリンタとして、任意の構成済みプリンタを指定できます。つまり、プリント・キューの指定されていないプリント・ジョブは、すべてこの省略時プリンタに出力されます。

必要なプリンタを選択して [標準に設定] を選択します。現在の省略時プリンタが「標準プリンタ」フィールドに表示されます。

以上で、プリンタは使用できるようになります。PostScript ファイルまたはカラー・グラフィック・ファイルのような適当なファイルをプリントして、プリンタの性能をテストしてください。8.1.6 項の説明にあるプリンタ・ユーティリティを使って、プリンタおよびキュー状態の確認ができるかどうかを確かめてください。

8.3.3 `printconfig` を使用したリモート・プリンタのセットアップ

リモート・プリンタは、リモート・ホストに直接接続されているプリンタ（リモート・ホストでローカルとして扱われているプリンタ）のことをいいます。リモート・プリンタは、そのプリンタのネットワーク・カードがリモート `lpd` プロトコルをエミュレートしていれば、ネットワークに直接接続することができます。その場合、プリンタは、プリンタが接続されたリモート・ホストのように見えます。

ローカル・プリント・キューを構成して、プリント・ジョブがネットワークを介してリモート・ホストに送られるようにします。そうすると、これらのジョブはリモート・ホストでプリントされるようになります。リモート・システムがサービスしているリモート・プリンタをセットアップする場合は、ローカル・システム（クライアント）がリモート・システム（ホスト）の `hosts.lpd` ファイルまたは `hosts.equiv` ファイルに登録されていなければなりません。

この操作を行なう前に、8.2 節を参照して必要なデータに関する情報を収集してください。次に、前述の項の説明に従って、`printconfig` を起動します（なお、リモート・キューは、その項の例で作成しているキューと同じです）。`printconfig` を使用してリモート・プリントを構成する方法を、次の例で説明します。

1. `[remote]` を選択します。
2. `[設定]` オプションを選択します。

注意

設定しようとしているプリンタはリモート・プリンタなので、「プリンタ設定: リモート・プリンタ設定」ウィンドウが表示されます。そして、次に使用可能なプリンタの名前が表

示されます (これがこのシステムで最初に設定しようとするプリンタである場合は `lp0`)。

3. プリンタの別名を入力します。たとえば、リモート・ホスト名とプリンタのタイプが分かるような名前にします。
4. リモート・システム名を入力します。
5. リモート・プリンタ名を入力します。たとえば、8.3.2 項では、プリンタ `lp0` をシステムに追加しました。
6. [コミット] を選択してオプションを `/etc/printcap` ファイルに書き込みます。

基本的な構成の場合は、以上で終わりです。プリント・ジョブの制限を設定するような拡張オプションを使用する場合は、[詳細] オプションを選択して「プリンタ設定: ローカル・プリンタ設定: 詳細」ウィンドウを開きます。プリンタをリモート・プリンタとして構成したので、拡張オプションとして指定できるのは、ローカル・ログ・ファイルやスプール・ディレクトリといったローカル処理に影響するオプションが少しあるだけです。

構成を確認すると、「プリンタ設定: *host name*」ウィンドウに戻ります。そこには、設定したプリンタが構成済みプリンタとしてリストに追加されているのが見られます。テスト・ページをプリンタにプリントするには、[テスト] オプションを使います。プリントできない場合は、データをよくチェックするとともに、8.7 節を参照してください。

8.3.4 `printconfig` による PC プリント・キューの構成

Advanced Server for UNIX (ASU) がインストールされ実行されている場合には、クライアント PC のプリンタ・キューを構成することができます。ASU にはプリント・キューの構成および管理機能もあります。ASU の詳細については、『*Advanced Server for UNIX* インストレーション/管理ガイド』を参照してください。PC クライアント用にプリンタの共用キューを作成する場合は、前もって、`/etc/printcap` ファイルに少なくとも 1 台のプリンタを構成しておく必要があります。

前述の方法に従って、`printconfig` を起動します。ASU のもとで `printconfig` を起動する場合、使用可能なプリンタ・タイプは、`Advanced_Server_Shared_Printer_Queue` です。このオプションを

選択すると、次に「プリンタ設定: Advanced Server Shared Print Queue Setting」というタイトルのウィンドウが開きます。このウィンドウには次の3つのオプションがあるだけです。

Advanced Server shared print queue name

psql のようなキュー名を入力します。

Printer devices

lp0,lp2 のようにデバイスのリストをコンマで区切って入力します。

Comment

キューの用途や制限事項についてのコメントや注記を入力します。

[OK] を選択してキューを作成し、printconfig のメイン・ウィンドウに戻ります。新しいキューが表示されます。

キューの状態を調べるには、次の ASU コマンドを使い、下の例に示すようなキュー名に対する出力を調べます。

```
# net share
.
.
.
Share name  Resource                Remark
-----
psql        lp0, lp2                    Spooled
```

他の ASU タスクを実行して、キューを PC システムで使用できるようにする必要があるかもしれません。詳細については、ASU のドキュメントを参照してください。

8.3.5 lprsetup によるプリンタのインストール

コンピュータに直接接続されているプリンタをローカルにインストールするには、lprsetup ユーティリティを使用します。lprsetup を使用してプリンタの構成を変更したり、プリンタを削除することもできます。これ以外の設定については、8.4 節を参照してください。

最初にプリンタをインストールする際には、省略時の値を使用することをお勧めします。

以降の例で使用するプリンタは LN03R です。

lprsetup プログラムは、端末ウィンドウでコマンド・プロンプトに対して /usr/sbin/lprsetup を入力することにより実行できます。lprsetup は、スーパーユーザで実行する必要があります。追加するプリンタのタイプや入力する情報に応じて、lprsetup は以下の処理を行います。

- 既存の /etc/printcap ファイルの編集または作成
- スプール・ディレクトリの作成
- エラー・ログ・ファイルの作成
- 課金ファイルの作成
- デバイス特殊ファイルの作成
- 選択したシンボルを変更するためのプロンプトの表示

lprsetup スクリプトを実行すると、まず、メイン・メニューに次のメッセージが表示されます。

```
# /usr/sbin/lprsetup
Tru64 UNIX Printer Setup Program
```

```
Command < add modify delete exit view quit help >:
```

lprsetup のコマンド・オプションを表 8-2 に示します。

表 8-2: lprsetup のオプション

| コマンド | 説明 |
|--------|--|
| add | プリンタを追加する。 |
| modify | 既存のプリンタの属性を変更する。 |
| delete | ユーザの構成から既存のプリンタを削除する。 |
| exit | lprsetup プログラムを終了する。 |
| view | 構成しているプリンタの現在の /etc/printcap ファイル・エントリを表示する。 |
| quit | lprsetup プログラムを終了する。 |
| help | lprsetup プログラムのオンライン・ヘルプを表示する。 |

コマンド・オプションはいずれも、最初の 1 文字に省略できます。

各プロンプトで情報を入力するか、または Return キーを押して省略時の情報を選択します (多くの場合、省略時の情報を入力します)。また、?

(疑問符) を入力すると、プロンプトに対して指定すべき情報についての説明が表示されます。

注意

lprsetup スクリプトで表示されるシンボルの中には、オペレーティング・システムではサポートされていないものもあります。サポートされているシンボルについての詳細は、printcap(4) を参照してください。

次の例では、lprsetup コマンドを使用して、ローカル・システムで使用する LN03R プリンタを設定する方法を示します。なお、例を簡略化するために一部の表を省いてあります。

```
# /usr/sbin/lprsetup
Tru64 UNIX Printer Setup Program
Command < add modify delete exit view quit help >: add
Adding printer entry, type '?' for help.
Enter printer name to add [lp1] : 

Printer Types:

1. Compaq Advanced Server ClientPS
2. Compaq Advanced Server ClientText
3. Compaq LN16
4. Compaq LN32
5. Digital Colormate PS
6. Digital DEClaser 1100
7. Digital DEClaser 1150
8. Digital DEClaser 2100
9. Digital DEClaser 2150
10. Digital DEClaser 2200
11. Digital DEClaser 2250
12. Digital DEClaser 2300
13. Digital DEClaser 2400
14. Digital DEClaser 3200
15. Digital DEClaser 3250
16. Digital DEClaser 3500
17. Digital DEClaser 5100
18. Digital LA100
19. Digital LA120
20. Digital LA210
21. Digital LA280
22. Digital LA30N

Press 'ENTER' to continue scrolling, type '(q)uit' to end scrolling: 
23. Digital LA30N A4
(and so on until)
59. Digital LN03
60. Digital LN03R
61. Digital LN03S
62. Digital LN03S-JA
63. Digital LN15
64. Digital LN15 A4
65. Digital LN17
66. Digital LN17 A4
```



```

Press 'ENTER' to continue scrolling, type '(q)uit' to end scrolling: q

Help Types:

?          - General help
printer?   - Specific printer type information

Enter index number, help type, '(q)uit', or 'ENTER' [Generic Unknown type]: 60

You chose printer type 'Digital LN03R'.
Is that correct? [y]:y

Enter printer synonym: marks
Enter printer synonym: Enter

Set device pathname 'lp' [] ? /dev/tty01

Do you want to capture print job accounting data ([y]|n)? y

Set accounting file 'af' [/usr/adm/lplacct]? Enter

Set spooler directory 'sd' [/usr/spool/lpd1] ? Enter

Set printer error log file 'lf' [/usr/adm/lplerr] ? Enter

Set printer connection type 'ct' [dev] ? Enter

Set printer baud rate 'br' [9600] ? Enter

```

各プロンプトに対する応答が終了すると、lprsetup は /etc/printcap に指定されているシンボルのいずれかの値を変更するか、または新たにシンボルを指定するかを尋ねるプロンプトを表示します。たとえば、特定のページ長あるいはページ幅を設定することができます。情報の変更あるいは追加を行いたい場合は、そのシンボル名を入力します。詳細については、printcap(4) を参照してください。

```

Enter the name of the printcap symbol you wish to modify. Other
valid entries are:
    'q'      to quit (no more changes)
    'p'      to print the symbols you have specified so far
    'l'      to list all the possible symbols and defaults
The names of the printcap symbols are:

af br cf ct df dn du fc ff fo fs gf if lf lo lp
mc mj mx nc nf of on pl pp ps pw px py rf rm rp
rs rw sb sc sd sf sh st tf tr vf xc xf xn xs ya
yd yj yp ys yt

Enter symbol name: q

Printer #1
-----
Symbol type value
-----
af STR /usr/adm/lplacct
br INT 9600
ct STR dev
fc INT 0177777

```

```

fs    INT    03
if    STR    /usr/sbin/lp03rof
lf    STR    /usr/spool/lp1err
lp    STR    /dev/tty01
mx    INT    0
of    STR    /usr/sbin/lp03rof
pl    INT    66
pw    INT    80
rw    BOOL   on
sd    STR    /usr/spool/lpd1
xc    INT    0177777
xf    STR    /usr/sbin/xf
xs    INT    044000

```

Are these the final values for printer 0 ? [y] **y**

次に `lprsetup` スクリプトは、`/etc/printcap` ファイルにコメントを追加するかどうかを質問します。コメントを追加しない場合は、`n` を入力してください。コメントを追加する場合は、`y` を入力してください。# プロンプトに対してコメントを入力します。# プロンプトで Return キーまたは Enter キーを押すと終了します。入力したコメントは、`/etc/printcap` ファイルの `printcap` エントリの直前に挿入されます。

```

Adding comments to printcap file for new printer, type '?' for help.
Do you want to add comments to the printcap file [n] ? : y
Enter comments below - Press ENTER on empty line to exit
# Use this printer for drafts only
# Enter

```

```

Set up activity is complete for this printer.
Verify that the printer works properly by using
the lpr(1) command to send files to the printer.

```

Command < add modify delete exit view quit help >: **exit**

詳細は、`lprsetup(8)` を参照してください。

8.3.6 Advanced Printing Software のためのプリント・シンボル

Advanced Printing Software を設定する場合は、`Generic_Remote_LPD` プリンタ・タイプを選択し、次のプリント・シンボルを設定してください。

```

rm    @dpa を指定して、このプリンタに送られたジョブを Advanced
      Printing Software 着信ゲートウェイに送ることを示します。着
      信ゲートウェイは、そのジョブを Advanced Printing Software
      スプーラに送ります。

```

```

rp    Advanced Printing Software ロジカル・プリンタの名前を指定
      します。

```

詳細は、*Advanced Printing* の『システム管理/操作ガイド』を参照してください。

8.4 プリント・システムの日常保守

この章の最初の部分で、プリンタをシステムへ新規に設定する方法について説明しました。この節では、設定済みのプリント・システムの日常の管理タスクについて説明します。これらのタスクは、`printconfig` および X11 準拠 (CDE) またはコマンド行ツールを使用して行うことができます。以降の項で説明する処理は、次のとおりです。

- 新しいプリンタの追加 (8.4.1 項)
- 既存プリンタの属性の変更 (8.4.2 項)
- システムからのプリンタの削除 (8.4.3 項)
- CDE ツールまたは `lpc` コマンドによるプリンタの制御 (8.4.4 項)
- プリンタ課金機能の設定 (8.4.5 項)

`/etc/printcap` ファイルから手作業でプリンタを削除した場合は、スプール・ディレクトリ/ファイル、課金ディレクトリ/ファイル、およびエラー・ディレクトリ/ファイルも手作業で削除する必要があります。

8.4.1 プリンタの追加

一度プリンタを設定しておけば、その他のローカル、リモート、およびネットワーク・プリンタをいつでも追加することができます。8.3 節で説明しているように、各プリンタについて情報を集めてください。

プリンタの追加は `printconfig` を使用して行うこともできます。手作業でプリンタの追加を行う場合は、次の手順に従います。

1. プリンタ・スプール・ディレクトリがなければ作成します (8.5.2.6.2 項を参照)。
2. `/etc/printcap` ファイルを編集して、`/usr/sbin/lprsetup` 内のファイルの構成データを使って追加するプリンタについての記述を追加します (8.5.1 項を参照)。
3. 課金ファイルとログ・ファイルを作成し、プリンタの課金機能を有効にします (8.4.5 項を参照)。このファイルの保護と所有権を適切に設定します。

/etc/inittab ファイルの設定で、プリンタが接続されているシリアル・ラインに対して getty プロセスが起動されないようにしてください。
printconfig を使用した場合は、自動的に設定されます。

8.4.2 プリンタ構成の変更

プリンタの構成を変更するには、printconfig を実行して、構成済みプリンタを選択します。次に [変更] を選択して「設定」ウィンドウを開きます。

スプール・ディレクトリ、課金ファイル、またはエラー・ログ・ファイルの名前を変更した場合、printconfig は、元の情報を削除する前に、変更内容が正しいかどうかを確認するよう要求してきます。

プリンタの構成を手作業で変更するには、/etc/printcap ファイルを編集し、そのプリンタ・エントリを修正します。/etc/printcap ファイルのシンボルについては、8.5.1 項 および printcap(4) を参照してください。

8.4.3 プリンタの削除

プリンタを削除するには、printconfig ユーティリティを実行し、削除したいプリンタを選択してから [設定解除] を選択します。エラー・ログ・ファイルおよび課金ファイルを削除するかどうか聞いてきます。課金ファイルは複数のプリンタで共用することができます。このような共用ファイルの場合は削除しないでください。

/etc/printcap ファイル・エントリの最初の行にそのプリンタに対するコメントを記述している場合、コマンド行ユーティリティ lprsetup は、プリンタを削除してもそのコメントを削除しません。コメントは、/etc/printcap ファイルを編集して手作業で削除してください。

手作業でプリンタを削除するには、/etc/printcap ファイルを編集し、そのプリンタに関連するエントリを削除します。不要になった課金ファイル、ログ・ファイル、およびスプール・ディレクトリも手作業で削除する必要があります。

lpc および CDE のプリント管理ツールを使用して、プリンタとキューに一時的にアクセスし制御することもできます。8.4.4 項 を参照してください。

8.4.4 ローカル・プリント・ジョブおよびキューの制御

プリント・ジョブのフローおよびローカル・プリント・キューの内容の管理には `lpc` コマンド行ユーティリティを使用します。

ローカル・ユーザ環境が CDE になっている場合は、CDE の「アプリケーション・マネージャ - デスクトップアプリケーション」フォルダにある「印刷マネージャ」を使ってプリント・ジョブを管理することもできます。グラフィカル・ユーザ・インタフェースの使用法についてはオンライン・ヘルプ、『*Common Desktop Environment: ユーザーズ・ガイド*』および『*Common Desktop Environment: 上級ユーザ及びシステム管理者ガイド*』を参照してください。

`lpc` コマンドを使用して次の処理を行うことができます。

- プリンタおよびスプール・キューを使用可能あるいは使用不可能に設定する。
- キューにいれられたジョブの順番を変更する。
- プリンタ、キュー、およびデーモンの状態を表示する。

たとえば、`disable` コマンドなど、いくつかの `lpc` コマンドには、スーパーユーザ特権が必要です。

注意

`lpc` コマンドで管理できるのは、ローカルなプリンタ・キューのみです。リモート・プリンタには、ローカル・キューとリモート・キューがあります。`lpc` コマンドが管理できるのは、ローカル・キューのみです。

`lpc` コマンドでは、いくつかのコマンド引数が指定できます。`lpc` コマンドは、対話形式でも使用できます。引数なしで `lpc` コマンドを入力すると、`lpc>` プロンプトが表示され、引数が入力できる状態になります。

`lpc` コマンドの構文は次のとおりです。

```
/usr/sbin/lpc [ argument] [all | printer...]
```

コマンド引数の中には、`all` を指定してすべてのプリンタを示したり、1 つまたは複数の `printer` 変数を指定して特定のプリンタを示したりすることがあるものがあります。

表 8-3 に `lpc` コマンドのコマンド引数を示します。

表 8-3: `lpc` コマンドの引数

| <code>lpc</code> の引数 | 説明 |
|------------------------------|---|
| <code>help [argument]</code> | 指定された <code>lpc</code> コマンドの引数の説明を 1 行で表示する。 <code>argument</code> 変数を指定しない場合、引数のリストが表示される。 |
| <code>? [argument]</code> | <code>help</code> 引数と同じ。 |
| <code>abort</code> | 実行中の <code>lpd</code> デーモンを終了させ、その後、プリントを使用不能にする。これによって <code>lpr</code> コマンドまたは <code>lp</code> コマンドで新しい <code>lpd</code> デーモンが起動されないようにする。 |
| <code>check</code> | 指定された各プリンタに関して、構成エラーがないか、プリンタ記述ファイルや、プリント環境の他の構成要素を調べる。 |
| <code>clean</code> | 一時ファイル、データ・ファイル、および制御ファイル (たとえば、完全なプリンタ・ジョブの形になっていないファイルなど) を、指定されたプリント・スプール・ディレクトリから削除する。 |
| <code>disable</code> | 指定されたプリント・スプール・キューを不能にする。これにより、 <code>lpr</code> コマンドまたは <code>lp</code> コマンドがキューに新しいジョブをいれないようにする。 |
| <code>down message...</code> | 指定されたプリント・キューを不能にし、プリントを使用不能にし、プリンタ状態ファイルに指定されたメッセージを記入する。残りの引数は <code>echo</code> と同様に扱われるため、メッセージを引用符で囲む必要はない。 <code>down</code> 引数を使用して、プリンタを停止させ、ユーザに知らせることができる。プリンタが停止した場合、 <code>lpq</code> コマンドは、プリンタが停止したことを示す。 |
| <code>enable</code> | 指定されたプリンタのスプール機能を可能にする。これによって、 <code>lpr</code> コマンドまたは <code>lp</code> コマンドが、プリント・ジョブをスプール・キューにいれることが可能になる。 |
| <code>exit</code> | <code>lpc</code> を終了する。 |
| <code>quit</code> | <code>lpc</code> を終了する。 |

表 8-3: lpc コマンドの引数 (続き)

| lpc の引数 | 説明 |
|------------------|---|
| restart | 指定されたプリンタに対する新しい lpd デーモンの再起動を試みる。何らかの異常状態によって、デーモンが中断し、ジョブがキューに残った場合、この引数が役に立つ。このような場合、lpq コマンドはデーモンが存在しないことを示す。デーモンがハングした場合、まずプロセスを強制終了させ、その後 restart 引数を使用してデーモンを再起動させる。 |
| start | 指定されたプリンタでのプリントを可能にし、スプール・デーモンを起動させる。 |
| status [printer] | 指定されたプリンタのデーモンおよびキューの状態を表示する。status 引数は、キューが使用可能か、プリントが可能か、キューのエントリ数およびプリンタの lpd デーモンの状態を表す。プリンタ名を省略した場合、すべてのプリンタ・デーモンとキューについての情報が表示される。 |
| stop | 現在のジョブの終了後、スプール・デーモンを停止させ、プリントを使用不能にする。 |
| topq printer | プリント・ジョブを、指定された順番でキューに入れる。プリント・ジョブは、request_ID 変数または username 変数を指定することによっても、指定することができる。 |
| up | すべてのプリントを可能にし、新しいプリンタ・デーモンを起動させる。down 引数を取り消す。 |

次の例では、tester というプリンタに対して lpd デーモンが有効で、キューにエントリが 1 つ存在することを示しています。

```
# /usr/sbin/lpc
lpc> status tester
tester:
    printer is on device '/dev/tty02' speed 9600
    queuing is enabled
    printing is enabled
    1 entry in spool area
lpc>
```

詳細については、lpc(8) を参照してください。

8.4.5 プリンタ課金機能の設定

プリンタ課金機能を使用すると、印刷サービスに対する支払をユーザに請求したりプリンタの使用量を調べることができます。

注意

課金情報は、フォーマット化されていないテキスト・ファイルです。PCL や PostScript のようなフォーマット化されたファイルではありません。

プリンタ課金機能には、2 つのタイプがあります。

| | |
|--------------|---|
| プリンタ・ユーザ課金機能 | プリンタ・ユーザ課金機能は、プリント要求をしたシステム名およびユーザ名に従って、プリンタの使用量に関する情報を提供します。 |
| プリンタ・サマリ課金機能 | プリンタ・サマリ課金機能は、プリンタが消費した媒体の量 (出力ページ数、ロール紙またはフィルムのフィート長) についての情報を提供します。プリンタ・サマリ課金情報を生成するには、 <code>-s</code> オプションを指定して <code>pac</code> コマンドを実行します。 |

プリンタ課金ファイルは省略時のディレクトリ `/var/adm` に作成されます。`lprsetup` を使ってプリンタを追加すると、`lprsetup` は指定した課金ファイルを作成します。省略時の課金ファイルは、`/usr/adm/lpd/lpacct` です。

`printconfig` ユーティリティでは、[詳細] オプションの中で省略時の課金ファイルを選択します。課金が不要の場合は、プリンタの構成時または[変更] オプションを使って後日いつでもこれらのエントリを削除することができます。プリンタを手動で追加する場合は、ユーザが課金ファイルを作成する必要があります。

注意

ユーザ `adm` は、ディレクトリ `/var/adm/lpd` を所有し、`adm` グループに属します。また、ユーザ `adm` は、プリンタ課金ファイルも所有します。このファイルは、保護モードが 644 で、グループ `system` に属します。

`/etc/printcap` のプリンタ・エントリにあるパラメータ `af` は、課金ファイルの名前を指定しています。課金処理は、各プリンタに各ユーザがプリントしたページ数を記録するためにこのファイルを使用します。課金ファイル名

は、システムにあるプリンタの間で一意です。プリンタ課金ファイル内の情報を表示するには、`pac` ユーティリティを使用します。ユーザ `daemon` は、課金ファイルを所有し、グループ `daemon` に属します。`printconfig` ユーティリティを使用してプリンタ課金ファイルを指定すれば、正しいファイル所有権が自動的に設定されます。`af` パラメータは、リモート・プリンタのエントリには適用されません。

課金機能はプリント・フィルタと呼ばれるプログラムを通して実行されます。`printconfig` ユーティリティは、省略時のプリント・フィルタを提示してくれます。課金機能に必要なプリント・フィルタのシンボルは、`if` です。例を以下に示します。

```
if=/usr/lbin/ln03rof
```

システムでプリンタごとに別々の課金ファイルを使用する場合、そのファイル名は一意でなければなりません。1つの課金ファイルを共用できるプリンタの数には制限がありません。しかし、リモート・プリンタに対しては、課金ファイルを指定することができません。プリント・デーモンは、課金ファイルを所有します。課金ファイルを指定すると、必要に応じて中間ディレクトリが自動的に作成されます。

8.5 参照情報

この節では、プリンタの構成に必要な情報について説明します。また、ライン・プリンタ・デーモン (`lpd`) およびプリント・システムのファイルの情報について説明します。8.3.2 項に示すように `printconfig` を使用した場合、システム・ファイルが自動的に作成されます。

これらのファイルは手作業で作成/変更することもできます。ただし、手作業で作成した場合、変更内容を有効にするために `/etc/printcap` ファイルも手作業で変更する必要があります。

8.5.1 `/etc/printcap` ファイル

`lpd` デーモンは、`/etc/printcap` データベース・ファイルを使用して要求をプリントします。`/etc/printcap` ファイルの各エントリはプリンタについて記述しています。プリンタ特性は、シンボルと呼ばれる2文字の英字からなる省略形を使って指定します。プリント・シンボルについては、この項と `printcap(4)` に説明があります。`/etc/printcap` ファイルの変更は `lprsetup` ユーティリティで行います。

省略時のプリンタ名は `lp` で、任意のタイプ (ローカルやリモート) のプリンタの別名として使用されます。 `lp0`, `lp1` などの名前は、`printconfig` によって割り当てられる省略時の名前です。これらを採用することも、無視することも可能です。

TCP/IP 接続プリンタ、LPD リモート・プリンタ、シリアル・プリンタ、およびパラレル・プリンタに対する `/etc/printcap` エントリを、以下に示します。

TCP/IP 接続プリンタの例

```
lp0|myprinter|hp8000:\
:ct=tcp:\
:if=/usr/lbin/ppdof +OPageSize=Letter +Chplj8000.rpd:\
:lf=/usr/adm/lp0err:\
:lp=@myprinter.com/9100:\
:mx#0:\
:of=/usr/lbin/ppdof +OPageSize=Letter +Chplj8000.rpd:\
:pl#66:\
:pw#0:\
:rw:\
:sd=/usr/spool/lpd0:\
:xf=/usr/lbin/xf:
```

LPD リモート・プリンタの例

```
lp0|remote:\
:lf=/usr/adm/lp0err:\
:lp=:\
:rm=system:\
:rp=queue:\
:sd=/usr/spool/lpd0:
```

注意

この構成では、大部分の `printcap` オプションが無効になっています。

シリアル・ポートの例

```
lp0|myprinter|la75:\
:af=/usr/adm/lp0acct:\
:br#9600:\
:ct=dev:\
:fc#0177777:\
:fs#03:\
```

```

:if=/usr/lbin/la75of:\
:lf=/usr/adm/lp0err:\
:lp=/dev/tty00:\
:mx#0:\
:of=/usr/lbin/la75of:\
:pl#66:\
:pw#80:\
:rw:\
:sd=/usr/spool/lpd0:\
:xc#0177777:\
:xf=/usr/lbin/xf:\
:xs#044000:

```

パラレル・ポートの例

```

lp0|myprinter|la75:\
:af=/usr/adm/lp0acct:\
:ct=dev:\
:if=/usr/lbin/la75of:\
:lf=/usr/adm/lp0err:\
:lp=/dev/lp0:\
:mx#0:\
:of=/usr/lbin/la75of:\
:pl#66:\
:pw#80:\
:sd=/usr/spool/lpd0:\
:sh:\
:xf=/usr/lbin/xf:

```

/etc/printcap に記述されている，ローカル・プリンタおよびリモート・プリンタのエントリ例を次に示します。注釈には，そのシンボル・エントリの指定方法を説明してあります。

```

#
#
lp|lp0|0|dotmatrix|mary:\
:af=/usr/adm/printer/lp.acct:\
:br#9600:\
:ct=dev:\
:fc#0177777:\
:fs#023:\
:if=/usr/lbin/la75of:\
:lf=/usr/adm/lperr:\
:lp=/dev/tty01:\
:mx#0:\
:of=/usr/lbin/la75of:\
:pl#66:\
:pw#80:\

```

1

```

:sh:\ [2]
:sd=/usr/spool/lpd:\
:xc#0177777:\ [3]
:xf=/usr/lbin/xf:\
:xs#044000:\
#
sqirrl|3r3|ln03r3|postscript3|In office 2T20:\
:lp=:rm=uptown:rp=lp:sd=/var/spool/printer/ln03r3:mx#0:\ [4]
#

```

- [1] 英字列付きでシンボルを指定します。
- [2] 論理式を表すシンボルを指定します。
- [3] 数値付きでシンボルを指定します。
- [4] リモート・プリンタのエントリを指定します。クライアントとしてリモート・プリンタを使用するためには、シンボル `lp`、`rm`、`rp`、および `sd` を指定する必要があります。

各プリンタ・エントリの最初の行には、一次プリンタ参照名およびプリンタ名の同義語を指定するフィールドがあります。この最初の行およびこれらのフィールドは、ローカルおよびリモートいずれのプリンタにも必要です。

プリンタ参照名とは、後でこのプリンタへプリントするために指定する名前のことです。1 台のプリンタに対して、いくらでも別名を指定することができ、最初の行で各フィールドを | (縦線) を使って区切ります。また、最初の行の終わりには、: (コロン) が必要です。

注意

/etc/printcap ファイルのローカル・プリンタ・エントリには省略時のプリンタ参照名 `lp0` を割り当てて、プリント・コマンドでプリンタ参照名を指定しなかった場合にもプリント・ジョブにプリンタが指定されるようにしなければなりません。

各プリンタ・エントリの残りの行には、プリンタの構成を定義する記述シンボルと値を入力します。シンボルは、2 文字のニーモニックで、= (等号) と英字列、または # (番号記号) と数値を使って指定できます。シンボル名の中には論理値を持つものもあり、これらにはパラメータを付けません。各シンボルは 1 行に指定することも、各行に指定することもできますが、: (コロン) で区切らなければなりません。

/etc/printcap ファイルを読みやすくするため、行の先頭に：(コロン)を、行の終わりには \ (バックスラッシュ) を付けてシンボルを区切ることができます。

printcap(4) に、printcap シンボル名、シンボル名がとる値の型、省略時の値、およびシンボルの説明が示されています。

8.5.2 /etc/printcap のデータ

/etc/printcap ファイルのプリンタ・エントリで通常必要となるデータを以降に示します。

8.5.2.1 プリンタ名

プリンタ名は、lpr コマンドでプリンタを指定するために使用する名前です。例を次に示します。

```
# lpr -Pprintername
```

lprsetup ユーティリティは、0 ~ 99 の内部ナンバリング・スキームを使用します。使用可能な次の番号が、省略時の名前になります。Enter キーを押すことによって、省略時の名前を選択できます。他の英数字名を入力することもできます。lprsetup ユーティリティは少なくとも 2 つのプリンタ別名を必ず割り当てます。省略時の番号 *N* は、1 つの別名です。文字列 lp プラス省略時番号 (lp*N*) は、システムが割り当てるもう 1 つの別名です。省略時番号が 1 の場合、割り当てられる 2 つの名前は、1 および lp1 です。したがって、ジョブは、次のいずれかのコマンドを使ってこのプリンタのキューにいれることができます。

```
# lpr -P1
# lpr -Plp1
```

独自の別名を割り当て、それを使ってジョブをプリンタに送ることもできます。

最初に lprsetup が設定するプリンタの名前は 0 と lp0 になります。名前 lp は、省略時のプリンタとして予約されています (つまり、lpr コマンドで何も指定しなかったとき、使用されます)。

1 つのプリンタしかない場合、または多くのプリンタ名の最初のプリンタを入力する場合、最初の名前はプリンタ番号 0 になります。追加名 lp があれば、これが省略時のプリンタとして、認識されます。

ネットワーク上のリモート・プリンタの名前は、ネットワーク管理者に問い合わせてください。

8.5.2.2 プリンタ・タイプ

プリンタ・タイプは、LN03 レーザ・プリンタなど、そのプリンタの製品名に対応しています。lprsetup ユーティリティを使用すると、プリンタがタイプ別にリストされます。サポートされているタイプだけがリストされます。これらのプリンタには、省略時の値があり、設定ユーティリティに組み込まれています。

サポートされているプリンタ・タイプは /usr/lbin/lprsetup ディレクトリ内のファイルで定義されています。汎用プリンタ・タイプや HP の Advanced Server で使用されているプリンタ・タイプの情報は、lprsetup.dat(4) を参照してください。

リストにないプリンタでも、Generic_Unknown プリンタ・タイプを選択し、サポートされているプリンタと同様の値を使用してプロンプトに回答することで設定することができます。

プリンタ・タイプを指定する場合は、完全なコマンド名とプリンタ名を使用する必要があります。省略時のプリンタ・タイプは、Generic_Unknown です。

他社のプリンタをインストールする場合は、そのプリンタに付属のドキュメントを参照してください。

8.5.2.3 プリンタの別名

プリンタの別名は、そのプリンタの代替名です。いくつか例をあげると、draft, letter, および LA-75 Companion Printer などがそうです。プリンタの別名は、必要なだけ入力できますが、別名を指定する行は 80 字未満でなければなりません。名前を多く含む別名を入力する場合は、複数行になることがあります。その場合、空白行またはホワイト・スペースだけを含む行を入力すれば、その別名入力終了します。

コマンド行モードでは、別名を入力すると、再度プロンプトが表示されます。それ以上、別名を入力しない場合は、Enter キーを押して先に進みます。

各別名 (プリンタ番号も含む) は、プリントシステムに対してプリンタを指定するために使用します。たとえば、あるプリンタの別名として draft を

指定している場合、次のコマンドを入力することにより、そのプリンタにファイルを出力できます。

```
$ lpr -Pdraft files
```

特別なプリンタ別名 lp は、省略時のプリンタを指しています。

8.5.2.4 デバイス特殊ファイル

デバイス特殊ファイルは、プリンタが接続されているコンピュータのポートにアクセスするために使用します。デバイス特殊ファイルが使用されるのは、プリンタが直接ローカルのシリアル・ポートまたはパラレル・ポートに接続されている場合です。この場合は、`/etc/printcap` ファイルで lp シンボルを使用して、プリンタ・デバイス論理名がプリンタのデバイス特殊ファイル名と同等であると定義する必要があります。たとえば、次のように定義します。

```
lp=/dev/lp0
```

インストール・プロシージャによって、コンピュータに接続されているハードウェアのデバイス特殊ファイルがいくつか作成されます。通常、パラレル・プリンタ用のデバイス特殊ファイルには `/dev/lpn` (たとえば `lp0` , `lp1` , `lp2`) という名前が付けられます。シリアル・ライン・プリンタ用のデバイス特殊ファイルには `/dev/ttynn` (たとえば `tty00` , `tty01` , `tty02`) という名前が付けられます。 `n` および `nn` 変数はプリンタの番号を指定します。ほとんどのシステムでは、デバイス名が省略時の物理ポート (コネクタ) に対応します。

表 8-4 に、デバイス名と省略時の物理ポートの対応を示します。

表 8-4: 通信ポートとプリンタのデバイス特殊ファイル

| デバイス特殊ファイル | 通信タイプ | コネクタ・ラベル |
|-------------------------|-------|-------------|
| <code>/dev/lp0</code> | パラレル | プリンタまたは lp |
| <code>/dev/tty00</code> | シリアル | COMM1 または 1 |
| <code>/dev/tty01</code> | シリアル | COMM2 または 2 |

システムに 9-ピンのシリアル・コネクタが 1 個しかない場合は、そのコネクタにラベル表示がないこともあります。ラベルの代わりにグラフィカル・アイコンを使用しているシステムもあります。詳細については、システムのオーナーズ・マニュアルを参照してください。

注意

ログイン用にポートが使用されている場合、`lprsetup` スクリプトは `getty` プロセスで確立された端末回線の設定を解除します。

TCP/IP プリンタの場合は、`lp` シンボルに対して、次に示すとおり、アットマーク文字 (@) の後に `printer hostname` と、ポート番号またはサービス名のいずれかを続けたものを割り当てます。

```
lp=@printer hostname/servicename  
lp=@printer hostname/portnumber
```

`printer hostname` はネットワーク名またはネットワーク・アドレスで、プリンタのネットワーク・インタフェースの TCP/IP アドレスを表わします。`portnumber` は、整数の TCP ポート番号で、プリンタが raw ソケット・プリントに使用するポートを表わします。`servicename` は `/etc/services` ファイルで定義された名前です。サービス定義には、サービス名、プロトコル「tcp」、およびプリンタが raw ソケット・プリントに使用するポートの番号が含まれます。例として、`/etc/services` ファイルのエントリを示します。

```
print_port_9100      9100/tcp      # printer port 9100
```

また、`/etc/printcap` ファイルのエントリを以下に示します。

```
lp1|1|myprinter:\  
:lp=@printer123.sf.ourcomp.com/print_port_9100:\  
:lf=/usr/adm/lplerr:\  
:if=/usr/lbin/ppdof +OPageSize=Letter +Chplj9000.rpd:\  
:mx#0:\  
:of=/usr/lbin/ppdof +OPageSize=Letter +Chplj9000.rpd:\  
:pl#66:\  
:pw#0:\  
:rw:\  
:sd=/usr/spool/lpd1:\  
:xf=/usr/lbin/xf:
```

8.5.2.5 接続タイプ

`ct` パラメータは、そのプリンタの接続タイプを指定します。プリンタは、ポートあるいはターミナル・ラインからコンピュータへ直接接続できます。LAT (ローカル・エリア・トランスポート) ターミナル・サーバあるいはリモート・ホストに接続されているネットワーク・プリンタへアクセス

スすることができます。 `lprsetup` を使用している場合に選択する接続タイプは次のとおりです。

`dev` ローカル・デバイス

`lat` LAT デバイス

`tcp` TCP/IP デバイス

8.5.2.6 スプール・ディレクトリ

`/etc/printcap` では、`sd` パラメータは、ファイルを出力する前にキューイングしておくスプール・ディレクトリを指定します。各スプール・ディレクトリは、一意です。すべての `printcap` ファイル・エントリには、ローカルおよびリモートの両方で、スプール・ディレクトリを指定しなければなりません。`printconfig` によりスプール・ディレクトリが作成される際、必要に応じて中間ディレクトリが作成されます。

各プリンタには、`/usr/spool` ディレクトリの下に各プリンタ固有のスプール・ディレクトリが必要です。スプール・ディレクトリはプリンタのスプール・キューとして機能します。このディレクトリには、そのプリンタへ出力するためにキューにいれられたファイルが置かれます。プリンタ用のスプール・ディレクトリは、プリンタの参照名と同じ名前で、プリンタが接続されているシステムに存在しなければなりません。プリンタの参照名は、出力先のプリンタを指定する際に使う名前です。

`lprsetup` を使用した場合、ユーティリティは省略時の値 `/usr/spool/lpdn` を使用します。変数 `n` は、プリンタの番号を指定します。たとえば、2 回目のライン・プリンタのスプール・ディレクトリの省略時の名前は、`/usr/spool/lpd2` になります。

そのプリンタが他のシステムまたは他のネットワークに接続されていても、`/etc/printcap` ファイル中の各プリンタ・エントリには、スプール・ディレクトリを指定します。スプール・ディレクトリは、`sd` シンボルを使って指定します。たとえば、次のように指定します。

```
sd=/usr/spool/local_printer1
sd=/usr/spool/network_printer1
sd=/usr/spool/remote_printer1
```

注意

`/usr/spool/lpd` または `/var/spool/lpd` には、サブディレクトリを置かないでください。これは親 `lpd` プロセスで使用される CDSL であり、クラスタ内の各ノードに固有のものなので、名前を変えたり、削除したりしないでください。詳細は、第 6 章の CDSL についての項を参照してください。

8.5.2.6.1 スプール・ディレクトリのファイル

プリントのためにファイルがキューにいれられると、スプール・ディレクトリには、`lpd` デーモンによって `status` ファイルと `lock` ファイルが作成されます。`/var/spool/lpd/lpd.lock` ファイルには、親 `lpd` プロセスのプロセス ID が格納されています。この親プロセスは、ローカル `/dev/printer` ソケットおよびネットワーク・ソケット 515 上でプリント・ジョブ要求を待ちます。実際にジョブをプリントするプロセスは、親プロセスから生成される子デーモンです。ジョブをプリントするプロセスの ID は、スプール・ディレクトリの `lock` ファイル (たとえば `/usr/spool/local_printer1/lock`) 内に格納されます。

`lock` ファイルは、1 つのキュー内のジョブを処理する子プロセス (`lpd` デーモン) の生成を規制し、一時点に 1 つのキュー・デーモンだけが実行されるようにします。`lock` ファイルには、現在実行しているデーモンのプロセス ID 番号と、現在処理しているジョブの制御ファイル名が格納されます。`status` ファイルには、現在のプリンタ状態を示す行が格納されます。ユーザがプリンタ状態を照会すると、この行が表示されます。照会したプリンタの状態が使用不能の場合には、標準出力に `no entries` という状態メッセージが出力されます。スプール・ディレクトリには、この他に、次に示す 2 つのファイルが一時的に作成されます。

| | |
|-------------------------|--|
| <code>.no_daemon</code> | キューにエントリがあってもデーモンが実行されておらず、しかも過去 10 秒間にスプール・ディレクトリ内のファイルに削除または変更がなかった場合に、作成されます。 |
|-------------------------|--|

| | |
|-------------------|---|
| <code>.seq</code> | <code>/usr/bin/lpr</code> コマンドにより作成されるファイルで、0 から始まるジョブ番号を生成します。 |
|-------------------|---|

lpd デーモンがプリント要求により起動すると、プリンタのスパール・ディレクトリ内で lock ファイルを探します。lock ファイルが見つからない場合には、lpd デーモンは lock ファイルを作成して、そのファイルの連続した 2 行に、ID 番号と制御ファイル名を書き込みます。次に、lpd デーモンはプリンタのスパール・ディレクトリで制御ファイルを探します。制御ファイルは名前が cf で始まります。制御ファイルは、ユーザがサブミットしたプリント・ファイル名を指定し、これらのファイル用のプリント命令が入っています。制御ファイルの各行は、キー・キャラクタで開始します。このキャラクタは、その行の残りについての処理内容を示すものです。キー・キャラクタの詳細については lpd(8) を参照してください。

注意

ジョブ制御ファイル (cf で始まる) は最初、tf で始まる一時ファイルとして作成されます。ジョブ生成プロセスが異常終了すると、一時制御ファイルが残ります。この一時制御ファイルは削除してください。.seq ファイル内のジョブ番号が、一時制御ファイル名で使用されている値と等しくなると、ジョブ生成に支障をきたすからです。

通常 df で始まるデータ・ファイルも、スパール・ディレクトリにあります。データ・ファイルには、プリントされるジョブが入っています。データ・ファイルはプリント・フィルタで変更される場合と変更されない場合があります。プリント・フィルタは、/etc/printcap 内の if、または of (if が無い場合) によって指定されます。

ファイルがプリントされると、lpd デーモンは、プリンタのスパール・キューから制御ファイルとデータ・ファイルを削除します。そして、次に状態ファイルを更新し、スパール・キュー内の次のファイルをプリントする準備をします。

たとえば、milhaus というプリンタにジョブをサブミットした場合に、次のコマンドを実行すると、スパール・ディレクトリに格納されているファイルが表示されます。

```
# ls -l /usr/spool/local_printer1
-rw-rw---- 1 root 75 Jan 17 09:57 cfA0220mothra
-rw-rw---- 1 root 96 Jan 17 10:03 cfA143harald
-rw-rw---- 1 root 199719 Jan 17 09:57 dfA0220mothra
-rw-rw---- 1 root 9489 Jan 17 10:03 dfA143harald
```

```
-rw-r--r-- 1 root 20 Jan 17 10:06 lock
-rw-rw-rw- 1 daemon 113 Jan 17 10:00 status
```

8.5.2.6.2 スプール・ディレクトリの作成

手動でプリンタを追加したい場合は `mkdir` コマンドで各プリンタに対するスプール・ディレクトリを作成してください。スプール・ディレクトリの許可モードには 775 を、ディレクトリのグループと所有権モードには、`daemon` という名前を設定します。たとえば、次のように設定します。

```
# cd /var/spool
# mkdir ./printers
# mkdir ./printers/lp1
# cd printers
# chmod 775 lp1
# chgrp daemon lp1
# chown daemon lp1
# ls -l lp1
drwxr-xr-x 2 daemon daemon 24 Jan 12 1994 lp1
```

8.5.2.7 ボー・レート

ボー・レートは、直接接続のシリアル回線プリンタだけに適用されます。

ボー・レートは、データ・ソースとプリンタ間でデータを転送する最高速度です(たとえば 4800 または 9600 など)。ご使用のプリンタの省略時のボー・レートはハードウェア・マニュアルに記載されています。プリンタをインストールする際にボー・レートを再設定した場合は、`/etc/printcap` ファイルに設定するボー・レートも同じ値にする必要があります。

8.5.3 ライン・プリンタ・デーモン

プリンタは、`/usr/sbin` ディレクトリに存在するライン・プリンタ・デーモン `lpd` によって制御されます。プリント処理を行うためには、`lpd` デーモンを実行しておく必要があります。`lpd` デーモンには次に示すような多数の機能があります。

- `lpr` および `lprm` コマンドと一緒に becoming プリンタ・スプーリング処理を行います。スプーリングとは、プリンタがファイルを出力できる状態になるまでの間、ファイルをキューに保持しておくためのメカニズムです。
- `/etc/printcap` ファイルを走査してプリンタ特性を調べます。

- プリント要求に対して特定のプリント・フィルタを適用します。プリント・フィルタは、入力フォーマットをプリンタ固有の出力フォーマットに変換します。
- システムの動作停止時にプリント・ジョブがキューに残っていた場合、システムのリブート後、プリントされていないファイルをすべてプリントします。

`lpr` コマンドでファイルのプリントを要求すると、スプール・ディレクトリにそのファイルがコピーされ、`lpd` デーモンが起動されます。各ファイルは、要求がキューに入った順番で出力されます。ファイルのコピーは、プリンタの準備ができるまでキューに保持されます。`lpd` デーモンは、プリンタにそのジョブを送った後、スプール・キューから該当ファイルを削除します。

システムをインストールしてブートすると、通常、`/sbin/init.d/lpd` ユーティリティによって `lpd` デーモンが起動されます。次のコマンドを使用して `lpd` デーモンを起動したり停止したりできます。

`/sbin/init.d/lpd [-l] start`

`/sbin/init.d/lpd stop`

`-l` オプションを指定すると、`lpd` デーモンは、ネットワークからの有効な要求のログを記録します。このオプションはデバッグの際に便利です。

ライン・プリンタ・デーモンが実行されているかどうかを確認するには、次のように入力します。

```
# ps agx | grep /usr/sbin/lpd | grep -v grep
```

8.5.4 エラー・ロギング

`lpd` デーモンは、`/var/adm/syslog.dated/<date>/lpr.log` (または最新ログ・ファイルにシンボリック・リンクされている `/var/adm/syslog.dated/current/lpr.log`) にエラーのログを記録します。ディレクトリ名 `<date>` は、ログが保存された日時を示す名前が付けられます。ログ・ファイルのエントリ例を次に示します。

```
Apr 15 16:36:28 cymro lpd[1144]: ERROR -- lpr: cannot open printer description file
Apr 15 16:36:28 cymro lpd[1144]: ERROR -- exiting ...
Apr 15 16:36:46 cymro lpd[1130]: ERROR -- lpq: cannot open printer description file
Apr 15 16:36:46 cymro lpd[1130]: ERROR -- exiting ...
#
```

ログ・ファイルにエラーがないかどうかを定期的にチェックし、使用可能なディスク・スペースがなくならないようにログを削除(または保管)します。

cron ユーティリティを使用すれば、定期的なクリーンアップを自動的に行わせることができます。また、必要な優先度を `/etc/syslog.conf` ファイルに指定すれば、`lpr.log` に記録されるメッセージの重要度を制御することもできます。詳細については、`syslogd(8)` を参照してください。

`lf` パラメータは、プリント・フィルタ・エラーのほとんどが報告されるログ・ファイルを指定します。省略時のログ・ファイルは、`/dev/console` です。複数のプリンタがシステムに接続されている場合は、各ログ・ファイルに一意の名前を付与します。`printconfig` を使ってログ・ファイルが作成される場合は、パス名内の中間ディレクトリが必要に応じて作成されます。

`lpd` デーモンは、`lf` パラメータで指定されたエラー・ファイルではなく、エラー・ログ・ファイルにほとんどのプリンタ・エラーのログを取ります。したがって、エラー・ログ・ファイルの指定は任意です。プリンタのインストールに `lprsetup` ユーティリティを使用した場合、省略時のエラー・ログ・ファイルは `/usr/adm/lpd/lperr` になります。エラー・ログ・ファイルを指定しないと、エラーのログは `/dev/console` に取られます。

エラー・ログ・ファイルは、次の例のように `/etc/printcap` ファイルにシンボル `lf` を使って指定します。

```
lf=/var/adm/lpd/lpderr
```

エラー・ログ・ファイルは、通常、ディレクトリ `/var/adm` に置かれます。ローカル・プリンタでは、エラー・ログ・ファイルを共用できますが、`/etc/printcap` ファイルのそれぞれのプリンタ・エントリにそのエラー・ログ・ファイルを指定する必要があります。

8.5.5 プリント・フィルタとフィルタ・ディレクトリ

`lpd` デーモン用のフィルタは、プリントするデータをそのプリンタに適したフォーマットに変換します。多くの場合、フィルタは、各プリンタに対応したものを指定します。たとえば、`LN03R` プリンタでファイルをプリントするには、`ln03roff` フィルタを使用します。

入力フィルタと出力フィルタの両方を指定すると、入力フィルタは、プリンタに送るジョブ・データを処理し、出力フィルタは、`lpd` が生成するバナー・ページのデータを処理します。多数のフィルタが入力または出力フィルタのいずれにも指定できますが、入力フィルタとするか出力フィルタとするかによってそれぞれ異なる動作をします。これらのフィルタは、`/etc/printcap` ファイルで次のように指定します。

```
if=/usr/lbin/ln03rof
of=/usr/lbin/ln03rof
```

入力フィルタは、プリンタ課金の役割も果たします。つまり、テキストのプリント (PostScript のプリントは除く) について、プリンタの使用量を記録します。プリンタ課金の構成方法については、第 10 章を参照してください。

1 つの入力フィルタだけを指定した場合、それがジョブ・データとバナー・ページ・データのフィルタとなり、プリンタ課金の役割も果たします。

1 つの出力フィルタだけを指定した場合、それがジョブ・データとバナー・ページ・データのフィルタとなりますが、プリンタ課金の役割は果たしません。

これらの動作を、次の表にまとめます。

| フィルタ | ジョブ・データ | バナー・ページ・データ | 課金 |
|---------------|---------|-------------|----|
| 入力フィルタと出力フィルタ | | | |
| 入力フィルタ | x | | x |
| 出力フィルタ | | x | |
| 入力フィルタのみ | x | x | x |
| 出力フィルタのみ | x | x | |

詳細は、lpd(8) を参照してください。

汎用プリント・フィルタと、HP の Advanced Server で使用されるプリント・フィルタの詳細は、lprsetup.dat(4) を参照してください。これらのプリント・フィルタは、/usr/lbin/lprsetup ディレクトリにあります。lprsetup.dat(4) のリストにないプリンタ、またはディレクトリで定義されていないプリンタのフィルタ情報は、プリンタのマニュアルを参照するかプリンタの製造元に問い合わせてください。

注意

Tru64 UNIX でサポートされる他社製プリンタのフィルタは、インストレーション・ソフトウェアでインストールされるようになって

います。他社製プリンタのソフトウェアとそれが作成するキューは、プリンタの製造会社以外は変更しないことをお勧めします。

8.5.6 フラグ・ビット

フラグ・ビットは、シリアル・ライン (LAT および RS232) におけるホストからプリンタへのデータ転送に関する特性、また、可能な場合は、プリンタからホストへのデータ転送に関する特性を指定します。プリンタからホストへ渡されるデータには、開始および停止の状態情報があります。この情報は、プリンタの入力バッファが入力を受け入れ可能か、またはバッファに余裕がないかをホストに知らせます。

遅延は、次のグループの文字列を入力バッファへ伝送する速度を遅くするために使用する特定の時間です。遅延により、復帰、改行、タブ、および改ページなどの動作を実行する時間がプリンタ機構に与えられます。

フラグ・ビットは、`fc` シンボルによってクリアされ、`fs` シンボルで設定されます。すべてのプリンタがすべてのフラグ・ビットを使用するわけではありませんが、ビットは必ず設定またはクリアしなければなりません。フラグ・ビットの詳細は、プリンタのマニュアルを参照してください。

フラグ・ビットは、16 ビット・ワードの 8 進数として指定されます。8 進数値は、先頭に 0 (ゼロ) を付けて 8 進法の値であることを示します。すべてのビットをクリアするには、`fc` シンボルに値 `0177777` を指定します。すべてのビットを設定するには、`fs` シンボルに値 `0177777` を指定します。すべてのビットは、`fs` シンボルを呼び出す前に `fc#0177777` を使用してクリアしなければなりません。どのグループのビットの設定またはクリアにも、ビットを組み合わせるフラグ・ビットの数を表す 8 進数の合計値を指定します。

次の例は、フラグ・ビットの指定例です。

```
fc#0177777
fs#0141
```

この例では、`fc#0177777` ですべてのビットがクリアされ、`fs` シンボルには `0141` が設定されて、`OPOST`、`ONLRET`、および `OFILL` フラグ・ビットが設定されます。

それぞれのフラグ・ビット名とその 8 進数値、およびその説明を、表 8-5 にリストします。

表 8-5: フラグ・ビット

| フラグ | 8 進数値 | 説明 |
|--------|---------|---------------------|
| OPOST | 0000001 | 出力処理を可能にする。 |
| ONLCR | 0000002 | NL を CR-NL へマップする。 |
| OLCUC | 0000004 | 小文字を大文字へマップする。 |
| OCRNL | 0000010 | CR を NL へマップする。 |
| ONOCR | 0000020 | カラム 0 では CR を出力しない。 |
| ONLRET | 0000040 | NL が CR 機能を実行する。 |
| OFILL | 0000100 | 遅延のため充填文字を使用する。 |
| OFDEL | 0000200 | 充填文字は DEL。省略時は NUL。 |
| NLDLY | 0001400 | 復帰改行遅延。 |
| NL0 | 0000000 | |
| NL1 | 0000400 | |
| NL2 | 0001000 | |
| NL3 | 0001400 | |
| TABDLY | 0006000 | 水平タブ遅延。 |
| TAB0 | 0000000 | |
| TAB1 | 0002000 | |
| TAB2 | 0004000 | |
| TAB4 | 0006000 | |
| CRDLY | 0030000 | キャリッジ・リターン遅延。 |
| CR0 | 0000000 | |
| CR1 | 0010000 | |
| CR2 | 0020000 | |
| CR3 | 0030000 | |
| FFDLY | 0040000 | 改ページ遅延。 |
| FF0 | 0000000 | |
| FF1 | 0040000 | |
| BSDLY | 0100000 | バックスペース遅延。 |
| BS0 | 0000000 | |

表 8-5: フラグ・ビット (続き)

| フラグ | 8 進数値 | 説明 |
|--------|---------|---------------|
| BS1 | 0100000 | |
| OXTABS | 1000000 | タブをスペースに展開する。 |

詳細は、tty(7) を参照してください。

8.5.7 モード・ビット

モード・ビットは特定の端末の機能に関する詳細を指定し、通常、プリンタ動作には影響しません。モード・ビットは、xc シンボルでクリアされ、xs シンボルで設定されます。プリンタによってはモード・ビットをすべて使用するものもあるため、必ず設定またはクリアしなければなりません。モード・ビットは 16 ビットワード形式の 8 進数で指定されます。xs シンボルを指定する前に、xc#0177777 を指定してすべてのビットをクリアしなければなりません。

詳細は、tty(7) を参照してください。

次の例は、モード・ビットの指定例です。

```
xc#0177777
xs#044000
```

この例では、xc#0177777 ですべてのビットがクリアされ、xs シンボルには 044000 が設定されて、ECHO および ECHOCTL モード・ビットが設定されます。

表 8-6 に、それぞれのモード・ビットの説明をリストします。

表 8-6: モード・ビット

| モード | 8 進数値 | 説明 |
|---------|---------|--------------------------|
| ECHOKE | 0000001 | 行を消去することにより KILL をエコーする。 |
| ECHOE | 0000002 | 視覚的に文字を消去する。 |
| ECHOK | 0000004 | KILL の後で NL をエコーする。 |
| ECHO | 0000010 | エコーを可能にする。 |
| ECHONL | 0000020 | ECHO がオフの場合にも NL をエコーする。 |
| ECHOPRT | 0000040 | \ と / の間に、消去された文字をエコーする。 |

表 8-6: モード・ビット (続き)

| モード | 8 進数値 | 説明 |
|-----------|---------|------------------------------------|
| ECHOCTL | 0000100 | 制御文字を ^ (char) としてエコーする。 |
| ISIG | 0000200 | 特殊文字 INTR, QUIT および SUSP を使用可能にする。 |
| ICANON | 0000400 | 標準入力を可能にする。 |
| ALTWERASE | 0001000 | 代替消去アルゴリズムを使用する。 |
| IEXTEN | 0002000 | FLUSHO および LNEXT を使用可能にする。 |
| XCASE | 0040000 | 標準大文字/小文字表現。 |

8.5.8 リモート・プリンタの特性

リモート・システムからユーザがプリンタにアクセスする場合、ローカル・システムとリモート・システムの両方の `/etc/printcap` ファイルにプリンタ情報が必要です。プリンタが接続されているローカル・システムでは、`/etc/hosts.lpd` または `/etc/hosts.equiv` によりプリンタのセキュリティが制御されます。

オプションで `rs` シンボルを指定できます。このシンボルは、その他のプリンタ・シンボルと同様、論理値として `true (yes)` または `false (no)` のどちらかをとりまします。 `true` を指定した場合、リモート・ユーザはプリンタが接続されているローカル・システムにアカウントが必要になります。 `false` を指定した場合、`/etc/hosts` ファイルにローカル・プリンタがリストされていれば、リモート・ユーザはそのローカル・プリンタにアクセスできます。`/etc/printcap` ファイルの例については、8.5.1 項を参照してください。

リモート・システムでは、`rm`、`rp`、`lp`、および `sd` のシンボルを指定しなければなりません。

`rm` シンボルには、次のようにプリンタが接続されているシステムの名前を指定します。

```
rm=deccom
```

`rp` シンボルには、次のように、リモート・システム上のプリンタ・スプール名を指定します。

```
rp=ln03lab
```

リモート・プリンタには、次のように値なしで `lp` シンボルを指定します。

```
lp=
```

`sd` シンボルには、次のように、スプール・ディレクトリを指定します。

```
sd=/usr/spool/lpd
```

8.6 プリント・フィルタ

Tru64 UNIX には、汎用およびローカル言語用のプリント・フィルタがあります。

8.6.1 `pcfof` プリント・フィルタ

`pcfof` プリント・フィルタは、プリンタ制御ファイル (PCF) を使用することにより、さまざまなプリンタに対処できるように設計されています。PCF ファイルには、用紙トレイ選択、両面、印刷の向きなど、プリンタ固有の機能の設定および選択を行うためのプリンタ制御文字列が含まれています。フィルタは、テキスト、ANSI、PCL、および自動選択多言語 PostScript プリンタで動作するように設計されていますが、PostScript 専用プリンタでは動作しません。

PCF ファイルはテキスト・ファイルです。任意のテキスト・エディタを使用することにより、既存のファイルを変更してプリント動作をカスタマイズしたり、サポートされていないプリンタ用に新しいファイルを作成したりすることができます。Tru64 UNIX で提供されている PCF ファイルは、アップデート・インストール中に置き換えられるため、カスタマイズしたファイルはバックアップに保存しておくようにしてください。新しい PCF ファイルまたは変更した PCF ファイルに対してファイル名の接頭語を使用すると、ファイル名の競合を避けることができます。たとえば、カスタマイズを行う前に、次のような名前でファイルをコピーします。

```
# cp ln32.pcf my_ln32.pcf
```

プリンタ・フィルタは `/usr/sbin` にあり、PCF プリンタに固有のファイルは `/usr/sbin/pcf` にあります。`template.pcf` ファイルは、PCF ファイル形式についてのドキュメントです。

8.6.2 `wwpsmf` プリント・フィルタ

`wwpsmf` フィルタは、プリンタ・カスタマイズ・ファイル (PCF) に定義されている設定を使って、ローカル言語文字のフォント・グリフを探し、

PostScript ファイル内のそのフォント・データを埋め込みます。このフィルタは、ローカル・システムに PostScript のアウトライン・フォントがインストールされていればそれを使用し、そうでなければ、フォント・サーバからビットマップ・フォントを持ってきて使用します。この機能は、国際化文字セットを含むプリント・ジョブをサポートしているので、このようなジョブを特別なプリンタに送る必要がありません。オプションのリストについては、`wwpsof(8)` を参照してください。ローカル言語プリンタのサポートについては、『国際化ソフトウェア・プログラミング・ガイド』および `il8n_printing(5)` を参照してください。

8.6.3 フィルタ使用における既知の制限事項

プリント・フィルタを使用する際の制限事項は、現在次のとおりです。

- TCP/IP プリントは、ローカル・サブネット内でプリントする場合に動作しますが、1 つ以上のルータを介するような複雑なネットワークでプリントすると、信頼性の問題が生じることがあります。ルータを識別するために、プリンタのネットワーク・カードを構成する必要があるかもしれません。詳細は、プリンタのマニュアルを参照してください。
- PostScript 形式でないファイルを PostScript および非 PostScript のフィルタを通してプリントすると、予想外の結果になることがあります。

表 8-7 に、このような問題が生じる恐れのあるフィルタを示します。

表 8-7: 非 PostScript および PostScript フィルタ

| フィルタ名 | フィルタ・タイプ |
|-----------|--------------|
| lpf | 非 PostScript |
| la75of | 非 PostScript |
| la324of | 非 PostScript |
| lqf | 非 PostScript |
| hplaserof | PostScript |

旧型のプリンタで期待どおりの出力を得るために、これらの非 PostScript のフィルタでは、ユーザが `/etc/printcap` ファイルで 020 (8 進) ビットを `fs` 制御変数に指定すると、改行の後に自動的にキャリッジ・リターンが追加されるという、シリアル・ポート・ドライバの機能に依存しています。

この制御ビットはネットワーク・ソケット・ドライバでは解釈されないため、シリアル・ポート・ドライバで提供されていたフォーマット処理がなくなります。したがって、プリンタ用に事前にフォーマットされていない非 PostScript 形式のファイルは、シリアル・ポート接続構成でのプリントと同じようにはプリントされない可能性があります。特に、この問題は、キャリッジ・リターンが埋め込まれていない ASCII 形式のテキスト・ファイルに影響する可能性があります。

- `lpf`、`la75of`、`la324of`、および `lqf` の非 PostScript フィルタを使用しているほとんどのプリンタでは、ネットワーク・インタフェース・カードをサポートしていません。しかし、シリアルおよびパラレル・ポートからネットワーク・ポートへのコンバータを使用するユーザにとっては依然として未解決の問題です (たとえば、HP RapidPrint ネットワーク・インタフェース・ボックスなどがあり、プリンタが TCP/IP プリンタのように動作します)。
- `hplaser4ps` の PostScript フィルタは、PostScript ファイルおよび事前にフォーマットされている非 PostScript 形式のファイル (PCL ファイルなど) に対して動作しますが、事前にフォーマットされていないファイル (キャリッジ・リターンが埋め込まれていない ASCII テキストなど) に対して使用すると、予想外の結果になる可能性があります。
- キャラクタ・セット (ASCII のような) のプリントに対して動作するように設計されているフィルタの中には、TCP/IP プリントでは動作しないものがあります。

8.7 プリンタのテストとトラブルシューティング

この節では、プリンタに関する障害の診断のためのチェックリストを示します。ほとんどのプリンタ・エラーは、`/var/adm/syslog.dated/current/lpr.log` ファイルにログが取られます。しかし、`/usr/adm/lperr` ファイルにログされるものもあります。

`printconfig` のウィンドウの「プリンタ設定: host」には、構成の終了後に直ちにプリンタへテスト出力を行うオプションがあります。出力がプリントされない場合は、この節で説明するトラブルシューティングの手順に従ってください。

`lpr` コマンドを使ってテキストを数ページ出力して、プリンタをテストすることもできます。PostScript や両面プリントなど、このプリンタでいつ

も使用する機能もテストする必要があります。詳細は、`lpr(1)` を参照してください。

`lptest` コマンドは、標準出力にリプル・テスト・パターンを出力します。テストパターンの出力先をプリンタに設定することもできます。各カラムに 96 個のプリント可能な ASCII 文字をすべて含むパターンが 96 行プリントされます。それぞれの文字は、次の行へ移るたびに右へ 1 文字ずれていきます。このテストは、各ページの行数と省略時のページ・パラメータを確認するのに便利です。リプル・テスト・パターンを使用して、プリンタ、ターミナル、およびデバッグ中の端末ポートのテストを行うこともできます。

`lptest` コマンドの構文は次のとおりです。

`/usr/sbin/lptest` [*length* [*count*]]

簡単なランダム・データが必要な場合は、`lptest` を使用して次のように出力します。

```
# /usr/sbin/lptest |lpr -P3r44
```

詳細については、`lptest(8)` を参照してください。

既存のプリンタで問題が発生したときや、システムにプリンタを追加したときは、次のようにして問題を診断します。

- 8.5.4 項 で説明しているエラー・ログ・ファイルを調べます。
- 物理的な接続をチェックし、可能であればケーブルを入れ替えます。
- 部品番号をすべてチェックして、ケーブルおよびコネクタが今の構成に対して適切であるかどうかを確認します。ケーブルの長さが、使用可能なボー・レートや通信方法に影響することがあります。
- ほとんどのプリンタには、内部テストおよびプリント・テストのオプションがあるので、ハードウェアの確認にはこれらのオプションを使います。
- プリンタ構成を見直し、デバイスにとって入力した値が適切かどうか確認します。エントリが正しいようなら、汎用またはパス・スルー・フィルタを使って単純な ASCII テキスト・ファイルをプリントしてみます。製造者のマニュアルを見て設定をチェックしてください。
`/etc/printcap` ファイルに正しい設定が記録されていることを確認します。 8.5.1 項 を参照してください。
- 次のコマンドを使って、プリンタ・デーモンが存在するかどうか確認します。

```
# ps agx | grep /usr/sbin/lpd
```

親 `lpd` プロセスが動作不能になっていたり、子プロセスが動作不能になっていることがあります。デーモンが実行されていない場合は、`kill -9` コマンドまたは 8.5.3 項にある停止コマンドと起動コマンド使って、各プロセスを強制終了して再起動する必要があります。

`-l` オプションを使用して `/usr/sbin/lpd` を実行することにより、デーモンにネットワークからの要求のログを取らせることができます。このオプションは、リモート・プリンタの問題をデバッグする際に便利です。詳細については、`lpd(8)` を参照してください。

- CDE グラフィカル・ツールまたは `lpc` コマンド行ユーティリティを使ってキューの状態とプリンタの状態を調べ、プリンタとキューが有効で使用可能になっていることを確認します。キューが滞っている場合は、キューをリセットしてみてください (8.4.4 項を参照)。プリント・ジョブが作成されキューに入っている場合は、他のローカルまたはリモートプリンタを構成してみてください。
- 適切なスプールまたはデバイス・ファイルが存在し、所有権とアクセスが正しいことを確認します (8.5.2.6.1 項を参照)。
- `/etc/printcap` ファイルやその他の構成上の問題を調べるには、`lpc check printername` コマンドを使用します。

ネットワーク接続のプリンタとリモート・プリンタに対しては、さらに、システムが適切に接続されしかもプリント・ジョブを転送する権限が与えられていることを確認する必要があります。ネットワークのトラブルシューティングについては、『ネットワーク管理ガイド：接続編』を参照してください。

アーカイブ・サービスの管理

システム管理者の重要な作業の 1 つに、損失または破壊したファイルの回復作業のサポートがあります。この作業を効果的に行うには、ファイルを頻繁にかつ定期的にバックアップするための手順を設定しておく必要があります。この章では、常駐のコマンドおよびユーティリティを使用してファイルやディレクトリをバックアップ (アーカイブ) およびリストアする方法について説明します。

重大な障害が発生した場合に、オペレーティング・システム全体およびユーザ・ファイルを正常な動作に復旧させるための手順を、障害復旧計画としてあらかじめ設計し、運用できるようにしておくことが大切です。障害復旧計画は、サイトの運用形態やビジネスの必要条件によって大きく異なりますので、この章では説明していません。ただし、バックアップは、このような計画を立てる際の重要な要素です。

この章では、以下の項目について説明します。

- バックアップ作成時の手順とオプションの概要 (9.1 節)
- バックアップ作成時の主な作業 (9.2 節)
- バックアップ・スケジュールの設定方法 (9.3 節)
- バックアップの作成方法 (9.4 節)
- 読んでおく必要のある他の参考資料、作成しておくシステム・ファイル、関連するユーティリティ、および事前に行うべき作業など、バックアップの準備に必要な情報 (9.5 節)
- バックアップを実行する `dump` コマンドの使用方法 (9.6 節)
- バックアップからデータを復元する `restore` コマンドの使用方法 (9.7 節)
- ファイル・システム全体でなく、個々のファイルやディレクトリのアーカイブが可能なコマンドの使用方法 (9.8 節)

- ファイルやディレクトリをアーカイブするためのグラフィカル・ユーザ・インタフェースである、`dxarchiver` の使用方法 (9.9 節)
- ルート・ファイル・システム，および障害復旧の際に役立つ重要なシステム・ファイルの，ブート可能なバックアップである，ブート可能テープの作成方法 (9.10 節)

9.1 バックアップ作業の概要

この章では，UFS ファイル・システムを使用しているシステムの基本的なバックアップ操作について説明します。使用しているシステムが次のいずれかの条件に該当する場合は，他のバックアップおよびリストア・ユーティリティも使用する必要があります。

- AdvFS (Advanced File System) ファイル・システムを単独で使用しているか，システムに接続されているディスクの一部で AdvFS ドメインを使用している場合は，『*AdvFS 管理ガイド*』を参照してください。AdvFS ファイル・システムを使用すると，ドメインのクローニングなどの，より多くのバックアップ機能を利用できます。UFS ファイル・システムの欠点の 1 つに，バックアップ中の UFS ファイル・システムへのアクセスを防がなければならないことがあります。バックアップ処理中にファイルにアクセスすると，そのファイルに行った変更がバックアップに記録されないことがあります。UFS ファイル・システムのバックアップを厳密に行うには，ディスクをオフラインにするか，システムをシャットダウンしてシングルユーザ・モードにしなければなりません。システムをシャットダウンするスケジュールを設定できない場合は，AdvFS ファイル・システムを使用してください。
- LSM (Logical Storage Manager) を使用している場合は，『*Logical Storage Manager*』を参照してください。ボリュームのミラーリングなどの LSM の機能を使用すると，UFS でのバックアップ上の制限のうちの一部が解消できます。たとえば，別のディスク上にファイル・システムのミラーリングを行うと，UFS ファイル・システムの正確なスナップショットを即座に得ることができます。その後，いつでもミラーを中断し，アーカイブを作成することができます。このとき，システムの運用の一時停止は，短くてすみます。LSM を使用するとディスク・スペースのスペアが必要になるため，ディスクの数が少ない小規模なシステムには適していません。

- ルート・ボリュームをバックアップして別のシステムへリストアする場合は、構成のクローニングが使用できます。この機能は、『インストレーション・ガイド — 上級ユーザ編』で説明されています。構成のクローニングを使用すると、障害時に別のプロセッサ上にカスタマイズ済みのオペレーティング・システムを再作成したり、1 つまたは複数のシステムに環境を再作成することができます。
- この章では、ベース・オペレーティング・システムをインストールしたときに提供されるバックアップ・ユーティリティおよびアーカイブ・ユーティリティについてだけ説明します。

「*Associated Products*」CD-ROM には、追加のバックアップ・アプリケーション (さらにライセンスが必要な可能性があります) が入っていることがあります。詳細は、『インストレーション・ガイド』を参照してください。サードパーティ製品の使用方法については、バックアップ・アプリケーションに付属のドキュメントを参照してください。

バックアップとリストア操作の主な作業は、以下のとおりです。

- データ回復と障害復旧の計画の作成
- 以下の手順で構成される、データのバックアップ
 - バックアップ・スケジュールの選択
 - `pax`, `tar`, `cpio` コマンド、または対応するグラフィカル・ユーザ・インタフェースである `dxarchiver` を使用した、小さいアーカイブの作成
 - `dump` ユーティリティによる、UFS のフル・バックアップの実行
 - 増分バックアップの実行
 - リモート・バックアップの実行
 - バックアップ・ツールの使用
- 特に以下のデータのリストア
 - 小さいアーカイブからのファイルのリストア
 - ダンプからのファイル・システムのリストア
 - 新しいパーティションでのダンプしたファイル・システムのリストア
 - ファイルのリストア
 - ファイルの対話式リストア

- リモート・リストアの実行
- ブート可能テープからのスタンドアロン・システムのリストア

9.2 データ・ファイルとシステム・ファイルのバックアップ

基本バックアップでは、`dump` および `restore` コマンドを使用できます。サポートされているすべてのコマンド・オプションについての詳細は、`dump(8)` を参照してください。Tru64 UNIX システムでは、アーカイブ、およびスタンドアロン・システム (SAS) のブート可能テープを作成するための、グラフィカル・ツールとコマンド行ツールもサポートされています。

バックアップと復元の手法で重要なこととして、データの損失を防止することがあります。データの損失につながる可能性のある状況の発生を予防する上で有用な、システムをモニタするツールが各種あります。たとえば、システムによっては環境のモニタリングをサポートしています。また、周辺機器のテストや試験を行なうツールもあります。また、イベントやエラーのロギング・システムもあり、バックアップ障害のような優先イベントに対してシステムをモニタするように構成できます。自身のシステムやサイトに、イベント報告の手法をセットアップする方法については、Event Manager の使用方法について説明している第 13 章を参照してください。Event Manager は、バックアップが成功したことの報告にも使えるので、スケジュールどおりにバックアップが行われたことを確認できます。

システムのすべてのファイル、ユーザ・ファイル、およびシステム・ファイルが失われないようにする必要があります。システム・ソフトウェアを含めて、システム全体をバックアップしておく必要があります。多くのシステム・ファイルは静的です。つまり、一度インストールされると、変更されることはありません。したがって、データ・ファイルほどには頻繁にバックアップする必要がありません。増分バックアップも可能です。これは、データが短時間に大きく変化する場合に検討する必要があります。

各ファイル・システムのバックアップは 1 つのプロセスとして実行されます。バックアップ・プロセスを簡単にするには、定期的にバックアップするファイル・システムに動的なファイルを置き、必要に応じてバックアップするファイル・システムには静的な (システムまたはプログラム) ファイルを置くように、ファイル・システム構成を設定します。しかし、必要に応じてバックアップを行うファイル・システムの中に、動的なファイルが混ざっていることがあります。この場合、動的なファイルを定期的にバックアップす

るには、定期的にバックアップするファイル・システムに動的なファイルをコピーしてから、バックアップを実行する必要があります。こうすることによって、ファイル・システム全体をバックアップすることなく、動的なファイルをバックアップできます。シェル・スクリプトを記述してこの処理を自動化することもできます。また、`cron` コマンドを使用してスケジュールを自動化することができます。詳細は、`cron(8)` を参照してください。

9.3 バックアップ・スケジュールの選択

各ファイル・システムのバックアップの間隔を決定する場合、ユーザの時間やデータが失われた場合の影響の大きさと、バックアップに要する時間を比較検討する必要があります。最低限保存しておくべき情報量が決まると、最低限のバックアップ間隔が決まります。ほとんどのシステムではバックアップは毎日行いますが、他の間隔も選択できます。

増分バックアップの場合は、バックアップのたびにファイル・システムの全ファイルをバックアップする必要はありません。前回のバックアップ以降変更されたファイルのみをバックアップします。`dump` コマンドと `restore` コマンドを使用して、最高で 9 レベルの増分バックアップが実行可能です。たとえば、レベル 0 のダンプは全体のファイル・システムをバックアップし、レベル 1 のダンプは、前回行ったレベル 0 のダンプ以降変更されたファイルをバックアップし、レベル 7 のダンプは、前回の下位レベルでのダンプ以降変更されたファイルをバックアップします。

ファイル・バックアップのスケジュールに増分バックアップを組み込むには、バックアップに要する時間およびテープ容量と、システム故障の際にシステムをリストアする時間とを比較検討する必要があります。たとえば、連続する 10 日間のバックアップのバックアップ・レベルは次のようにスケジューリングすることができます。

[0 1 2 3 4 5 6 7 8 9]

最初の日に、全体のファイル・システム (レベル 0) を保存します。2 日目に最初のバックアップ以降の変更内容を保存し、シーケンスを再開する 11 日目まで、その方式で継続します。こうすることによって、最初の日以外は、バックアップに費す時間と保存するデータ量を、少なくすることができます。しかし、10 日目にシステム障害が発生して、システム全体をリストアしなければならないとすると、10 本のテープすべてをリストアする必要があります。

ほとんどのシステムは、一般的なハノイの塔方式の変形バックアップ・スケジュールに従っています。定期的にバックアップを取るファイル・システムすべてを、レベル 0 で月に 1 回テープにダンプします。そして週に 1 回、レベル 1 のダンプを行い、それ以外の日は次のようなダンプ・レベルでバックアップを行います。

[...3 2 5 4 7 6 9 8 9 9 ...]

平日にバックアップを日に 1 回だけ行う場合、毎月のバックアップ・スケジュールは次のようになります。

[0 1 3 2 5 4 1 3 2 5 4 ...]

このスケジュールは少し複雑ですが、システム故障が 1 ヶ月のうちいつ発生してファイルが破壊されても、最高 4 本のテープをリストアするだけですみます。もちろん、毎日レベル 0 のダンプをすれば、いつでも 1 本のテープをリストアするだけですみますが、バックアップごとに大量の時間とテープの記憶領域を必要とします。ハノイの塔方式のスケジュールでは、ほとんどの日で、バックアップに必要な時間とテープの記憶領域が、非常に小さくなります。

9.4 バックアップの方法

データのバックアップには、必要性やローカル・システムの構成に応じて、次のようにいくつかの選択肢があります。

- 次のようなコマンド行インタフェースを端末から実行することができます。

- `dump`, `rdump`, `restore`, および `rrestore`
- `tar`, `pax`, および `cpio`

これらのコマンドは、ファイルのアーカイブを高速に作成したり、`cron` スケジューラで実行するスクリプトを作成するのに使用します。

- ブート可能テープ機能 `bttape` は、SysMan Menu アプリケーションであり、コマンド行、SysMan Menu または CDE から呼び出すことができます。呼び出し方に応じて、コマンド行インタフェース、またはウィンドウ環境に適したグラフィカル・ユーザ・インタフェースで実行されます（詳細は第 1 章を参照）。使用するコマンドは、`btcreate` および `btextract` です。

ブート可能テープ機能を使用して、回復用のブート可能テープの作成、および重要なシステム・データとカスタマイズしたシステム・ファイルのバックアップを行います。この機能は、任意の端末、および多数のウィンドウ環境から使用できるため、リモート操作に適しています。

- CDE のフォルダ「アプリケーション・マネージャ - システム管理」から、「ストレージ管理」フォルダを開き「ブート・テープ」アイコンをクリックします。これにより、`bttape` ユーティリティのグラフィカル・ユーザ・インタフェースが呼び出されます。
- CDE で、フロント・パネルから [アプリケーション・マネージャ] ポップアップ・メニューを開き、「デスクトップツール」フォルダを開いて、次のユーティリティを使用します。
 - プロジェクトまたはユーザ・アカウントをアーカイブする場合など、ファイルやフォルダを高速アーカイブするときには、「保管」を使用します。関連するインタフェースとして「保管リストの内容」および「保管ファイルの展開」があり、これらのアーカイブを管理することができます。これらは、いずれも最小限のオプションを備えた簡単なグラフィカル・インタフェースです。
 - CDE の「アプリケーション・マネージャ - システム管理」フォルダから、「日常管理」フォルダを開いてアーカイバ・ユーティリティを使用します。アーカイバは、コマンド行ツールに対するグラフィカル・ユーザ・インタフェースであり、圧縮などのアーカイブのタイプやオプションが選択できるようになっています。このインタフェースでは、ドラッグ・アンド・ドロップ操作でファイル・システム全体またはディレクトリ (フォルダ) をバックアップできます。

ツールによっては、スーパーユーザ (root) として実行すると、追加のオプションが利用できるものがあります。

9.5 バックアップ実行の準備

以降の項では、バックアップの準備に必要な情報について説明します。また、バックアップの準備に役立つユーティリティのリスト、および事前に行う作業についても説明します。

第 6 章では、UFS ファイル・システムについて説明しています。『ハードウェア管理ガイド』では、ディスクおよびテープ・デバイスの使用方法と、バックアップするディスクおよびテープ・デバイスの決定方法について説明

しています。定期的なバックアップのスケジューリングについては、第3章の `cron` コマンドの説明も参照してください。バックアップを実行する際に必要なその他の情報は、以降の項に記載されています。

9.5.1 関連ドキュメント

バックアップ・ユーティリティの使用に関する追加ドキュメントが、マニュアル、リファレンス・ページ、および オンライン・ヘルプ にあります。

9.5.1.1 マニュアル

以下のマニュアルは、アーカイブ・サービスについての役に立つ情報を含んでいます。

- AdvFS ファイル・システムおよび LSM ストレージ管理の機能については、『*AdvFS 管理ガイド*』および『*Logical Storage Manager*』で説明しています。
- 使用している周辺機器（たとえば、テープ・デバイス）のオーナーズ・マニュアルには重要な情報が含まれています。このようなマニュアルには、デバイスの記憶容量、メディア・タイプ、圧縮密度、および一般的な操作手順が記載されています。

9.5.1.2 リファレンス・ページ

各ユーティリティには、起動方法とオプションを説明したリファレンス・ページがあります。

以下のリファレンス・ページには、ファイル・システムをテープへダンプし、それをディスクへリストアする基本的なユーティリティについての情報があります。

- `dump(8)`, `rdump(8)`
- `vdump(8)`
- `restore(8)`, `rrestore(8)`

以下のリファレンス・ページには、アーカイブ・ファイルを作成したり操作するための基本的なユーティリティに関する情報があります。

- `tar(1)`
- `pax(1)`

9-8 アーカイブ・サービスの管理

- `cpio(1)`

以下のリファレンス・ページには、ブート可能テープ・インタフェースに関する情報があります。

- `btcreate(8)`
- `btextract(8)`
- `bttape(8)`

以下のリファレンス・ページには、特定の日に実行するバックアップ・スクリプトのための `cron` エントリの作成に関する情報があります。

- `cron(8)`
- `crontab(1)`

`mcutil(1)` のリファレンス・ページは、メディア・チェンジャ操作ユーティリティについて説明しています。

9.5.1.3 オンライン・ヘルプ

アーカイバおよびブート可能テープのグラフィカル・ユーザ・インタフェースには、オンライン・ヘルプがあり、ユーザのオプションを説明するとともに、各ウィンドウのデータ・フィールドに入力するデータを定義しています。

9.5.2 システム・ファイル

指定したファイル・システムや作成したアーカイブ・ファイルの他に、次のファイルがバックアップの作成時に使用または作成されます。

- `dump` コマンドおよび `restore` コマンドは、次のファイルを作成または使用します。

| | |
|-----------------------------|---|
| <code>/etc/dumpdates</code> | バックアップされたファイル・システムのリスト、各ファイル・システムがバックアップされた日付、およびバックアップ・レベルが記録されています。 |
|-----------------------------|---|

| | |
|---------------------------|-------------------------------|
| <code>/tmp/rstdir*</code> | 省略時のテープに格納されているディレクトリをリストします。 |
|---------------------------|-------------------------------|

`/tmp/rstmode*` 格納されているディレクトリの所有者、パーミッション・モード、タイムスタンプを記録します。

`./restoresymtab` 増分リストアまたは `rrestore` 操作中に必要な情報を保持します。

- ブート可能テープ機能は、次のファイルを作成または使用します。

`/var/adm/btcreate.log`

`btcreate` プロセスのログ。

`/usr/lib/sabt/sbin/custom_install.sh`

ミニルートに追加するファイルを指定する。

`/usr/lib/sabt/etc/addlist`

ブート可能テープ上に作成されるミニルート・ファイル・システムに追加するファイルおよびディレクトリを指定する、データ・ファイル。

`/usr/lib/sabt/etc/fslist`

バックアップするファイル・システムを指定する、データ・ファイル。

`/usr/run/bttape.pid`

複数の `btcreate` ユーティリティが同時に起動されないようにする、ロック・ファイル。

9.5.3 関連ユーティリティ

バックアップを実行するとき、次のユーティリティが便利です。

SysMan Station

SysMan Station を使用すると、システム上の使用可能なストレージ・デバイスをグラフィカルに表示できます。このインタフェースを使用すると、ディスクおよびテープ・デバイスを識別してそれらのデバイス名を調べるのに便利です。

**CDE のディスク使用
状況マネージャ**

CDE の「アプリケーション・マネージャ — デスクトップツール」フォルダには、「ディスクの使用状

況」ツールがあります。このツールは、`du` コマンドを実行してディスク使用量のデータを返します。
`/usr/users` などのディレクトリのサイズをブロック単位で調べるには、「フォルダのサイズ」オプションを使用してください。コマンド行ユーティリティの `du` および `df` は、それと同じデータを返します。

CDE のアプリケーション・マネージャ

CDE の「アプリケーション・マネージャ — 日常管理」フォルダには、「システム情報」インタフェースがあります。このツールは、ファイル使用量など、システム・リソースをグラフィカルに表示します。事前に設定されたファイル使用量の制限値を超えたときに警告を表示するように、このモニタを設定することができます。第 1 章の説明にあるように、`SysMan Station` を使用して、ファイル・システムをモニタすることもできます。

Event Manager

Event Manager は、ファイル・システムの制限を監視して問題を警告したり、自動的にファイル・システムのバックアップとクリーンアップを開始することができます。

`dsfmgr` および `hwmgr`

コマンド行インタフェースの `dsfmgr` および `hwmgr` により、デバイス名やディスクのパーティション・サイズなどのデバイス情報をシステムに照会することができます。

Disk Configuration GUI を使用しても、同様の情報を得ることができます。この GUI は、CDE の「アプリケーション・マネージャ — システム設定」フォルダまたは `SysMan Menu` から呼び出すことができます。このインタフェースでは、サイズの情報を M バイト、バイトおよびブロック単位で表示します。`disklabel` コマンドを使用すると、コマンド行インタフェースでディスクを構成することができます。

9.5.4 事前に行う作業

次に示す事前に行う作業は、すべてのバックアップ方法に共通のものです。

- インタフェースの使用方法およびコマンドの参考資料を確認する。この参考資料はシステムのダウン時でも利用可能である必要があります。シングルユーザ・モードで回復操作をたびたび行う必要がありますが、このモードではリファレンス・ページが利用できない可能性があります。
- 必要な製品またはユーティリティがすべてインストールされ構成済み (必要な場合) であることを確認する。これを最も簡単に確認するには、ツールの呼び出し方法について説明してあるリファレンス・ページを参照し、入力なしの状態でコマンド行インタフェースを呼び出すか、またはグラフィカル・ユーザ・インタフェースを起動してテストを実行します。
- テープ・ハードウェアがインストールされ構成済みであることを確認する。確認するには、`/usr/field/tapex` テープ・エクササイズを使用します。他のテスト機能についてハードウェア・マニュアルを参照してください。9.5.3 項でリストしたハードウェア情報ツールも参照してください。
- バックアップするディレクトリのサイズを調べる。たとえば、次のコマンドを使用して調べることができます。

```
# df /usr
Filesystem            512-blocks    Used   Available Capacity  Mounted on
/devices/disk/dsk0g  1498886     688192    660804      52%      /usr
# du -s -x /usr/users
1835      /usr/users
```

9.5.3 項でリストしたグラフィカル・ツールまたはコマンド行ツールを使用して調べることもできます。

- 正規のメディアに必要な数だけ用意して、バックアップするファイルを格納するのに十分な記憶容量があることを確認する。この確認は、ディスク、または WORM ドライブや光磁気ディスク・ドライブなどの他の書き込み可能メディアへアーカイブする場合にも必要です。
- アーカイブするファイルまたはディレクトリを確認し、アーカイブに適切な名前を付ける。異なるディレクトリを 1 つのボリュームにまとめてからアーカイブする場合には、一時的に作業用のディスク・スペースが必要になる場合があります。ただし、コマンド行を使って、アーカイブに直接まとめることもできます。または、既存のアーカイブにディレクトリを追加することによってもできます。使用するバックアップ・ユーティリティのマニュアルを参照してください。ツールによっては、省略時のファイル名とロケーションが提供されるものがあります。たとえば、ブート可能テープ・インタフェースでは次のファイル名が

プロンプト表示されます。省略時のファイル名を使用するか、または別のファイル名を入力します。

`/usr/lib/sabt/etc/fslist`

ミニルートに追加するファイルおよびディレクトリを指定する
データ・ファイル

`/usr/lib/sabt/etc/addlist`

バックアップするファイル・システムを指定するデータ・ファイル

アーカイバには次のファイルが必要です。

- 1 つ以上のソース・ファイルまたはディレクトリ。CDE では、ディレクトリはフォルダとして識別されるので、`/usr/lib/sabt/sbin` のような長いパス名を入力する代わりに、「File View」ウィンドウから「アーカイバ」ウィンドウにディレクトリをドラッグ・アンド・ドロップすることができます。
- デスティネーション・ファイル。ディスク上の tar ファイル `/usr/backups`、またはテープ・デバイスのデバイス名 `/dev/tape/tape0_d0`。なお、アーカイブ・ファイル名には拡張子 (接尾語) は不要です。必要なデバイス情報を得るには、9.5.3 項 にリストされているユーティリティを使用すると便利です。特に、システムに 2 つ以上のテープ・デバイスが接続されている場合に便利です。
- アーカイブをリストア (アンパック) する場合には、アーカイブ名 (`/usr/archives/userfiles_990802.z` や、テープ・アーカイブの場合は `/dev/tape/tape0_d0` など) を指定する必要があります。

- アクセスするデバイスのデバイス名と、関連するデバイス特殊ファイル。以下に有効なデバイス名とデバイス特殊ファイルの例を示します。

| デバイス名 | デバイス特殊ファイル | 説明 |
|--------|-------------------------------|-----------------------|
| dsk0a | <code>/dev/disk/dsk0a</code> | ディスク番号 0 のパーティション A |
| disk1b | <code>/dev/rdisk/dsk1b</code> | raw ディスク 1 のパーティション b |

| デバイス名 | デバイス特殊ファイル | 説明 |
|----------|---------------------|---------------------------------------|
| tape0c | /dev/tape/tape0c | 省略時設定の密度の巻き戻しをするテープ・デバイス (圧縮付き) |
| tape0_d0 | /dev/ntape/tape0_d0 | 巻き戻しをしないテープ・デバイス 0。_d0 接尾語は、密度を指定します。 |

デバイス名は、/dev ディレクトリの下の子・ディレクトリ /disk、/rdisk、/tape、または /ntape にあります。9.5.3 項にリストされているグラフィカル・ツールまたはコマンド行ツールを使って、デバイス名に合致したデバイスを探すこともできます。

注意

テープ・デバイスは、種々の記録密度および圧縮オプションを備えている場合が多いので、1 つのアーカイブに、より多くの情報を記録することができます。テープ記録密度のオプション、およびデバイス名によって記録密度を指定する方法については、tz(7) を参照してください。

- フル・バックアップでは、システムのシャットダウンが必要になることがあります。システムのバックアップは、システムがマルチユーザ・モードになっていてもシングルユーザ・モードになっていても可能です。ただし、現在変更が行われているファイル・システムでバックアップを行うと、バックアップ・データが壊れる可能性があります。dump コマンドは、バックアップするファイルの i ノードを検査します。i ノードには、テーブル・エントリおよび他の統計情報などのデータが含まれています。dump コマンドを使用してファイル・システム内のファイルをバックアップする場合は、各ファイルに i ノードが添付されます。ファイルの i ノードが記録された後、ファイルのバックアップを行わないうちにシステムまたはユーザがそのファイルを変更した場合は、バックアップ・データが壊れる可能性があります。

システムのシャットダウン、ファイル・システムのアンマウント、ファイル・システムの完全性の確認などは、次の手順で行ってください。

1. SysMan Menu の [一般的なタスク] オプション、または /usr/sbin/shutdown コマンドを使ってシステムをシャットダウン

ンします。たとえば、システムを 5 分以内にシャットダウンし、ユーザに対して警告メッセージを定期的に送りたい場合は、次のように入力します。

```
# /usr/sbin/shutdown +5 'System going down to perform backups'
```

システムのシャットダウンの詳細については、第 2 章を参照してください。

2. 次のように `umount` コマンドを `-a` オプションを付けて実行し、バックアップするファイル・システムをアンマウントします。

```
# /sbin/umount -a
```

ルート・ファイル・システムはマウントされたままです。

3. `fsck` コマンドを使用して、ファイル・システムの完全性を確認します。

たとえば、次のコマンドを使って `c` パーティション (ディスク全体) のファイル・システムを検査します。

```
# /sbin/fsck -o /dev/disk/dsk0c
```

9.6 dump コマンドの使用

`dump` コマンドは、指定されたすべてのファイル・システムや、指定の日付以降に変更のあった個々のファイルおよびディレクトリを、ファイル、パイプ、磁気テープ、ディスク、またはディスクセットにコピーします。AdvFS ファイル・システムのコピーについての詳細は、『*AdvFS 管理ガイド*』を参照してください。`dump` コマンドを使用するには、スーパーユーザの特権を持っている必要があります。

注意

ファイル・システムを正しくバックアップするには、ファイル・システムがアクティブでないときにバックアップを行う必要があります。ファイル・システムをアンマウントした後、ファイル・システムの一貫性を確認することをお勧めします。また、システムをシングルユーザ・モードにしてからバックアップの作業を開始することも、1 つの方法です。ただし、AdvFS ファイル・システムの場合はこれはあてはまりません。

9.6.1 フル・バックアップの実行

すべてのシステム・ソフトウェアを含むシステム全体の各ファイル・システムのフル・バックアップを実行するスケジュールを設定してください。システムのフル・バックアップの従来のスケジュールは、レベル0 ダンプをハノイの塔方式を使用して1ヶ月に1回実行するというものです。しかし、記憶媒体の信頼性に応じてスケジュールを設定することができます。たとえば、磁気テープの信頼性は約2年です。ファイル・システムをバックアップするには、`dump` コマンドを使用します。ブロック・サイズ、テープ記録密度、およびテープ長などのテープ・デバイスの特性を指定するために使用するコマンド・オプションについては、`dump(8)` を参照してください。`dump` コマンドを使うときは、絶対パス名でファイル・システムを指定します。`dump` コマンドはファイル・システムを一度に1つしかバックアップできませんが、数個の `dump` プロセスを同時に使用して、複数のテープ・デバイスにファイルを書き込むことができます。

以下に、`dump` コマンドでよく使用されるオプションを説明します。

- | | |
|---------------------------|--|
| <code>-integer</code> | ダンプ・レベルを整数 (0~9) で指定します。ダンプ・レベル0 では、指定されたファイル・システムのフル・ダンプが行われます。他のダンプ・レベルではすべて、増分バックアップが行われます。つまり、下位のダンプ・レベルで行った前回のダンプ以降変更されたファイルのみがバックアップされます。 <code>/etc/dumpdates</code> ファイルには、各ファイル・システムに対する <code>dump</code> コマンドの使用記録がダンプ・レベルとともに収められています。 <code>dump</code> コマンドに <code>-u</code> オプションを付けると、 <code>dumpdates</code> ファイルが更新されます。 |
| <code>-f dump_file</code> | 省略時のデバイス <code>/dev/tape/tape0_d0</code> ではなく、 <code>dump_file</code> で指定されたデバイスにダンプを書き込みます。 <code>dump_file</code> にダッシュ (-) を指定すると、 <code>dump</code> コマンドは標準出力に書き込みを行います。 |
| <code>-u</code> | バックアップ時に、 <code>/etc/dumpdates</code> ファイル内の、ファイル・システムに対するダンプ時間とダンプ・レベルを更新します。このファイルは、増分 |

バックアップの際にダンプ・レベルを調べるために使用されます。必要であれば、`/etc/dumpdates` ファイルを編集して、レコードやフィールドを変更することができます。詳細については、`dump(8)` を参照してください。

ファイル・システム全体を省略時のバックアップ・デバイスにバックアップするには、マシンの各ファイル・システムに対して `dump` コマンドを使用します。`dump -0u` コマンド・オプションは、レベル 0 でダンプを実行し、`/etc/dumpdates` ファイル内の各ファイル・システムのバックアップ時刻と日付を更新します。これにより、次回のフル・バックアップつまりレベル 0 ダンプまでのすべての増分バックアップの基準となる始点が作成されます。各ファイル・システムは、別々にバックアップしなければなりません。

たとえば、ルート (/)、`/usr`、および `/projects` ファイル・システムのパーティションに対してレベル 0 のダンプを実行したい場合、次の手順に従ってください。

1. ルート・ファイル・システムをバックアップするには、以下の手順を実行します。
 - a. テープをテープ・ドライブにロードします。
 - b. 次のコマンドを実行します。

```
# dump -0u /
```
 - c. バックアップ終了後、テープをテープ・ドライブから取り出します。
2. `/usr` ファイル・システムをバックアップするには、以下の手順を実行します。
 - a. 新しいテープをテープ・ドライブにロードします。
 - b. 次のコマンドを実行します。

```
# dump -0u /usr
```
 - c. バックアップ終了後、テープをテープ・ドライブから取り出します。
3. `/projects` ファイル・システムをバックアップするには、以下の手順を実行します。
 - a. 新しいテープをテープ・ドライブにロードします。

- b. 次のコマンドを実行します。

```
# dump -0u /projects
```

- c. バックアップ終了後、テープをテープ・ドライブから取り出します。

各ファイル・システムを個別のテープにバックアップすることも、出力デバイスとして `/dev/ntape/tape0_d0` を指定することによって、巻戻しをしないで 1 本のテープに複数のファイル・システムをバックアップすることもできます。次の例では、ルート (`/`)、`/usr`、および `/projects` ファイル・システムを 1 本のテープにバックアップします。

```
# dump -0uf /dev/ntape/tape0_d0 /
# dump -0uf /dev/ntape/tape0_d0 /usr
# dump -0uf /dev/ntape/tape0_d0 /projects
```

上記の例では、テープとダンプ・ファイルの相互参照、特に単一のダンプ・ファイルがメディアにまたがっている場合には、さらにメディア管理が必要です。この種のバックアップ・メディアへのラベル付けにも注意してください。

9.6.2 増分バックアップの実行

バックアップ・スケジュールの一環として、定期的なバックアップが容易にできるように定期的なバックアップのプロセスを確立します。バックアップとそのダンプ・レベルのログを取り、作成したテープの内容をリストする機構を組み込みます。システムが壊れる可能性もあるため、この情報はローカル・コンピュータ・システムに保持しない方がよいでしょう。

増分バックアップをするためのプロセスが確立すると、その後のバックアップ作業は簡単になります。次のバックアップ・スケジュールを使用して、`/usr` の毎日のバックアップを行うとします。

```
0 1 9 9 9 1 9 9 9 9 ...
```

月曜日に次のコマンドを使用して、レベル 0 ダンプを実行します。

```
# dump -0u /usr
```

火曜日に次のコマンドを使用して、レベル 1 ダンプを実行します。

```
# dump -1u /usr
```

レベル 1 ダンプは、月曜日以降に変更された全ファイルをバックアップします。水曜日から金曜日までは、次のコマンドを入力してレベル 9 のダンプを実行し、常に、火曜日のレベル 1 ダンプ以後変更された全ファイルをバックアップします。

```
# dump -9u /usr
```

省略時のテープ・デバイスではなく、`/dev/tape/tape1_d0` というテープ・デバイスに同じレベル 9 のダンプを実行するには、次の例に示すように `-f` オプションを使用します。

```
# dump -9uf /dev/tape/tape1_d0 /usr
```

`-f` オプションに対する引数には、ダンプを実行しているシステムのローカルなテープ・デバイスを指定します。

9.6.3 リモート・バックアップの実行

ネットワーク化されたシステム環境では、バックアップを取るために使用できるローカルのテープ・ドライブを持たないマシンもあります。この場合、`rdump` コマンドを使用して、リモートに配置されたテープ・デバイスにバックアップを取ることができます。`rdump` コマンドは `dump` コマンドとほとんど同じですが、マシン名およびそれに接続されているバックアップ・デバイスを指定するための `-f` オプションを必要とする点が異なります。`rdump` コマンドのオプションについては、`dump(8)` を参照してください。

`rdump` コマンドは、ローカル・マシンの `/etc/dumpdates` ファイルを `dump` コマンドと同じ方法で更新します。`rdump` コマンドは、リモート・マシン上でリモート・サーバ `/usr/sbin/rmt` を起動して、記憶媒体をアクセスします。このサーバ・プロセスは透過的です。詳細は、`rmt(8)` を参照してください。

`bhost2` に接続されているテープ・ドライブ `/dev/rmt0h` に `bhost1` から `/projects` ファイル・システムをバックアップするには、`bhost1` で次のコマンドを入力してください。アクセスするには、`bhost1` の名前が `bhost2` の `/.rhosts` ファイルに定義されていなければなりません。

```
# rdump -0uf bhost2:/dev/tape/tape0_d0 /projects
```

9.6.4 バックアップ・スクリプトの使用

シェル・スクリプトを使用してバックアップ・プロセスを自動化することができます。このシェル・スクリプトは、システムの変更によって `dump` コマンドがエラーを起こす可能性が少ない深夜に `cron` デーモンによって実行します。

バックアップ・シェル・スクリプトは、次のような処理に使用します。

- ダンプ・レベルの決定

- ダンプのシステムへの通知
- テープ内容のリストの作成
- オペレータに対する終了の通知

あらかじめテープ・ドライブにテープをロードしておきます。指定した時刻に、`cron` デーモンによってバックアップ・シェル・スクリプトが実行されます。シェル・プロシージャが終了したら、バックアップ・テープを取り外し、保管してください。

バックアップ・シェル・スクリプトは、ダンプが少量で、1本のテープに収まる場合に最も効率よく使用できます。`dump` コマンドに `-N` オプションおよび非巻戻し型のデバイスを指定して、各ダンプの完了時に自動的にテープがオフラインにならないようにする必要があります。`dump` コマンドがテープの終端に達すると、そのテープがオフラインになるため、誰かがテープを交換する必要があります。

9.7 データのリストア

バックアップ・テープからのファイルの取り出しが必要なことがあります。また、ファイル・システム全体のリストアが必要になることもあります。効率的なバックアップ手順を設定すると、ファイルまたはファイル・システム全体のリストアが簡単になります。

重大な問題が発生した場合には、システム全体をリストアする必要が生じます。リストアをする前に、システムの問題の原因を調べてください。

問題の原因が判明した後、最初のブート・テープからシステムを再インストールする方法があります。システムに付属のインストレーション・ガイドを参照してください。

システムを起動したら、システムをシステム・クラッシュの直前の状態に復元してください。AdvFS ファイル・システムを使用している場合は `vrestore` コマンドを使用します。AdvFS ファイル・システムのリストアについての詳細は『*AdvFS 管理ガイド*』を参照してください。UFS ファイル・システムをバックアップするのに `vdump` コマンドを使用した場合は、リストアするのに `vrestore` コマンドも使用することができます。ただし、`dump` コマンドを使用した場合、ファイルの回復には `restore` コマンドを使わなければなりません。`dump` コマンドは、一度に1つのファイル・システムしか保存しないため、リストアしたいファイル・システムそれぞれ

について `vrestore` コマンドを実行しなければなりません。コマンド構文については、`restore(8)` を参照してください。

9.7.1 ファイル・システムのリストア

ここでは、ディスク障害またはその他のデータ損失の後などに行う、ファイル・システムのリストアについて一般的な方法を説明します。個々のファイルのリストアについては 9.7.2 項を参照してください。

ファイル・システムをリストアするには、新しくファイル・システムを作成し、以下のコマンドを使ってダンプ・ファイルからファイル・システムをリストアします。

| | |
|----------------------|--|
| <code>newfs</code> | 新規に UFS ファイル・システムを作成する。詳細は、 <code>newfs(8)</code> を参照してください。 |
| <code>mount</code> | ファイル・システムをマウントし、一般に使用できるようにする。詳細は、 <code>mount(8)</code> を参照してください。 |
| <code>cd</code> | 現在の作業ディレクトリを変更する。詳細は、 <code>cd(1)</code> を参照してください。 |
| <code>restore</code> | バックアップ媒体からディスクへアーカイブ・ファイルをリストアする。詳細は、 <code>restore(8)</code> を参照してください。 |

AdvFS のリストアについては『*AdvFS 管理ガイド*』を参照してください。

ディスクにラベルがない場合は、`disklabel` コマンドを使用してラベルを書き込んだ後、新しいファイル・システムを作成する必要があります。詳細は、`disklabel(8)` を参照してください。

カスタマイズしたパーティション・テーブル設定でラベルを書き込むと、ディスク全体に影響する場合がありますので注意してください。次のコマンドで省略時のディスク・パーティション・テーブルを書き込みます。

```
# /sbin/disklabel -rw dsk1
```

`disklabel` コマンドの編集オプションを呼び出して、カスタマイズしたパーティション・テーブル設定を使用します。詳細については、第 6 章を参照

してください。ディスク設定インタフェースを使用することもできます。
詳細は、diskconfig(8)を参照してください。

次の例は、/usr/projects というファイル・システムをテープからリストアするためのコマンドを示しています。

```
# disklabel -rw dsk1
# newfs /dev/rdisk/dsk1c
# mount /dev/rdisk/dsk1c /usr/projects
# cd /usr/projects
# restore -Yrf /dev/tape/tape0_d0
```

9.7.2 手動によるファイルのリストア

データ・ファイルが失われた場合、ユーザはシステム管理者に対してファイルをリストアするよう依頼します。ファイルの以前のバージョンをリストアするように依頼することもあります。UFS ファイル・システムをリストアする場合、ファイル・リストアの理由にかかわらず、必要なバージョンのファイルの入っているテープを確認する必要があります。ファイルの損失時期および最後の変更時間を確認し、バックアップ・ログを調べることによって、どのテープに必要なファイルの最新バージョンが入っているかが分かります。

-t オプションを付けて restore コマンドを実行し、選択したテープに必要なファイルがあるかどうかを調べます。-t オプションは、テープに作成されたファイルとディレクトリのリストを作成します。たとえば、特定のバックアップ・テープにある /usr ファイル・システムの working サブディレクトリの内容をリストするには、テープをロードして次のコマンドを入力します。

```
# restore -t ./working
```

バックアップ・テープ全体の内容のリストを作成するには、バックアップ・テープをロードして次のコマンドを入力します。

```
# restore -t
```

バックアップ・テープ作成後に、それぞれのリストを作成しておいてください。これによって、バックアップの正常終了が確認され、そのテープにどのようなファイルが存在するか検索できます。

ファイルの位置が判明した後、そのファイル用に新しいディレクトリを作成してください。ファイルを既存のディレクトリにリストアしようとする

場合に、ファイルがすでに存在していると、既存のファイルはリストアされるファイルで上書きされます。

たとえば、`working/old.file` ファイルを `/usr` ファイル・システム・バックアップ・テープから現在のディレクトリにリストアするには、バックアップ・テープをロードして次のコマンドを入力します。

```
# restore -x ./working/old.file
```

同じテープから `working` サブディレクトリの全内容をリストアするには、次のコマンドを入力します。

```
# restore -x ./working
```

ダンプ用メディアに複数のダンプ・イメージがある場合、必要なファイルが何番目のダンプ・イメージに存在するかを知っていなければなりません。メディアの最初のダンプ・イメージの内容を調べるには、テープをロードして、次のコマンドを入力します。

```
# restore -ts 1
```

`-s` オプションの後に番号 1 を付けると、最初のダンプ・イメージが指定されます。

たとえば、バックアップ・テープの 3 番目のダンプ・イメージである `/usr` ファイル・システムの `working/old.file` ファイルを、現在のディレクトリへリストアするには、バックアップ・テープをロードして、次のコマンドを入力します。

```
# restore -xs 3 ./working/old.file
```

9.7.3 ファイルの対話式リストア

複数のファイルを簡単にリストアするためには、`restore` コマンドに `-i` オプションを付けて実行します。このオプションは、`restore` セッションを対話式で開始させます。対話式モードには、シェル・コマンドと類似したコマンドがあります。

対話式 `restore` セッションを開始するには、次のコマンドを入力します。

```
# restore -i
```

次のプロンプトが表示されます。

```
restore >
```

対話式リストア・モードでは、次のコマンド行オプションが利用可能です。

| | |
|--------------------------------------|--|
| <code>ls [<i>directory</i>]</code> | 現在のディレクトリまたは指定されたディレクトリのファイルをリストします。ディレクトリには後ろに / (スラッシュ) が付きます。読み取りが指定されているエントリには、先頭にアスタリスク (*) が付きます。 |
| <code>cd [<i>directory</i>]</code> | 現在のディレクトリを <i>directory</i> 引数で指定されたディレクトリに変更します。 |
| <code>pwd</code> | 現在のディレクトリのパス名をリストします。 |
| <code>add [<i>files</i>]</code> | 現在のディレクトリのファイルまたは <i>files</i> 引数に指定されたファイルを、テープから復元するファイルのリストに追加します。add コマンドで「読み取るように」指定すると、ファイルにアスタリスク (*) のマークが付きます。このアスタリスクは、ls コマンドでファイルをリストすると表示されます。 |
| <code>delete [<i>files</i>]</code> | テープから復元されるファイルのリストから、現在のディレクトリの全ファイルまたは <i>files</i> 引数で指定されたファイルを削除します。 |
| <code>extract</code> | 現在の作業ディレクトリに、「読み取るように」指定されているファイルをテープからリストアします。 extract コマンドは、マウントする論理ボリューム (通常は 1) を尋ねるプロンプトと、ドット(.) つまり現在のディレクトリのアクセス・モードを設定するかどうかを尋ねるプロンプトを表示します。root ディレクトリ全体をリストアする場合、yes と答えます。 |
| <code>setmodes</code> | 読み取りファイル・リストに追加された全ディレクトリについて、所有者、アクセス・モード、およびファイル作成時刻を設定します。ファイルはテープからまったく復元されません。restore コマンドが中断した場合に、このコマンドを使用してファイルを整理します。 |

| | |
|---------|---|
| verbose | 詳細報告 (verbose) モードに切り替えます。詳細報告モードでは、各ファイル名が標準出力にプリントされます。省略時の設定では、詳細報告モードはオフになっています。restore コマンドに <code>-v</code> オプションを付けることと同じです。 |
| help | 対話式コマンドの一覧をリストします。 |
| ? | 対話式コマンドの一覧をリストします。 |
| what | テープ・ヘッダ情報をリストします。 |
| quit | 対話式リストア・セッションを終了します。 |
| xit | 対話式リストア・セッションを終了します。xit コマンドは、quit コマンドと同じです。 |

バックアップ・テープからファイル `./working/file1` および `./working/file2` を対話形式でリストアするには、テープをロードして次のコマンドを入力してください。

```
# restore -i
```

対話式モードに入ったら、抽出したいファイルのリストに必要なファイルを追加し、リストアします。

1. ディレクトリを `working` に変更します。
たとえば、次のように入力します。

```
restore > cd working
```
2. プロンプトに対して次のようにファイル名を入力します。

```
restore > add file1
```
3. 次のように 2 番目のファイルの名前を入力します。

```
restore > add file2
```
4. 次のコマンドを使用して、ファイルを抽出します。

```
restore > extract
```
5. マウントする論理ボリュームの入力を求められます。

次の例のように、通常このプロンプトには 1 と答えます。

```
You have not read any tapes yet.  
Unless you know which volume your file(s) are on you can start  
with the last volume and work towards the first.
```

```
Specify next volume #: 1
```

次に、リストア時にドット (.) つまり現在のディレクトリのアクセス・モードを設定するかどうかを聞いてきます。例では n と答えています。

```
set owner/mode for '.'? [yn] n
```

6. ファイルが抽出された後、次のコマンドを入力して対話式セッションを終了させます。

```
restore > quit
```

上記の手順を実行すると、ファイル file1 および file2 が現在のディレクトリに存在します。

上記の手順を restore コマンドの -F オプションが読み取るコマンド・ファイルに記述して自動化することができます。たとえば、次の、restore_file というコマンド・ファイルは、前述の例で示したリストア操作を実行します。

```
cd working  
add file1  
add file2  
extract  
1  
n  
quit
```

このシェル・スクリプトを読み取って実行させるには、次のコマンドを入力します。

```
# restore -iF restore_file
```

このコマンドの実行結果は、前の対話式リストア・セッションのものと同じです。

9.7.4 ファイルのリモート・リストア

rrestore コマンドを使用してリモート・テープ・デバイスからローカル・ディレクトリにファイルをリストアします。rrestore コマンドでリモート・マシン名およびそのバックアップ・デバイスを指定するには、-f オプ

ションが必要です。詳細は `rmt(8)` を参照し、`rrestore` コマンドのオプションについての説明は 9.7 節を参照してください。

バックアップ・デバイスが接続されているリモート・システムの名前と、そのリモート・システムのバックアップ・デバイスの名前を `system:device` の形式で指定します。

バックアップ・デバイス `/dev/rmt0h` が接続されている `system2` にマウントされているバックアップ・テープから `./working/file1` ファイルを `system1` にあるローカル・ディレクトリにリストアするには、`system1` で次のコマンドを入力します。`system1` から `system2` にアクセスできるようにするには、`system1` の名前が `system2` の `/.rhosts` ファイルになければならないことに注意してください。

```
# rrestore -xf system2:/dev/tape/tape0_d0 ./working/file1
```

`rrestore` コマンドは、リモート・システム上でリモート・サーバ `/usr/sbin/rmt` を起動して記憶媒体にアクセスします。

9.7.5 システム (ルート) ディスクの回復または複製

以前のバージョンのオペレーティング・システムでは、SCSI バスのターゲットに従い、ドライブの物理的な位置をベースにデバイス名が割り当てられていました。Version 5.0 以降では、デバイス名は論理的に割り当てられ、データベースに格納されています。デバイス名とデバイスのバス・アドレスには、何も関連がありません。

ルート・ファイル・システムを正しくリストアしたり、ルート・ディスクを容量がより大きいディスクへ移動するには、デバイス・データベースを回復し、場合によってはアップデートしなければなりません。デバイスをテープ・バックアップ・メディアからリストアする場合は、デバイス (テープ・デバイスなど) をデバイス・データベースにインストールしなければならないこともあります。

システムをリブートしてリストアしている際に、次のメッセージが表示されることがあります。

```
Unable to save existing hardware configuration.  
New configuration will be used
```

このメッセージは、デバイス・データベースが回復不可能なため、リストアしなければならないことを示します。

次の手順は、ルート・ディスクの回復または複製(クローニング)を行う一般的な方法です。次のシナリオが考えられます。

- ディスクおよびルート・パーティションに損傷はないが、異なるディスク(より大きい容量の異なるモデルの場合もある)と交換したい場合。
- ルート・パーティションが置かれているディスク・ドライブに損傷があり、次の処理を行わなければならない場合。
 - 新しいディスク・ドライブ(タイプおよび容量が異なる場合もある)をインストールする。
 - すでにインストールされ使用可能な、代替ディスク・ドライブを選択する。
- ルート・ファイル・システム(/)と、場合によっては /usr ファイル・システムか /var ファイル・システムが壊れているが、これらのファイル・システムが置かれているディスク・ドライブが完全に機能している場合。

注意

この手順では、ネットワーク・バックアップからの回復方法は特に示していません。また、AdvFS ファイル・システムの回復についても特に示していません。詳細は、『*AdvFS 管理ガイド*』を参照してください。

次の項については、情報をご存知の場合は読まなくても構いません。

- 9.7.5.1 項で説明されている回復の準備方法
- 9.7.5.2 項で説明されている回復の要件
- 9.7.5.3 項で説明されている回復(または複製)手順
- 9.7.5.3 項で説明されている代替手順

9.7.5.1 回復または複製の準備

システムのセットアップ方法や、システムに対する知識に応じて、次の処理が必要です。

- ルート・ディスク・ドライブの交換。この手順は、オリジナルのルート・ディスクが使用できない場合、新しい交換用のディスクがインストールされているか、すでにシステムにインストールされている代替

ディスクの使用を決定していることを前提とします。ドライブのオーナーズ・マニュアルの説明どおりに、ドライブをインストールします。オペレーティング・システムが、自動的にドライブを検出します。

この手順では、新しいドライブまたは代替ドライブを識別するのに役立つステップがあります。

- ファームウェアのアップデート。

システムのファームウェアが最新版か確認します。この情報とダウンロード・キットは、次の URL の「*Firmware Updates*」Web ページで入手できます。

<http://ftp.digital.com/pub/DEC/Alpha/firmware>

(<http://www.hp.com> の Web サイトに行き、「Support」オプションを選択して、この情報を検索することもできます。)

- コンソール・コマンドについての情報の入手。

一部のタスクを実行するために、システム・コンソール・プロンプト(>>>) から Alpha System Reference Manual (SRM) コンソール・コマンドを使用します。これらのコマンドの説明は、AlphaServer system のオーナーズ・マニュアルにあります。

印刷されたドキュメントがない場合は、システムに付属の CD-ROM に収められている印刷可能ファイルを探してください。CD-ROM が利用できない場合は、次の URL の「*Alpha Systems Technology*」にこのドキュメントがあります。

<http://www.compaq.com/alphaserver/technology/index.html>

- この手順の指示は、新しいプロセッサに共通のものです。システムが古い場合は、オーナーズ・マニュアルと、ご使用のオペレーティング・システムのバージョンの『インストレーション・ガイド』を参照して、実際のコマンドと手順を調べてください。

システムの状態は、表 9-1 のとおりでなければなりません。

表 9-1: 回復の準備

| 要件 | 説明 |
|-------------------------|---|
| 最近のフル・バックアップ | ルート・ボリューム上にあるオペレーティング・システムのすべてのファイル・セットのフル・バックアップが必要。このバックアップには、ルート (/), /usr, および /var が含まれる。 |
| システム構成 | この手順は、/usr ファイル・システムと /var ファイル・システムも含むことがある、ルート・パーティションに使用されるディスクが 1 つ存在する構成すべてに適用される。リストアされたルート・ディスクを入れるために、正常に動作するディスク・ドライブが必要。このディスクは、オペレーティング・システムのリストア・リリースの制限値に定義されている、最小ストレージ・スペースを持たなければならない。リストア・デバイス (一般的にはテープ・ドライブ) はローカルでなければならない、リモート・バックアップ・デバイスは使用できない。 |
| Logical Storage Manager | Logical Storage manager (LSM) を使用している場合、ルート・ボリュームの回復について、『 <i>Logical Storage Manager</i> 』を参照。 |
| ユーザ・インタフェース | この手順では、コンソールにログインする必要がある。 |
| システムの可用性への影響 | クラスタ・システム以外では、ルート・ディスクの損失により、システムのシャットダウンおよびリブートが必ず 1 回以上必要になる。この手順は、できるだけ早く完全に操作できる状態に戻すのに役立つ。 ディスクの複製や回復に必要な時間は、ディスクのサイズによって異なる。 |
| 特権 | システムのストレージ・アレイやバックアップ・デバイスに物理的にアクセスできる、root ユーザでなければならない。 |

9.7.5.2 リストア要件の判断

ルート・ディスクのリストアを実行するには、次のリソースが必要です。システムの構成を熟知している場合、または回復を行うのに必要なすべての情

報が記録されている回復計画がある場合には、この項を読む必要はありません。次の項目が必要になる可能性があります。

- オペレーティング・システムの配布メディア

インストール・シェルを使用して、ルート・ディスクをリストアします。インストール・シェルはオペレーティング・システムの簡易バージョンで、mount などのコマンドを実行できます。このシェルは、配布キットの一部として、オペレーティング・システム・ソフトウェアと一緒にパッケージされています。

ローカル・サイトには、ネットワークからシステムをブートできる RIS (Remote Installation Service) サーバが用意されていることがあります。RIS サービスが CD-ROM メディアの代わりに利用できる場合は、サイト固有の手順に従ってください。また、『インストール・ガイド』を参照してください。

- CD-ROM ドライブ名またはネットワーク・デバイス名

ルート・ボリュームをリストアするには、CD-ROM ドライブかネットワーク・デバイスからシステムをブートします。必要であればオーナーズ・マニュアルを参照して、ご使用のシステムに合ったコマンドを見つけます。通常、コンソール・プロンプトで CD-ROM デバイス名を次のようにして調べます。

```
>>> show device | grep -E 'RR|CD'
dka400.4.0.5.0 DKA400 RRD47 1206
```

通常、ネットワーク・デバイス名は、次のようにして調べます。

```
>>> show config | more
```

前述のコマンドを入力すると、完全なシステム構成が、一度に 1 ページずつ表示されます。「Slot Option」という見出しのセクションまでスクロール・ダウンし、ネットワーク・デバイスを見つけます。ネットワーク・デバイスは通常、ew* または ei* (* は英字) という名前です。例を次に示します。

```
11 DE500-BA Network Con ewa0.0.0.0.11.0 08-00-99-1Z-67-BB
```

RIS サーバからのシステムのブートについては、『インストール・ガイド — 上級ユーザ編』を参照してください。SRM (System Reference Manual) コンソール・デバイスの命名規則の詳細は、システムのオーナーズ情報を参照してください。

- ブート・デバイス名

省略時のブート・デバイス名は、次の方法で調べます。

```
>>> show bootdef_dev
bootdef_dev          dka0.0.0.5.0
```

この例では、省略時のブート・デバイスは dka0 です。

注意

Fibre Channel を使用している場合、ブート・デバイス名は、ストレージ・デバイスの構成中に定義した名前です。

現在のルート・デバイスが使用可能で、同じデバイスにリストアしている場合、このデバイス名を後でリストア手順中に使用します。新しいディスクをインストールしたり、代替ディスクを使用する場合、ディスク名を指定しなければなりません。代替ディスクは、リストア手順中に *b/t/1* をブート・デバイス名に変換して調べます。

- バックアップ・メディアとリストア・デバイス

オリジナルのルート・ディスク上のファイル・システムによっては、ルート (/), /usr, および /var ファイル・システムの最新のフル・バックアップ・テープが必要となることがあります。

より大きな容量のディスクを使用することによって利用可能なディスク・スペースを増やすなどのために、ディスクの複製(クローニング)を行っている場合は、ソース・ディスクをターゲット・ディスクに直接バックアップすることができます。

- ターゲット・ディスクのディスク・ラベル

オリジナルのドライブが使用できる場合、ルート・ファイル・システムを同じドライブにリストアできます。このドライブが損傷している場合は、代替ドライブを選択するか、新しいドライブを設置しなければなりません。代替ドライブまたは新しいドライブには、リストア後のファイル・システムを保持できるだけの十分なストレージ容量があり、リストア後のファイル・システムを保持できるパーティションが作成されていなければなりません。

オリジナルのルート・ドライブにカスタム・パーティションがある場合、カスタム・ディスク・ラベルをリストアするか、少なくともリストア後のファイル・システムを保存できる大きさのパーティション(でき

れば、このファイル・システムの将来の拡張も考慮して)を選択するようにしてください。オリジナルのディスクに格納されたデータによって、以下に示すパーティションを計画する必要があります。

- ルート (/) ファイル・システムを保持するための、256MB の a パーティション。
- ルート・ドライブ上にある /usr ファイル・システムをリストアする必要がある場合、少なくともリストア後のファイル・システムを収めることができる大きさのパーティションも 1 つ必要です (必要に応じて、拡張要件を検討してください)。
- ルート・ドライブ上にある /var ファイル・システムをリストアする必要がある場合、少なくともリストア後のファイル・システムを収めることができる大きさのパーティションも 1 つ必要です (このファイル・システムがオリジナルのルート・ドライブ上で /usr に含まれている場合を除く)。
- 1 次スワップ・パーティションか 3 次スワップ・パーティションがオリジナルのルート・ドライブに存在していた場合、これらのパーティションを代わりのドライブ上に再作成する必要があります。

必要に応じて、他のファイル・システムもリストアできます。また、他のファイル・システムを別のデバイスにリストアし、リストアの完了後に /etc/fstab ファイルをアップデートしてマウントし直すこともできます。

カスタム・パーティションのディスク・ラベルの作成についての詳細は、disklabel(8) を参照してください。

9.7.5.3 手順の適用

この手順の各ステップには、システムのオリジナルの構成によって異なるものがあります。ご使用の構成に合わない場合は、そのステップを省略して構いません。省略可能なステップには、/構成に依存/ というマークが付いています。

この手順では、行先が明記されていないかぎり、必ず次のステップへと進みます。

1. 次のいずれかの方法で、オペレーティング・システムの配布メディアからシステムをブートします。

- ・ リストアしたいバージョンのオペレーティング・システムが入っている配布 CD-ROM を挿入し、以前に調べた CD-ROM リーダのデバイス名を指定してオペレーティング・システムをブートします。たとえば、次のコマンドを入力します。

```
>>> boot dka400
```

- ・ ローカル RIS サーバからブートします。たとえば、次のコマンドを入力します。

```
>>> boot ewa0
```

2. /構成に依存/すでに文字セル・インストレーション・プロシーダを使用している場合は、ステップ 3 に進みます。使用していない場合は、次の作業を実行します。

システムにグラフィック・コンソールがある場合、インストレーションは省略時の設定でグラフィック・モードになります。インストレーション・プロシーダが「Installation Welcome」というタイトルのダイアログ・ボックスが表示されるまで待ちます。

[ファイル] メニューをプルダウンし、[終了] を選択して、文字セル・モードを起動します。

3. バックアップ・デバイスとターゲット・ディスク (リストア後のルート・ディスク) のステータスを、次のコマンドを使用して確認します。

```
# hwmgr view devices
```

hwmgr コマンドは、次の例のように、現在システムが認識しているすべてのデバイスのリストを表示します。

| HWID: | Device Name | Mfg | Model | Location |
|-------|--------------------|--------|---------------|--------------------|
| 4: | /dev/kevm | | | |
| 28: | /dev/disk/floppy0c | | 3.5in floppy | fdi0-unit-0 |
| 31: | /dev/disk/dsk0c | DEC | RZ26L (C) DEC | bus-0-targ-0-lun-0 |
| 32: | /dev/disk/dsk1c | DEC | RZ26 (C) DEC | bus-0-targ-1-lun-0 |
| 33: | /dev/disk/dsk2c | COMPAQ | HB00931B93 | bus-0-targ-3-lun-0 |
| 34: | /dev/disk/cdrom0c | DEC | RRD45 (C) DEC | bus-0-targ-4-lun-0 |
| 35: | /dev/disk/dsk3c | COMPAQ | HB00931B93 | bus-0-targ-5-lun-0 |
| 37: | | DEC | TLZ06 (C) DEC | bus-0-targ-6-lun-0 |

次のデータを探し、書き留めておきます。

- ・ ターゲット・ディスクのデバイス名。

代わりの新しいディスクを取り付けた場合は特に重要です。

このデバイス名前は、「Device Name」欄の下のエントリ

(/dev/disk/dsk2 など) です。パーティションのサフィックス

(c) は無視します。

新しく設置したターゲット・ディスクのエントリがない場合は、進むことができません。システムをシャットダウンし、ディスクの物理的な取り付け状況を確認してから、回復手順を再度実行してください。

- バックアップ・デバイスのデバイス名。

デバイス名は、「Device Name」欄に表示されます。たとえば、省略時のテープ・デバイス TLZ06 からルート・ディスクをリストアしている場合、「Device Name」欄には、`/dev/ntape/tape0` というデバイス特殊ファイル名が表示されます。

バックアップ・デバイスの「Device Name」欄には、前述の例で示すように、デバイス特殊ファイル名が表示されないことがあります。この場合、ステップ 4 に進んでテープ・デバイスをインストールします。

4. 次のコマンドを使用して、バックアップ・デバイスをインストールします。

```
# dn_setup -install_tape
```

インストールされたこととデバイス名 (たとえば `tape0_d0`) を確認するには、ステップ 3 の `hwmgr` コマンドを繰り返します。

5. /構成に依存/オリジナルのファイル・システム・フォーマットが分からない場合は、次の手順で、フォーマットと、読み取り可能なバックアップ・テープであることを確認します。

- a. バックアップ (ダンプ) メディアをデバイスにロードします。
- b. ステップ 4 で調べたバックアップ・デバイス名を指定して、`restore` コマンドを対話型モードで起動します。たとえば、次のコマンドを入力します。

```
# restore -i -f /dev/ntape/tape0_d0
```

- c. バックアップの状態が良い場合は、対話型リストアのプロンプトが表示されます。 `what` コマンドを入力してヘッダを表示し、情報を書き留めます。

6. 次の情報を使用して、ディスク・ラベルを作成し適用します。

- a. リカバリ計画の際に作成したパーティション計画 (スワップ領域の要件を含む)。 (9.7.5.2 項を参照)

b. ステップ 3 で調べた新しいルート・デバイス名。

a パーティションを指定し、ブート可能デバイスとしてドライブにラベルを付けます。たとえば、次のコマンドを入力します。

```
# disklabel -wr /dev/disk/dsk2a
```

7. 次のようにして UFS ターゲット・ファイル・システムを作成します。

リストアする必要のある各ファイル・システムに対して、新しいルート・ドライブ上にファイル・システムを作成しなければなりません。たとえば、a パーティションと g パーティションに新しいルートと /usr ファイル・システムを作成するには、次のようなコマンドを使用します。

```
# newfs /dev/disk/dsk2a
# newfs /dev/disk/dsk2g
```

8. 代替りのディスクを、ファイル・システムのタイプに応じて、一時マウント・ポイント /mnt にマウントします。たとえば、次のコマンドを入力します。

```
# mount /dev/disk/dsk2a /mnt
```

9. vrestore コマンドまたは restore コマンドを使用して、ファイルをリストアします。たとえば、次のコマンドを入力します。

```
# cd /mnt
# vrestore -x device
```

10. 次のコマンドを使用して、システムをシャットダウンし停止します。

```
# shutdown -h now
```

11. リストアされたルート・ドライブをブート・デバイスとして指定して、シングルユーザ・モードでシステムをブートします。たとえば、次のコマンドを入力します。

```
>>> boot dka2 -flags s
```

代替ドライブを使用している場合、または新しいドライブをインストールした場合は、システム・デバイス名を適切なブート・デバイス名に変換しなければならないことがあります。新しいデバイスのデバイス・データベース・エントリは、hwmgr コマンドを使用して、ステップ 3 で調べました。エントリの例を、次に示します。

```
33: /dev/disk/dsk2c    COMPAQ    HB00931B93    bus-0-targ-3-lun-0
```

デバイスを表示するには、次のコマンドを使用します。

```
>>> show device
```

`b/t/1` の値 (この場合 0.3.0) を代替デバイスまたは新しいデバイスにマッピングし、`dka300` などのブート・デバイス名を確認します。

12. ブートが成功した場合、次のスクリプトを実行して、デバイス・データベースをアップデートします。

```
# /sbin/mountroot
```

`dsfmgr` コマンドがデバイス・データベースをアップデートしようとした際に、いくつかのエラーまたは警告のメッセージが表示されることがあります。このメッセージは無視してください。

13. /構成に依存/`root` 用に新しいドライブをインストールした場合や、代替デバイスを指定した場合、デバイスの名前を変更しなければなりません。ステップ 3 で調べたデバイス名情報を使用して、デバイスの名前を次のように変更します。

- 古いルート・ディスクを削除し、新しいデバイスと交換した場合は、`-m` オプションを指定して `dsfmgr` コマンドを実行し、デバイス名を移動します。たとえば、次のコマンドを実行します。

```
# dsfmgr -m dsk20 0
```

- 古いルート・ディスクがシステムに接続されたままの場合は、`-e` オプションを指定して `dsfmgr` コマンドを実行し、デバイス名を交換します。たとえば、次のコマンドを入力します。

```
# dsfmgr -e dsk20 0
```

14. `vrestore` コマンドの対話型モードを使用して、バックアップ・メディアをリストア・デバイスにロードし、デバイスのディレクトリをリストアします。このステップにより、カスタム・デバイス・ドライバを含む、すべての適切なデバイスが再作成されるようになります。

- 次のコマンドで、既存のディレクトリを削除します。

```
# rm -rf /cluster/members/member0/dev*
```

- `dump` コマンドを使用した場合は、次のようにしてディレクトリをリストアします。

```
# restore -i -f /dev/ntape/tape0_d0
restore > add /cluster/members/member0/dev
restore > add /cluster/members/member0/devices
restore > extract
```

15. `dsfmgr` コマンドを次のように使用して、デバイス・データベースとデバイス特殊ファイル名を確認します。

```
# dsfmgr -v
```

16. この手順はこれで終わりです。必要なファイル・システムすべて (`/usr` および必要であれば `/var` を含む) のリストアが終わっていれば、次のようにして、システムをシャットダウンし、ブート・デバイスを再定義し、システムをマルチユーザ・モードでリブートできます。

```
# shutdown -h now
>>> set bootdef_dev dka300
>>> boot
```

ブート処理でデバイスに関連するエラー・メッセージが表示されていなければ、成功したことを確認できます。この手順が成功していないと思われる場合は、配布メディアからオペレーティング・システムを再インストールし、カスタマイズされた環境を再作成する方法しかありません。

9.7.5.4 その他のルート・ディスク複製方法の使用

ルート・ディスクのリストアまたは複製の他の方法は、このような障害を想定してシステムを構成しているかどうかにより異なります。次の方法が利用できます。

- LSM や AdvFS を使用している場合に利用できる、ミラーリングのように、ルート・ディスクを複製する方法。詳細は、『*AdvFS 管理ガイド*』および『*Logical Storage Manager*』を参照してください。
回復可能なルート・ボリュームのセットアップについては、『*Logical Storage Manager*』を参照してください。
- ブート可能テープによる、通常のバックアップ・テープよりも高速の、カスタム・システムのリストア方法。この機能は、一部の構成でだけ利用可能です。詳細については、第 9 章を参照してください。
- カスタム・システム・ファイルのバックアップ。このバックアップは、オペレーティング・システムを再インストールしなければならない場合の回復に役立ちます。以前のインストールに対してアップデート・インストール機能 (`updateinstall`) を使用した場合は、カスタム・システム・ファイルのアーカイブを作成できます。詳細については、『*インストール・ガイド*』を参照してください。

9.7.6 /usr および /var ファイル・システムのリストア

/usr ファイル・システムをリストアする前に、9.7.5 項で示すようにルート・ファイル・システムをリストアする必要がある可能性があります。/var ディレクトリが、/usr 以外のファイル・システムにある場合、/var ファイル・システムのリストアにはこの項の手順を繰り返してください。

この項の手順では、/usr ファイル・システムの最新ダンプ・ファイルへアクセスすることを前提としています。次の手順では、作業を行うのにテキスト・ベース (つまり、文字セル) のインタフェースを使って、ファイルのレベル 0 ダンプからリストアする方法を示します。

1. `disklabel` コマンドを使い、必要に応じてディスクにラベルを付けるには、次のコマンドを入力します。

```
# disklabel -rw /dev/disk/dsk0
```

注意

この手順で `disklabel` コマンドに指定されているオプションでは、省略時のディスク・パーティション・テーブルがディスクに書き込まれます。リストア中のディスクがカスタマイズされたパーティション・テーブルを持っている場合は、`disklabel` コマンドの編集オプションを呼び出します。または、以前のファイル・ラベルからパーティション・テーブルをリストアします。詳細は、第 6 章 および `disklabel(8)` を参照してください。

2. `newfs` コマンドを使用して、新しいファイル・システムを作成します。たとえば、次のコマンドを入力します。

```
# newfs /dev/rdisk/dsk1c
```

3. `mkdir` コマンドを使い、必要に応じてマウント・ポイントを作成します。たとえば、次のコマンドを入力します。

```
# mkdir /usr
```

4. `mount` コマンドを使い、ファイル・システムをマウントします。たとえば、前の手順で作成したファイル・システムをマウントするには、次のコマンドを入力します。

```
# mount /dev/disk/dsk1c /usr
```

5. ファイル・システムをリストアします。

- ローカル・ファイル・システムからダンプ・ファイルをリストアする場合

リストア・ディレクトリに移動してダンプ・ファイルが入っているメディアを挿入し、`restore` コマンドを入力します。テープの場合には、次のコマンドを入力します。

```
# cd /mnt
# restore -Yrf /dev/tape/tape0_d0
```

- リモート・システムからダンプ・ファイルをリストアする場合

リストア・ディレクトリに移動して、`rsh` コマンドを入力します。次のコマンド・オプションを指定します。

remote_hostname ダンプ・ファイルのあるリモート・システムのホスト名

dumpfile リモート・システムにあるダンプ・ファイルの絶対パス名

blocksize テープからのデータの読み取りに必要なブロック・サイズ

テープに書き込んだときに使用したものと同じブロック・サイズでダンプ・ファイルを読み取ります。省略時のダンプ・レコード・サイズは 10K バイトです。

たとえば、省略時のブロック・サイズを使用して書き込まれた `remotesystem` というリモート・システムからダンプ・ファイルを TLZ06 にリストアするには、次のコマンドを入力します。

```
# cd /mnt
# rsh remotesystem "dd if=/dev/tape/tape0_d0 bs=10k" \
| restore -Yrf -
```

9.8 コマンド行ユーティリティ `tar` , `pax` , および `cpio` の使用

コマンド行ユーティリティ `tar` , `pax` , および `cpio` を使用すると、コマンド行からアーカイブを高速に作成したり、ファイルをバックアップするためのスクリプトを記述することができます。難点は、長いコマンド文字列を入

力しなければならないので、大容量のファイルやディレクトリをバックアップしたりリストアする場合に、このインタフェースでは簡単にいかないことです。これらのユーティリティは、ファイルの小さなアーカイブを作成し、プログラムや、プログラムのソース、および関連ドキュメントなどを小さなアーカイブ・ファイルにして他のユーザに配布する場合に使用されます。

次の例では、コマンド行ユーティリティ `tar`、`pax`、および `cpio` を使ってアーカイブ・ファイルを作成したり、リストアする方法について、典型的な例を示します。

tar を使用したアーカイブの作成

`tar` コマンドは、ディスクやテープなどの1台のデバイス上に、複数ファイルを保存したりリストアします。

`tar` を使ってテープ・ドライブ `/dev/tape/tape12_d0` にアーカイブを作成するには、次のようなコマンドを入力します。

```
# tar cvfb /dev/tape12 -e ../netscape -C /usr/glenn
```

作成されたアーカイブには、`../netscape` ファイルを除き、`/usr/glenn` ディレクトリの下すべてのファイルとディレクトリが含まれます。詳細については、`tar(1)` を参照してください。

pax を使用したアーカイブの作成

`pax` コマンドは、アーカイブ・ファイルのメンバの抽出、書き込み、およびリストを行います。また、`pax` コマンドは、ファイルおよびディレクトリの階層をコピーします。

`pax` コマンドを使って現在のディレクトリのアーカイブを `/dev/tape/tape0_d0` デバイスに作成するには、次のように入力します。

```
# pax -w -f /dev/tape/tape0
```

次のコマンドは、`a.pax` アーカイブを読み取り、ディレクトリ `/usr` 内のすべてのファイルを抽出して現在のディレクトリ内に置きます。

```
# pax -r -s ' ,^/*usr/* ,,' -f a.pax
```

詳細については、`pax(1)` を参照してください。

cpio を使用したアーカイブの作成

cpio コマンドは、アーカイブ・ストレージとファイル・システム間でファイルをコピーします。このコマンドは、従来の形式の cpio アーカイブのデータを保存したりリストアするのに使用します。

cpio コマンドを使ってテープ・デバイス /dev/tape/tape12_d0 にアーカイブを作成するには、次のように入力します。

```
# cpio -ov < file-list -O/dev/tape12_d0
```

詳細については、cpio(1) を参照してください。

9.9 dxarchiver の使用

アーカイバ (dxarchiver) は、9.8 節で説明したコマンド行ユーティリティのためのグラフィカル・ユーザ・インタフェース (GUI) です。このインタフェースは、次のような場合に使用します。

- 複数ファイルを 1 つの指定されたアーカイブ・ファイルに格納する、またはテープやディスクなどの出力デバイスへ出力する場合
- 受取ったアーカイブ・ファイルを解凍する場合や、新しく作成したファイルを圧縮する場合
- アーカイブ・ファイル、またはテープやディスクなどのデバイスから、格納されているファイルを取り出す場合

Archiver GUI は CDE アプリケーションなので、長いコマンドを入力することなく、ファイルおよびディレクトリ (フォルダ) をドラッグ・アンド・ドロップ操作で、アーカイブ・セットにまとめることができます。

以下の作業を行なうにあたって、あらかじめ 9.5.4 項に示す情報を収集します。また、オーナーズ・マニュアルに従って、ターゲット・デバイスにテープやその他のメディアをロードまたはアンロードしておきます。アーカイブを作成するには、以下の手順を行ないます。

1. 端末からコマンド行を使用して /usr/bin/X11/dxarchiver GUI を呼び出します。または CDE の「システム管理」アプリケーション・グループを開いて、次に「日常管理」サブ・グループを開き、「アーカイバ」アイコンをクリックします。

2. アーカイブ・タイプを `tar` , `cpio` , `pax` の中から選択します。グラフィカル・ユーザ・インタフェースからは利用できないコマンド行オプションがあります。
3. アーカイブ・オプションを選択します。追加は、既存のアーカイブに対してのみ行うことができます。また、作成時に圧縮してあるアーカイブは、さらに圧縮することはできません。ディレクトリの格納方法として、絶対パス名または相対パス名のどちらかを指定します。なお、絶対パス名は完全パスで、`/usr/users` のようにルート・ディレクトリから始まります。相対パス名は、たとえばドット (.) または `users/chan` のように、現在のディレクトリから始まります。

これらのファイルの今後の復元の際、オリジナルのアーカイブ処理中に相対パスを指定していた場合のみ、これらのファイルを一時的なロケーションに書き込むことができます。絶対パスを指定していた場合は、ファイルは元のロケーションに書き込まれます。名前を変えていない場合は、知らないうちに既存のファイルに重ね書きされます。

4. アーカイブするソース、ファイルおよびディレクトリを指定します。指定するには、パス名を入力するか、あるいは「ファイル・マネージャ」ビューを開いてファイルとディレクトリ (CDE フォルダ) を「アーカイバ」ウィンドウ内の「ソースの内容」ボックスにドラッグします。パス名を入力した場合は、`OK` を選択してコンテナにそれらを追加します。
5. 必要なファイルをすべて指定した後、[アーカイブ...] オプションを選択すると、「アーカイバ: アーカイブ」ウィンドウが表示されます。
6. デスティネーション・パスを次のように入力します。
 - 省略時のテープ・デバイスの場合は、`/dev/tape/tape0_d0` と入力します。
 - ディスク・アーカイブの場合は、`/usr/backup/myback_991803` と入力します。ファイル名に拡張子は不要です。アーカイバが `.z` のような識別子を追加します。

[了解] を選択すると、「デスティネーションの内容」ボックスの下にデスティネーションが表示されます。

7. 「アーカイブの作成」をクリックします。「Archiver working」というタイトルの付いたウィンドウが表示され、緑のボタンが点滅して、アーカイブが書き込み中であることを示します。アーカイブ中のファイルは、「デスティネーションの内容」に表示されます。

8. アーカイブが完了したら、オプションでファイル・リストのコピーを出力できるので、テープと一緒に記録を保管しておくことができます。
9. [取消] を選択してアーカイバのメイン・ウィンドウに戻ります。アーカイブ・ファイルの名前を入力し、「内容の一覧...」オプションを使用してアーカイブが正しく書き込まれていることを確認できます。テープまたはアーカイブ・ファイルが読み込まれ、内容が「内容の一覧」ウィンドウに表示されます。

アーカイブを復元するには、ディスクのようなターゲット・デバイス上に、デスティネーションを指定する必要があります。ディスク・パーティション全体に、損失したファイル・システムを復元する場合以外は、既存のディレクトリに重ね書きするのではなく、一時的なロケーションを使用しても構いません。その後、個々のファイルやディレクトリを必要に応じてリストアすることができます。次のように、アーカイブから選択したファイルだけをリストアすることもできます。

1. 端末からコマンド行で `/usr/bin/X11/dxarchiver GUI` を呼び出さか、または CDE の「システム管理」アプリケーション・グループを開きます。次に、「日常管理」サブ・グループを開いて「アーカイバ」アイコンをクリックします。
2. [内容の一覧...] を選択し、個々のファイルおよびディレクトリを選択します。テープまたはアーカイブ・ファイルが読み込まれ、内容が「内容の一覧」ウィンドウに表示されます。次のように個々のファイルまたはディレクトリを選択します。
 - 「アーカイバ: 内容の一覧」ウィンドウで、ファイルまたはディレクトリを選択してそれを強調表示します。
 - 別のファイルまたはディレクトリを、Ctrl キーを押しながらクリックして選択します。
 - 必要なファイルをすべて選択し終わったら、「アーカイバ: 内容の一覧」ウィンドウの [了解] ボタンをクリックします。アーカイバのメイン・ウィンドウ内の「ソースの内容」ボックスに選択したファイルが表示されます。選択を変更したい場合には、[編集] メニューを使います。たとえば、「ソースの内容」内のエントリを強調表示して [編集] メニューから [選択したソースをクリア] を選択すると、そのエントリが削除されます。

3. [展開...] オプションを選択して、「アーカイバ: 展開」ウィンドウを表示します。
4. デスティネーション・ディレクトリを入力します。このディレクトリは、ファイルが重ね書きされてもよければ、アーカイブと同じにできます。または、その代わりに一時的なロケーションのパスを入力します。このパスは既に存在するディレクトリである必要があります。存在しないディレクトリを使用する場合は、端末ウィンドウを開いて `mkdir` コマンドを使ってディレクトリを作成するか、あるいは CDE の「ファイル・マネージャ」の「新規フォルダ」オプションを使ってフォルダを作成します。選択したデスティネーションは、「デスティネーションの内容」ボックスの下に表示されます。
5. [内容の展開] を選択して抽出を開始します。「Archiver Working」というタイトルが付いたウィンドウが表示され、緑のボタンが点滅してアーカイブが抽出中であることを示します。復元中のファイルは、「デスティネーションの内容」に表示されます。
6. アーカイブの復元が完了したら、オプションでファイル・リストのコピーを出力できるので、記録として保管しておくことができます。
7. [取消] を選択してアーカイバのメイン・ウィンドウに戻ります。終了前に、「ファイル・マネージャ」または端末のウィンドウを使用して、ファイルが正しく復元されファイルの内容が壊れていないことを確認します。

注意

テープまたは他のメディアからアーカイブを削除する場合は、その前にこの手順を実行することをお勧めします。

デバイスのオーナーズ・マニュアルの説明に従って、テープまたは他のメディアを取り外し、安全な場所に保管するか、あるいはサイトのバックアップ・ポリシーと手順に従います。

9.10 スタンドアロン・システム・カーネルのテープ上での作成

テープにブート可能スタンドアロン・システム (SAS) カーネルを作成できます。SAS カーネルには組み込みのメモリ・ファイル・システム (mfs) があ

り、これには、システム・イメージのリストアに必要な最小限のコマンド・ファイル、およびディレクトリが含まれています。これはミニルート・ファイル・システムと呼ばれます。テープには、復元システム上で必要なデータまたはプログラム用のファイル・システムを追加することもできます。

SAS カーネルを作成するには、SysMan Menu の [ブータブル・テープの作成] オプションまたは `btcreate` コマンド行ユーティリティを使用する必要があります。カーネルを作成すると、`btextract` ユーティリティを使用して、カスタマイズしたイメージをリストアすることができます。`btcreate` ユーティリティは、同時に 1 つだけ呼び出すことができます。`/usr/run/bttape.pid` ロック・ファイルにより、このユーティリティが同時に複数起動されるのを防ぎます。

以降の各項では、`bttape` インタフェースの概要、SysMan Menu タスク、`btcreate` および `btextract` コマンド行ユーティリティについて説明します。

9.10.1 テープ・デバイスの必要条件

QIC テープ・ドライブを使用してブート可能テープを作成する場合は、320M バイト以上の高密度テープ以外は使用しないでください。固定 512 ブロックの QIC-24、QIC-120、および QIC-150 フォーマットのテープは使用できません。ブート可能テープ用には、QIC-320 および QIC-525 などの可変ブロック・サイズのテープを使用してください。適切に構成されていない QIC テープ・ドライブを使用して、ブート可能テープを作成すると、I/O エラー、書き込みエラー、またはアクセス許可違反エラーが発生します。したがって、次のいずれかの処置を必ず行ってください。

- インストレーション時にドライブを構成する。
- インストレーション後にドライブがシステムに接続された場合は、カーネルを再構築する。

`btcreate` コマンドを使って作成された QIC テープは、ブート時に次のようなエラー・メッセージを表示して失敗することがあります。

```
failed to send Read to mka... Be sure that the tape is  
write protected before booting.
```

複数のテープにわたるファイル・システムでブート可能テープを作成している場合、`/sbin/dump` コマンドは、テープを交換する必要がある旨のメッセージを表示します。テープを適切に交換しないと、テープを交換するまで警告メッセージが定期的に繰り返し表示されます。

テープ・デバイスに対する `open` 呼び出しの動作は変更されました。書き込み保護テープをオープンする際に、書き込みモードは使用できなくなりました。そのようなテープをオープンしようとしても、次のメッセージが表示されて失敗します。

```
EACCES (permission denied)
```

テープを読むだけのときに `O_RDWR` 許可でテープ・デバイスをオープンするように、アプリケーションが作成されている場合は、オープンが失敗します。 `O_RDONLY` 許可でデバイスをオープンするようにアプリケーションを変更します。アプリケーションが変更できない場合には、次のコマンドを使用して、`open` 呼び出しの動作を以前の状態にしてください。

```
# sysconfig -r cam_tape open_behaviour=0
```

9.10.2 btcreate ユーティリティの使用

ブート可能 SAS カーネルを UFS または AdvFS ファイル・システムにのみ構築するには、`btcreate` ユーティリティを使用する必要があります。以降の項では、テープに SAS カーネルを作成するために必要な情報の概要について説明します。

`btcreate` コマンドには、非対話式および対話式ユーザ・インタフェースの両方が用意されています。どちらのインタフェースを使用する場合にも、スーパーユーザ (root) の特権が必要です。

9.10.2.1 情報の収集

`btcreate` コマンドを使うための準備として、次の情報が必要です。

- `/usr/sys/conf` ディレクトリにあるカーネル構成ファイルの名前。省略時設定の名前は太文字で表したシステム (HOST) の名前と同じです。
- ミニルート・ファイル・システムを格納するディスク・パーティションの名前 (たとえば、`dsk2e`)。必要なディスク・スペースのサイズは、最低 38000 ブロック (1 ブロック = 512 バイト) です。このディスク・パーティションは、`btcreate` の実行中にはマウント状態にしないでください。
- SAS カーネルおよびファイル・システムを格納するテープ・デバイス名 (たとえば `/dev/tape/tape0_d0`)

- デバイス名，マウント・ポイント，およびテープ・デバイスにバックアップを取る各ファイル・システムのタイプ (UFS または AdvFS)。次の例は，有効な UFS および AdvFS のエントリを示しています。

UFS:

```
/dev/dsk1a /      ufs
/dev/dsk1g /usr   ufs
/dev/vol/rootdg/rootvol /ufs
```

AdvFS:

```
root_domain#root /      advfs
usr_domain#usr   /usr   advfs
```

注意

ファイル・システムのバックアップ用には，スワップ・パーティションを選択しないでください。

LSM ボリュームに格納されている UFS ファイル・システムの場合，ブート可能テープの作成時に `vdump` ユーティリティと `vrestore` ユーティリティが使用されます。

- `addlist_file` ファイル。これは，ミニルート・ファイル・システムに含めるファイルまたはディレクトリの一覧を示します。
- `fslist_file` ファイル。これは，バックアップを取るファイル・システムを指定します。
- `/usr/lib/sabt/sbin/custom_install.sh` スクリプト。これは，リストアしたシステム・イメージをカスタマイズするときに使用します。ミニルート・ファイル・システムで提供されている唯一のシェルが Bourne シェルなので，ファイルは Bourne シェル 言語 (`sh1`) を使って記述します。`btcreate` コマンドは，`custom_install.sh` ファイルをテープにコピーして，それをミニルート・ファイル・システムの `sbin` ディレクトリに格納します。`btextract` コマンドは，終了する前に `custom_install.sh` スクリプトを起動します。

次の追加機能は，ブート可能テープのレイアウトを計画するのに便利です。

- `-d` オプションを使用すると，`btcreate` が一時ファイルを作成するロケーションを指定できます。ロケーションを指定しないと，156000 ブ

ロック (1 ブロック = 512 バイト) のディスク・スペースが `/usr` ファイル・システムに必要なになります。

- カスタム `disklabel` スクリプトを使用してディスクにラベルを付けることができます。このスクリプトは、次の必要条件を満たさなければなりません。

- `/usr/lib/sabt/etc` ディレクトリに置くこと
- 名前は、`custom_disklabel_file` とすること。

カスタム `disklabel` スクリプトが存在しない場合は、`btextract` コマンドは通常の方法でディスクにラベルを付けます。詳細については、`disklabel(8)` を参照してください。

9.10.2.2 SAS カーネルの作成

SAS カーネルを作成するため、`btcreate` コマンドは `/usr/sys/conf/YOUR_SYSTEM_NAME` 構成ファイルを `/usr/sys/conf/YOUR_SYSTEM_NAME.BOOTABLE` にコピーして、次のように変更します。

```
config      vmunix      root    on md
pseudo-device      memd    38000
```

この変更により、38000 ブロックのメモリ・ファイル・システムが構成されます。メモリ・ファイル・システムとミニルート・ファイル・システムが格納されるディスク・パーティションは、同じサイズです。

構成ファイルを変更した後、`btcreate` コマンドは `doconfig` コマンドを実行して、ブート可能カーネルを `/usr/sys/bin` ディレクトリに移動します。コマンド構文およびオプションについては、`btcreate(8)` を参照してください。

9.10.3 btextract ユーティリティの使用

`btextract` コマンドは、`btextract` ユーティリティを使用して作成された SAS カーネルが入っているテープから、ファイル・システムをリストアするシェル・スクリプトです。ユーザは、システムの省略時設定のリストアまたはアドバンスト・リストアのいずれかを実行できます。

省略時設定のリストアの際に、カスタマイズされたシステムを同一のハードウェア・プラットフォーム・タイプの複数システム上に複製するように選択することができます。リストア操作に使用するディスク・パーティションを

指定することはできません。代わりに、btcreate セッションの間に収集したディスク・パーティション情報を使用して、btextract コマンドがファイル・システムをリストアします。このとき既存のパーティション情報はすべて上書きされます。アドバンスド・リストアを実行する場合は、使用するディスク・パーティションをオプションで指定できます。ただし、カスタマイズしたシステムは、同一ハードウェア・プラットフォーム・タイプのシステム上でのみ複製できます。

btextract コマンドを使用するには、システムを停止状態にして、システムを初期化した後、次のようにテープからブートします。

```
>>> init
>>> show dev
>>> boot -fl "nc" MKA500
```

この例では、show dev コマンドが BOOTDEV のデバイス名を表示します。MKA500 が BOOTDEV です。

初期ブートが完了すると、シェルはbtextract ユーティリティを起動します。btcreate セッション中に /usr/lib/sabt/sbin/custom_install.sh スクリプトを作成した場合には、btextract コマンドは、終了する前に custom_install.sh スクリプトを呼び出します。custom_prerestore 応答ファイルを作成して回復手順を自動化することもできます。詳細については、btcreate(8) を参照してください。

btextract コマンドが作業を完了した後は、シャットダウンして、次のように、リストアしたディスクからシステムをリブートしなければなりません。

```
# shutdown -h now
>>> boot DKA100
```

この例では、DKA100 が BOOTDEV です。

詳細および例については、btextract(8) を参照してください。

9.10.4 SysMan Menu の [ブータブル・テープの作成] オプションの使用

ブート可能テープの基本的な作成手順を以下に示します。ここでは、9.10.2.1 項で説明されている、必要なデバイス・データの収集が完了し、格納用のテープが用意できていることを前提にしています。

1. SysMan Menu から [ブータブル・テープの作成] タスクを起動するか、プロンプトで次のコマンドを入力します。

```
# sysman boot_tape
```

2. 「ブータブル・テープの作成: *hostname*」ウィンドウが表示されます。次のようにして、フィールドへの入力またはオプションの選択を行ないます。
 - 「カーネル名」フィールドには、ホストの省略時設定のカーネル名が表示されます。これは通常、ローカル・ホストの名前と同じですが、保存されるカーネルに対して、任意の名前を入力できます。
 - 「Miniroot ファイル・システム」フィールドには次のオプションがあります。
 - ミニルートをメモリ・ファイル・システム (mfs) または [ディスク・パーティション] として作成するオプション。使用したいオプションを選択します。
 - dsk0b などのディスク・パーティション名を指定するオプション。[ディスク・パーティション/mfs の指定...] で指定します。これにより、ディスク・パーティション名を入力するダイアログ・ボックスがオープンします。
 - 「テープ・デバイス」フィールドには、省略時設定のテープ・デバイス名 (通常 *tape0_d1*) が表示されます。これは SAS カーネルが格納されるデバイスの名前ですが、他のサポートされているデバイスを指定することもできます。
 - 「Miniroot ファイル・システムのカスタマイズ」フィールドには、*addlist* ファイルの省略時設定のファイル・ロケーションが表示されます。これは、コマンドやユーティリティなどの追加ファイルのリストが入っているデータ・ファイルです。mfs のデータ容量は、360K バイト未満でなければなりません。このリストは、省略時では */usr/lib/sabt/etc/addlist* ファイルに格納されていますが、別のロケーションを指定することもできます。

新しく追加ファイルを作成する、あるいは既存の追加ファイルを変更するには、以下の手順に従ってください。

 - a. [Miniroot Append File の作成または修正] を選択して「作成/修正」ウィンドウをオープンします。
 - b. [追加] を選択して、「追加/修正」ウィンドウをオープンします。ローカル・ホストに置きたいファイルのロケーションを

指定します。たとえば、kill コマンドを追加するには、
/sbin/kill と入力します。次にそのファイルを置くミニルー
ト・ファイル・システム上のロケーションを、/sbin のよ
うに指定します。[了解] を選択して「作成/修正」ウィンド
ウに戻ります。

- c. 「ファイルの内容:」ボックスには、追加されるファイルがリ
ストされます。[了解] を選択して「ブータブル・テープの作
成」のメイン・ウィンドウに戻ります。

- 「ファイル・システムの選択」オプションでは、/usr のような
ファイル・システム、または root_domain#root のような AdvFS
ドメインのバックアップが可能です。バックアップを取るファイル
のリストは、/usr/lib/sabt/etc/fslist に格納されています
が、別の名前を指定することもできます。以下のようにしてファイル
・システムを追加します。

- a. [ファイル・システム・バックアップ・ファイルの作成/修正...]
を選択して、「作成/修正」ウィンドウをオープンします。
- b. [追加] を選択して「追加/修正」ウィンドウをオープンします。
ローカル・ホストにマウントされているディスク・パーティ
ションを、/dev/disk/dsk0g のように指定します。次にマウ
ント・ポイントを /usr のように指定します。[了解] を選択し
て「作成/修正」ウィンドウに戻ります。
- c. バックアップを取るファイル・システムが、「ファイルの内
容:」ボックスにリストされます。[了解] を選択して「ブータブ
ル・テープの作成」のメイン・ウィンドウに戻ります。

- 3. 必要なフィールドへの入力完了すると、テープ作成の準備完了です。
「ブータブル・テープの作成」のメイン・ウィンドウで [了解] を選択し
て先に進みます。作業の開始を示すメッセージ・ウィンドウがオープン
します。テープの作成には、使用するデバイスの速度によって 20 分ま
たはそれ以上かかることがあります。

作業が完了できない場合には、メッセージが表示され、エラー・ログが
/var/adm/btcreate.log にあることを知らせます。

4. テープに正常に書き込まれると、正常終了とログ・ファイル
`/var/adm/btcreate.log` のローケーションを確認するメッセージが
表示されます。

`btextract(8)` を印刷して、後で参照できるようにテープと一緒に保管
してください。

5. ブート可能 SAS カーネルをリストアする際は、9.10.3 項 の手順および
`btextract(8)` を参照してください。意図したとおりに以後の回復が行
われるように、テスト回復の実行を検討してください。



システム課金サービスの管理

この章では、システム課金サービスを設定および使用方法を説明します。この章の内容は、次のとおりです。

- システム課金機能の概要、記録される対象、使用するコマンドとスクリプト、およびシステムのファイルとログ (10.1 節)
- システム課金機能の設定方法 (10.2 節)
- 課金機能の開始と停止の方法 (10.3 節)
- システムへの接続を記録する方法、ログ・ファイル、および関連するコマンド (10.4 節)
- システム上で実行中のユーザ・ジョブを記録する方法、ログ・ファイル、および関連するコマンド (10.5 節)
- ディスク・ストレージの使用量を記録する方法と、そのデータを取得するための関連コマンド (10.6 節)
- システム管理サービスを記録する方法と、そのデータを取得するための関連コマンド (10.7 節)
- プリンタ・サービスの使用量を記録する方法と、そのデータを取得するための関連コマンド (10.8 節)
- すべての課金機能の動作で利用可能なレポート機能と、その関連コマンド (10.9 節)

10.1 課金機能の概要

システム課金サービスは、課金データベースを操作し、システム・リソースの使用量およびユーザ・アクティビティの診断ヒストリの情報を取得して、レポート・ファイルを作成するために使用するシェル・スクリプトおよびコマンドです。

情報を定期的に自動収集するように、課金機能を設定することができます。
また、課金シェル・スクリプトと課金コマンドを手動で実行して、必要に応じて課金情報を得ることもできます。

課金サービスで取得できる課金情報は次のとおりです。

- 接続時間の合計
- CPU 使用時間の合計
- 生成されたプロセスの数
- 接続セッションの数
- メモリ使用量の合計
- I/O 動作回数と転送された文字数
- ブロック内のディスク・スペース使用量
- モデム使用時間および電話接続時間の合計
- 各ユーザまたは各プリンタごとのプリンタ使用量 (プリント動作回数および印刷物の量)

課金機能が使用可能になると、カーネルおよび他のシステム・プロセスは、すべての課金情報のソースである課金情報データベース・ファイルに、レコードを書き込みます。

課金情報データベース・ファイルは `/var/adm` ディレクトリにあり、次のファイルが含まれています。

| ファイル | 説明 |
|--------------------|----------------------|
| <code>wtmp</code> | ログイン/ログアウト・ヒストリ・ファイル |
| <code>utmp</code> | アクティブ接続セッション・ファイル |
| <code>pacct</code> | アクティブ・プロセス課金ファイル |
| <code>dtmp</code> | ディスク使用量ファイル |

課金スクリプトおよび課金コマンドは、課金情報データベース・ファイル内のレコードにアクセスし、保管、診断解析、リソース使用料請求などの目的にレコードを使用できるように、再フォーマットします。

いくつかの課金シェル・スクリプトおよび課金コマンドを使用して、次の処理を行うことができます。

- データベース・ファイル・レコードをフォーマットする。
- データベース・ファイル・レコードから新しいソース・ファイルを作成する。
- データベース・ファイル・レコードを表示する。
- 複数のファイルのデータを 1 つのフォーマットされたファイルにマージする。
- レポートを作成するためにファイル内のデータを要約する。

スクリプトおよびコマンド出力を、ファイル、または他のスクリプトおよびコマンドにリダイレクトしたり、引き渡すことができます。

システム課金機能では、就業時間と非就業時間を区別することができます。非就業時間中のシステム使用率は、就業時間中のシステム使用率よりも低いと考えることができます。非就業時間の指定は、データベース・ファイル `/usr/sbin/acct/holidays` に行います。通常、自動課金機能を使用可能にすると、課金機能は非就業時間中に実行されます。

`startup` シェル・スクリプトによって (課金機能がオンの場合) あるいは、通常、毎日実行される `runacct` スクリプトによって `/var/adm/pacct` ファイルが作成されると課金情報の取得が開始されます。

コマンド出力では、日付情報と時刻情報の順序が各利用者によって異なります。日付指定と時刻指定の順序は、環境変数 `NLTIME` の設定によって変更できます。

10.1.1 課金シェル・スクリプトと課金コマンド

Tru64 UNIX オペレーティング・システムでは、14 の課金シェル・スクリプトと 20 の課金コマンドが用意されています。いくつかのシェル・スクリプトは其中で、課金コマンドや他のシェル・スクリプトを実行します。課金コマンドと課金シェル・スクリプトは、レコードを作成し、課金情報データベース・ファイルに書き込みます。表 10-1 に、提供される課金コマンドおよび課金シェル・スクリプトを示します。

表 10-1: 課金コマンドおよび課金シェル・スクリプト

| 名前 | タイプ | 機能 |
|-----------|-------|---|
| ac | コマンド | 接続セッション・レコードを表示する。 |
| acctcms | コマンド | バイナリ・コマンドの使用法の要約ファイルをフォーマットする。 |
| acctcom | コマンド | 省略時の <code>pacct</code> データベース・ファイルまたは指定されたファイルのプロセス課金レコードの要約を表示する。 |
| acctcon1 | コマンド | <code>wtmp</code> ファイル内のレコードを ASCII フォーマットで要約する。 |
| acctcon2 | コマンド | <code>acctcon1</code> コマンドによってフォーマットされたファイルの内容を要約する。 |
| acctdisk | コマンド | 包括的なディスク使用量課金機能を実行する。 |
| acctdusg | コマンド | ディスク・ブロック使用量の課金機能を実行する。 |
| acctmerg | コマンド | 課金レコード・ファイルをマージする。 |
| accton | コマンド | プロセス課金をオンにする。 |
| acctprc1 | コマンド | ユーザ識別番号とログイン名で、 <code>acct</code> タイプの構造体のレコードを表示する。 |
| acctprc2 | コマンド | ユーザ識別番号とフルネームで、 <code>acct</code> タイプの構造体のレコードを表示する。 |
| acctwtmp | コマンド | レコードを <code>/var/adm/wtmp</code> ファイルに書き込む。 |
| chargefee | スクリプト | レコードをデータベース・ファイル <code>/fee</code> に書き込む。 |
| ckpacct | スクリプト | アクティブ・バイナリ・プロセス課金ファイル <code>/var/adm/acct/pacct</code> のサイズを検査して、ファイルが大きくなりすぎないようにする。 |
| diskusg | コマンド | ユーザ ID 番号に従ってディスク課金機能を実行する。 |
| dodisk | スクリプト | 毎日のディスク使用量課金レコードをディスク使用量課金データベース・ファイル <code>/var/adm/nite/dacct</code> に書き込む。 |
| fwtmp | コマンド | バイナリ・ファイル <code>/var/adm/wtmp</code> のレコードを ASCII フォーマットで表示して、ユーザがエラーを修正できるようにする。 |
| last | コマンド | ログイン情報を表示する。 |
| lastcomm | コマンド | 実行されたコマンドに関する情報を表示する。 |

表 10-1: 課金コマンドおよび課金シェル・スクリプト (続き)

| 名前 | タイプ | 機能 |
|-----------|-------|--|
| lastlogin | スクリプト | すべてのユーザの最終ログイン日付を /var/adm/acct/sum/loginlog ファ イルに書き込む。 |
| monacct | スクリプト | 課金レポート・ファイルの毎月の要約を作成する。 |
| nulladm | スクリプト | adm ユーザおよびグループによって所有され、 保護コードが 664 のファイルを作成する。 |
| pac | コマンド | プリンタ課金レコードを表示する。 |
| prctmp | スクリプト | 接続セッション・レコード・ファイル /var/adm/acct/nite/ctmp を表示する。 |
| prdaily | スクリプト | さまざまなファイルから日次課金レコー ドを収集し、表示する。 |
| printpw | コマンド | /etc/passwd ファイルの内容を表示する。 |
| prtacct | スクリプト | 日次課金ファイル tacct を ASCII 形式で フォーマットして表示する。 |
| remove | スクリプト | ファイル /var/adm/acct/sum/wtmp* , /var/adm/acct/sum/acct/pacct* , および /var/adm/acct/nite/lock* を削除する。 |
| runacct | スクリプト | 日次課金プロセスを実行する。このコマンド は、さまざまな課金コマンドおよびシェル・ス クリプトを定期的に行うして、さまざまな課 金ファイルに情報を書き込みます。 |
| sa | コマンド | 課金レコードの一覧を表示する。 |
| shutacct | スクリプト | 課金機能をオフにする。 |
| startup | スクリプト | 課金機能プロセスを開始する。 |
| turnacct | スクリプト | プロセス課金ファイルの作成を制御する。 |
| wtmpfix | コマンド | /var/adm/wtmp ファイルの日付およびタイ ム・スタンプの矛盾を訂正する。 |

10.1.2 課金ファイル

多数のバイナリ・ファイルおよび ASCII 形式のファイルが、カーネル、ま
たは課金コマンドおよび課金シェル・スクリプトによって、作成および
保守されます。

課金ファイル，特に，バイナリ形式のファイルは大きくなりすぎないように注意する必要があります。課金コマンドおよびシェル・スクリプトによって付随的なファイルがいくつか作成されますが，一般に，これらのファイルは，プロセスが実行されている間だけ存在する一時的なファイルです。たとえばプロセスが中断した場合など，状況によってはこれらの一時ファイルの1つまたは複数が，`/var/adm` サブディレクトリの1つに残る場合があります。これらのサブディレクトリを定期的に検査して，不要なファイルを削除する必要があります。

課金ファイルは，破壊されたり，失われたりすることもあります。課金データベース・ファイル `/var/adm/wtmp` および `/var/adm/acct/sum/tacct` のように，日次レポートまたは月次レポートを作成するのに使用されるファイルは，データの完全性が保証されなければなりません。これらのファイルが破壊されたり失われたりした場合，バックアップから復元することができます。また，`fwtmp` または `wtmpfix` コマンドを使用して，`/var/adm/wtmp` ファイルを修正することもできます。詳細は，10.4.2 項および 10.4.1 項を参照してください。また，`acctmerg` コマンドを使用して `/var/adm/acct/sum/tacct` ファイル内のエラーを修正することができます。詳細は，10.9.2 項を参照してください。

`/var/adm/acct/nite` ディレクトリには，`runacct` スクリプトが毎日繰り返し利用されるファイルが含まれています。これらのファイルの中には，`/var/adm/acct/sum` ディレクトリにバイナリ形式のファイルを持っているものがあります。`/var/adm/acct/sum` ディレクトリには，`runacct` シェル・スクリプトによって更新され，また，月次レポートを作成するために `monacct` シェル・スクリプトによって利用される，累積要約ファイルが含まれています。

表 10-2 に，`/var/adm` ディレクトリ内のデータベース・ファイルを示します。

表 10-2: `/var/adm` ディレクトリ内のデータベース・ファイル

| ファイル名 | タイプ | 説明 |
|-------------------|-------|--|
| <code>dtmp</code> | ASCII | <code>dodisk</code> シェル・スクリプトによって生成された一時的な出力が含まれる。 |
| <code>fee</code> | ASCII | <code>chargefee</code> シェル・スクリプトからの出力が含まれる。 |

表 10-2: /var/adm ディレクトリ内のデータベース・ファイル (続き)

| ファイル名 | タイプ | 説明 |
|--------------|------|---|
| pacct | バイナリ | 有効なプロセス課金データベース・ファイルを指定する。プロセスが、ユーザ、別のプロセス、またはスクリプト・ファイルによって実行されると、このファイルにプロセス処理情報が書き込まれる。 |
| pacctn | バイナリ | turnacct switch コマンドによって作成された代替の pacct ファイルを指定する。システムが多数のユーザを持っていると、pacct データベース・ファイルはすぐに大きくなってしまふ。1 つの pacct ファイルは、最大 500 の 1024 バイト・ブロックからなるディスク・スペースに制限されている。これらのファイルのサイズは、runacct シェル・スクリプトによって監視される。新しい pacctn ファイルが作成されるたびに、値 <i>n</i> が 1 ずつ大きくなる。 |
| qacct | バイナリ | キュー登録 (プリンタ) システム課金レコードが含まれている。このファイルは、runacct シェル・スクリプトによって使用される。 |
| savacct | バイナリ | システム・プロセス課金要約レコードを保存するために sa コマンドが使用するファイルを指定する。 |
| Spacctn.mmdd | バイナリ | <i>mm</i> と <i>dd</i> でそれぞれ指定された月日ごとに runacct シェル・スクリプトが作成する pacctn ファイルを指定する。 |
| usracct | バイナリ | ユーザ・プロセス課金要約レコードを保存するために sa コマンドが使用するファイルを指定する。 |
| utmp | バイナリ | ユーザが接続セッションを生成するプロセスを実行すると書き込まれる、アクティブな接続セッション課金データベース・ファイルを指定する。 |
| wtmp | バイナリ | 累積ログイン/ログアウト課金データベース・ファイルを指定する。ユーザがシステムにログインすると、このファイルに接続時間およびユーザ情報が書き込まれる。 |

表 10-3 に、/var/adm/acct/nite ディレクトリ内のデータベース・ファイルを示します。

表 10-3: /var/adm/acct/nite ディレクトリの日次ファイル

| ファイル名 | タイプ | 説明 |
|-------------|-------|---|
| active | ASCII | 日次 runacct シェル・スクリプト進捗ファイルを指定する。runacct シェル・スクリプトが実行されると、このファイルに、その進捗に関する情報が書き込まれる。このファイルには、エラー・メッセージおよび警告メッセージも書き込まれる。 |
| activemmdd | ASCII | mm と dd でそれぞれ指定される月日の、日次 runacct シェル・スクリプト・エラー・ファイルを指定する。このファイルは、active ファイルと類似している。 |
| cklock | ASCII | ckpacct シェル・スクリプトが使用するファイルを指定し、複数の runacct シェル・スクリプトがある時間から 24 時間実行されないようにする。このファイルは、runacct シェル・スクリプトが終了すると削除される。 |
| cms | ASCII | 使用中の総合日次コマンド要約ファイルを指定する。このファイルは、/var/adm/acct/sum/cms ファイルの ASCII バージョンである。cms ファイルは、runacct シェル・スクリプトによって作成される。runacct は、acctcms コマンドによって実行されて、/var/adm/acct/sum/cms ファイルのレコードの再書き込みを行う。monacct シェル・スクリプトで、このファイルを初期化する。 |
| ctacct.mmdd | バイナリ | mm と dd によってそれぞれ指定されている月日の、接続セッション課金レコードから得られる接続課金レコードを、tacct.h フォーマットで指定する。このファイルは一時ファイルであり、個々の課金期間に daytacct ファイルのレコードが書き込まれた後、削除される。 |
| ctmp | ASCII | 一時ログイン/ログアウト・レコード・ファイルを指定する。このファイルには、wtmp ファイルのレコードの再書き込みを行うために runacct シェル・スクリプトによって実行される課金コマンド acctconl の出力が含まれている。 |
| daycms | ASCII | 日次コマンド要約ファイルを指定する。バイナリ・ファイル /var/adm/acct/sum/daycms の ASCII バージョン。runacct シェル・スクリプトは、acctcms コマンドを実行してファイルを作成する prdaily シェル・スクリプトを実行する。 |
| daytacct | バイナリ | 前日の総合課金レコードが tacct.h 形式で含まれている。 |

表 10-3: /var/adm/acct/nite ディレクトリの日次ファイル (続き)

| ファイル名 | タイプ | 説明 |
|---------------|-------|---|
| dacct | バイナリ | acctdisk コマンドが dodisk シェル・スクリプトによって呼ばれたときに、毎週の総ディスク使用量課金レコードが書き込まれる。 |
| lastdate | ASCII | runacct シェル・スクリプトが実行された最後の日を指定する。 |
| lineuse | ASCII | 端末 (tty) 回線接続時間が書き込まれる。このファイルは、前回の課金期間中に使用された各端末回線の回線使用統計情報を提供する。 |
| lock | ASCII | cron デーモンが 24 時間の間に 2 回以上 runacct シェル・スクリプトを実行しないようにするために使用するファイルを指定する。このファイルは、毎日、runacct シェル・スクリプトが終了すると削除される。 |
| log | ASCII | runacct スクリプトが acctcon1 コマンドを実行すると作成される診断出力が書き込まれる。 |
| owtmp | バイナリ | wtmpfix コマンドによる修正後に、日次 wtmp ファイルを指定する。 |
| ptacctn.mmdd | バイナリ | mm と dd でそれぞれ指定される月日ごとの、追加の日次 pacctn ファイルを指定する。これらのファイルは、日次プロセス課金ファイル pacct が 501 以上のディスク・ブロックを必要とする場合に作成される。 |
| reboots | ASCII | 前の課金期間中のシステム・リブートのリストを内容として持つ。 |
| statefile | バイナリ | 最後の runacct シェル・スクリプトの実行状態を指定する。 |
| wtmp.mmdd | バイナリ | mm と dd でそれぞれ指定された月日の、修正された日次ログイン/ログアウト・データベース・ファイルを指定する。前日中にシステムにログインしたユーザの接続セッション・レコードが、このファイルに書き込まれる。 |
| wtmperror | ASCII | wtmpfix コマンドの実行中に wtmp ファイルが修正された場合に生成されるエラー・メッセージが記録されている。 |
| wtmperrormmdd | ASCII | mm と dd でそれぞれ指定された月日について、wtmpfix コマンドの実行中に runacct シェル・スクリプトがエラーを検出すると生成されるエラー・メッセージが記録されている。 |

表 10-4 は、`/var/adm/acct/sum` ディレクトリ内のデータベース・ファイルを示します。

表 10-4: `/var/adm/acct/sum` ディレクトリ内の要約ファイル

| ファイル名 | タイプ | 説明 |
|------------------------|-------|--|
| <code>cms</code> | バイナリ | 使用中の総合コマンド要約ファイルを指定する。 <code>runacct</code> シェル・スクリプトが実行されると、総合コマンド要約ファイルを得るために、このファイルにレコードが記録されている。 |
| <code>cmsprev</code> | バイナリ | 前日の <code>/var/adm/acct/sum/cms</code> ファイルを指定する。 |
| <code>daycms</code> | バイナリ | 前日のコマンド要約ファイルを指定する。 <code>runacct</code> シェル・スクリプトが実行されると、前日の月次コマンド要約レコードがこのファイルに書き込まれる。 |
| <code>loginlog</code> | ASCII | 個々のユーザ名の、最後の月次のログイン日付のリスト。 |
| <code>rprrtmmdd</code> | ASCII | <code>mm</code> と <code>dd</code> でそれぞれ指定された月日の日次課金レポートを指定する。 |
| <code>tacct</code> | バイナリ | 累積課金ファイルを指定する。このファイルは、システム使用に関する総合日次課金ファイルである。このファイルは、 <code>runacct</code> シェル・スクリプトによって毎日更新される。 |
| <code>tacctmmdd</code> | バイナリ | <code>mm</code> と <code>dd</code> でそれぞれ指定された日付の課金ファイルを指定する。 |
| <code>tacctprev</code> | バイナリ | 前日の <code>tacct</code> ファイルを指定する。このファイルは、前の課金期間のバイナリ・ファイル <code>tacct</code> である。 |

表 10-5 は、`/var/adm/acct/fiscal` ディレクトリ内のデータベース・ファイルを示します。

表 10-5: `/var/adm/acct/fiscal` ディレクトリ内の月次ファイル

| ファイル名 | タイプ | 説明 |
|--------------------|------|---|
| <code>cmsmm</code> | バイナリ | <code>mm</code> で指定された月の、アクティブなコマンド要約ファイルを指定する。 |

表 10-5: /var/adm/acct/fiscal ディレクトリ内の月次ファイル (続き)

| ファイル名 | タイプ | 説明 |
|-----------|-------|---|
| fiscrptmm | ASCII | mm で指定された月の課金レポートを指定する。 |
| tacctmm | バイナリ | 累積総合課金ファイルを指定する。このファイルは、システム使用についての総合課金ファイルである。このファイルは、monacct シェル・スクリプトによって、毎月 1 回更新される。 |

10.2 課金機能の設定

課金機能を使用するには、System Accounting Utilities サブセットをインストールしなければなりません。このサブセットがインストールされているかどうかは、次のコマンドで確認できます。

```
# setld -i | grep count
OSFACCTxxx    installed  System Accounting Utilities \
(System Administration)
```

このサブセットがインストールされていない場合は、setld コマンドを使用して、RIS サーバまたは CD-ROM からインストールします。サブセットをインストールすると、必要な課金サービスを使用可能にすることができます。

多数のユーザが競合してシステム・リソースを使用するようなシステム環境では、UNIX システム課金機能を使用することによってシステムの使用状況を追跡することができます。この場合、追跡しようとする情報のタイプおよび量を決定する必要があります。また、自動課金機能を設定するかどうかにも決定する必要があります。自動課金機能を使用可能にするには、/usr/spool/cron/crontabs ディレクトリ内のファイルに課金機能コマンドおよびシェル・スクリプトを指定します。

ネットワークを形成しているすべてのマシンの課金情報を得る場合は、単一のマシンに課金機能を設定しておきます。システム課金機能を使用可能にするには、次の手順を実行します。以降の項で、これらの手順について詳しく説明します。

1. /etc/rc.config ファイルに課金情報自動取得の設定を行います (10.2.1 項)。
2. qacct ファイル、pacct ファイル、fee ファイルを確認します (10.2.2 項)。

3. `/usr/sbin/acct/holidays` ファイルを編集して、就業時間、非就業時間、および休日を指定します (10.2.3 項)。
4. 自動課金機能を起動するために、`/usr/spool/cron/crontabs` ディレクトリ内のファイルを変更して、課金シェル・スクリプトおよびコマンドを実行します (10.2.4 項)。

プリント・ドライバ・ソフトウェアは、別のサーバ、デーモン、およびルーチンを使用しているため、プリンタ課金情報は、リソース課金情報とは別に処理されます。プリンタ課金機能を設定する方法については、第 8 章を参照してください。この章では、自動プリンタ課金機能を使用可能にする方法のみを説明します。

10.2.1 rc.config ファイルの編集

課金機能を設定するには、次の 1 行を `/etc/rc.config` ファイルに追加する必要があります。

```
ACCOUNTING="YES"
```

次のように、`rcmgr` コマンドを使用して変数を設定できます。

```
# rcmgr set ACCOUNTING YES
```

`startup` コマンドを使用すれば、システムをリブートしなくても課金機能を開始することができます。詳細は、10.3 節を参照してください。

10.2.2 qacct ファイル、pacct ファイル、および fee ファイルの確認

課金が機能するためには、システム上にキュー登録課金ファイル `qacct` およびプロセス課金データベース・ファイル `pacct` が存在しなければなりません。これらのファイルは、コンテキスト依存のシンボリック・リンク (CDSL) のパス名で、ブランク・ファイルとしてプリインストールされています。 `ls -l` ディレクトリ表示コマンドを使用すると、リンクは次のパスとして表示されます。

```
/usr/var/cluster/members/member0/adm/acct/fee
/usr/var/cluster/members/member0/adm/acct/pacct
/usr/var/cluster/members/member0/adm/acct/qacct
```

オリジナルのファイルがない (または誤って壊してしまった) 場合、これらのファイルを CDSL として再作成しなければなりません。CDSL の再作成については、第 6 章、`cdslinvchk(8)` および `mkcdsl(8)` を参照してください。代

わりに、取っておきたい課金データや構成ファイルを保存してから、課金ソフトウェア・サブセットを再インストールする方法もあります。

これらのファイルは `adm` ユーザおよびグループが所有し、パーミッションが `644` でなければなりません。必要であれば、`chown` コマンドおよび `chgrp` コマンドを使用して、これらの値を設定しなおしてください。

10.2.3 holidays ファイルの編集

`/usr/sbin/acct/holidays` ファイルは、24 時間単位で、就業時間および非就業時間を指定します。また、休日を指定することもできます。休日は非就業時間に含まれます。就業時間に含まれるのは、月曜日から金曜日までです。非就業時間中のシステム使用率は、就業時間中のシステム使用率より低いと考えられます。自動課金機能を使用する場合は、非就業時間中にコマンドが実行されるように設定します。

`/usr/sbin/acct/holidays` ファイルが存在しない場合は作成する必要があります。また、このファイルがすでに存在する場合は、取得しようとする課金情報に合わせて編集する必要があります。

`NHOLIDAYS` 環境変数を指定することによって、休日の上限数を `holidays` ファイルの中に設定できます。

10.2.4 crontab ファイルの変更

自動課金機能を使用可能にするには、`crontab` コマンドを使用して、`/usr/spool/cron/crontabs` ディレクトリ内のファイルを変更する必要があります。`/usr/spool/cron/crontabs` ディレクトリ内のファイルには、`cron` デーモンが特定の特権の下に指定された時間に実行するコマンドが含まれています。たとえば、`/usr/spool/cron/crontabs/root` ファイル内のコマンドは、`root` の特権で実行され、`/usr/spool/cron/crontabs/adm` ファイル内のコマンドは、`adm` の権限で実行されます。

次のコマンドとシェル・スクリプトを `/usr/spool/cron/crontabs/adm` ファイルに記述することができます。

```
ckpacct          このシェル・スクリプトは、プロセス課金データ
                  ベース・ファイル pacct のサイズを検査し、大き
                  くなりすぎないようにします。
```

runacct

このシェル・スクリプトは、他の課金シェル・スクリプトおよびコマンドを含んでおり、日次課金ファイルと月次課金ファイルを作成します。runacct シェル・スクリプトを変更して、不必要な課金のコマンドを削除することもできます。

monacct

このシェル・スクリプトは、月次要約課金ファイルを作成します。monacct シェル・スクリプトを変更して、不必要な課金コマンドを削除することもできます。

ac

このコマンドは、接続時間レコードを表示します。出力先としてファイルを指定することもできます。このコマンドを runacct シェル・スクリプトに追加することもできます。

pac

このコマンドは、プリンタ課金レコードを表示します。出力先としてファイルを指定することもできます。プリンタ課金機能を使用可能にする方法についての詳細は、10.8 節を参照してください。

dodisk シェル・スクリプトは、/usr/spool/cron/crontabs/root ファイルに記述することができます。dodisk シェル・スクリプトは、ディスク使用量課金レコードを作成します。このシェル・スクリプトは、非就業時間中に毎週 1 回実行させます。

crontab デーモンに対するコマンドの発行についての詳細は、第 3 章および crontab(1) を参照してください。

次の例は、課金機能コマンドおよびシェル・スクリプトが記述されている /usr/spool/cron/crontabs/adm ファイルの一部を示しています。

```
0 2 * * 1-6 /usr/sbin/acct/runacct > /usr/adm/acct/nite/fd2log&
5 * * * * /usr/sbin/acct/ckpacct&
0 4 1 * * /usr/sbin/acct/monacct&
10 3 * * * /usr/sbin/ac -p > /var/adm/timelog&
40 2 * * * /usr/sbin/pac -s&
```

次の例は、dodisk シェル・スクリプトが記述されている /usr/spool/cron/crontabs/root ファイルの一部を示しています。

```
0 3 * * 4 /usr/sbin/acct/dodisk > /var/adm/diskdiag&
```

10.3 課金機能のスタートアップと停止

シェル・スクリプト `startup` および `shutacct` は、さまざまな課金機能プロセスを使用可能にあるいは使用不能に設定します。この2つのスクリプトは、ログイン名としてシステム名を使用し、`/var/adm/wtmp` ファイルにレコードを追加する `acctwtmp` プログラムを実行します。

`startup` シェル・スクリプトは、課金機能を初期化します。構文は次のとおりです。

`/usr/sbin/acct/startup`

注意

`startup` スクリプトによって作成される `pacct` ファイルは、グループ `adm` とユーザ `adm` によって所有され、保護コード 664 でなければなりません。所有権が正しくない場合、`accton` コマンドは動作せず、次のメッセージが表示されます。

```
accton: uid/gid not adm
```

`shutacct` スクリプトは、プロセス課金を無効にし、必ず課金機能が停止してから、システムがシャットダウンするようにします。`shutacct` シェル・スクリプトの構文は次のとおりです。

`/usr/sbin/acct/shutacct` [*Reason*]

Reason 文字列は、ユーザが定義した、コマンド起動の理由です。`shutacct` シェル・スクリプトが実行されると、`/var/adm/wtmp` ファイル内のシャットダウン・レコードの `ut_line` フィールドに *Reason* メッセージが書き込まれます。次に、`turnacct off` シェル・スクリプトが実行されて、実行中の課金機能を使用不能にしなければならないことをカーネルに通知します。

10.4 接続セッション課金機能

ユーザがログインまたはログアウトすると、`login` コマンドおよび `init` コマンドがバイナリ・データベース・ファイル `/var/adm/wtmp` にユーザ・

ログイン・ヒストリおよびユーザ・ログアウト・ヒストリのレコードを書き込みます。バイナリ・データベース・ファイル `/var/adm/utmp` は、使用中の接続セッション・ファイルです。ハングアップ、`login` コマンドの終了、ログイン・シェルの終了が発生すると、システムはログアウト・レコードをこのファイルに書き込みます。したがって、ログアウト数がセッション数よりも多くなることがよくあります。

注意

課金レコードが、オペレーティング・システムのバージョン 4.0A のときに作られたものである場合は、`wtmpconvert(8)` で、ファイルの変換方法を調べてください。

接続セッション・コマンドは、`/var/adm/wtmp` ファイルのレコードを有効な接続セッション課金レコードに変換できます。接続セッション課金情報を得られるのは、`/var/adm/wtmp` ファイルが存在する場合だけです。

`/var/adm/wtmp` ファイルのフォーマット付きレコードには、各接続セッションに関して下記の情報が記述されています。

- ユーザ・ログイン名 (`/etc/passwd` ファイルから)
- 回線識別番号 (`/etc/inittab` ファイルから)
- デバイス名 (コンソールあるいは `tty23` など)
- エントリのタイプ
- プロセス ID 番号
- プロセス終了状態
- プロセス `exit` 状態
- エントリが行われた時間
- ホスト・マシン名

システム接続セッションに関する情報の取得や変更には、次の 2 つのシェル・スクリプトおよび 7 つのコマンドが使用されます。

| コマンド | 機能 |
|-----------|--|
| ac | システム全体および個々のユーザの接続セッション・レコードを表示する。 |
| acctcon1 | 接続セッション・レコードを要約して、それらのレコードをASCII形式(各接続セッションに1行を使用する)で表示する。 |
| acctcon2 | acctcon1 コマンドの出力を使用して、全接続セッションの課金レコード・ファイルをASCII形式で作成する。 |
| acctwtmp | キーボードからレコードを入力することによって、レコードをwtmpファイルに書き込めるようにする。 |
| fwtmp | ファイル構造 utmp.h を持つファイルのレコードを表示する。 |
| last | ログイン情報を表示する。 |
| lastlogin | 個々のユーザが最後にログインした日付を示すために、 /var/adm/acct/sum/loginlog ファイルを更新する。 |
| prctmp | acctcon1 コマンドが作成したセッション・レコード・ファイル (通常は /var/adm/acct/nite/ctmp) の内容を表示する。 |
| wtmpfix | 日付変更によって影響を受ける接続セッション・レコード wtmp を修正し、wtmp ファイル内のログイン名フィールド に書き込まれたログイン名を有効にする。 |

ヘッダ・ファイル /usr/include/utmp.h の構造は、次の接続セッション・ファイル用のレコード形式です。

- /var/adm/wtmp
- /var/adm/utmp
- /var/adm/acct/nite/wtmp.mmd
- /var/adm/acct/nite/ctmp

ヘッダ・ファイル /usr/include/utmp.h は、9つのフィールドから構成されています。表 10-6 は、utmp をヘッダ・ファイルのフィールド番号、メンバー名説明および必要に応じて、文字の長さをASCII形式で示したものです。

表 10-6: utmp ASCII 形式によるファイル構成要素

| フィールド | 要素 | 説明 |
|-------|-----------------------|--|
| 1 | ut_user | ユーザ・ログイン名。sizeof(ut_user) 文字が入っていないといけない。 |
| 2 | ut_id | inittab ID。sizeof(ut_id) 文字が入っていないといけない。 |
| 3 | ut_line | レコードのタイプ (たとえばデバイス名) を記述するために使用される情報を保持しているメモリ位置。sizeof(ut_line) 文字が入っていないといけない。 |
| 4 | ut_pid | プロセス ID 番号。 |
| 5 | ut_type | エントリのタイプ。いくつかのシンボリック定数値を指定できる。シンボリック定数は、ヘッダ・ファイル /usr/include/utmp.h に定義される。 |
| 6 | ut_exit.e_termination | プロセス終了状態。 |
| 7 | ut_exit.e_exit | プロセス exit 状態。 |
| 8 | ut_time | 開始時間 (秒単位)。 |
| 9 | ut_host | ホスト名。sizeof(ut_host) 文字が入っていないといけない。 |

10.4.1 wtmpfix コマンド

/usr/sbin/acct/wtmpfix コマンドは、utmp.h ヘッダ・ファイル構造体を持つファイルにおける日付とタイム・スタンプの不一致を修正し、そのレコードを表示します。runacct スクリプトが wtmpfix コマンドを実行します。

システムの起動時、あるいは date コマンドが実行された場合など、/var/adm/wtmp ファイルに日付が入力されるたびに、1 組の日付変更レコードが wtmp ファイルに書き込まれます。最初の日付変更レコードは、フィールド ut_line および ut_type に指定されている古い日付です。2 番目の日付変更レコードは、ut_line フィールドおよび ut_type フィールドに指定されている新しい日付です。wtmpfix コマンドは、これらのレコードを使用して /var/adm/wtmp ファイルにおけるすべての日付および日時のスタンプを同期化し、これらの 1 組のレコードを削除します。日付変更レコードは、出力ファイルには表示されません。

wtmpfix コマンドは、ユーザ名が英数字、\$ (ドル記号)、スペースのみで構成されているかなど、ユーザ名フィールド (ut_user) の有効性も検査します。無効なユーザ名が検出されると、wtmpfix コマンドはログイン名を INVALID に変更して、診断メッセージを表示します。

wtmpfix コマンドの構文は次のとおりです。

/usr/sbin/acct/wtmpfix [*filename*]...

filename 変数には、入力ファイル名を指定します。省略時の入力ファイルはバイナリ・ファイル /var/adm/wtmp です。

10.4.2 fwtmp コマンド

fwtmp コマンドを使用することにより、wtmp ファイルを修正することができます。このコマンドは、utmp.h ヘッダ・ファイル構造体を持つファイルのバイナリ・レコードをフォーマット付き ASCII レコードに変換します。ASCII バージョンの wtmp ファイルを編集することにより、不良レコードの修復や、通常のファイル保守を行うことができます。表 10-6 に使用すべき ASCII 構造体を示しています。

システム動作中に日付変更およびリブートが起こると、/var/adm/wtmp ファイルにレコードが書き込まれます。wtmpfix コマンドは、/var/adm/wtmp ファイルのタイム・スタンプを調整します。しかし、修正内容によっては wtmpfix コマンドによる検出が不可能になり、acctcon コマンドが異常終了することがあります。その場合、fwtmp コマンドを使用して、/var/adm/wtmp ファイルを修正します。

fwtmp コマンドの構文は次のとおりです。

/usr/sbin/acct/fwtmp [-ic]

fwtmp ファイルは標準入力を使用しますが、コマンドの出力としてファイルを指定することもできます。

fwtmp コマンドでオプションを指定しないと、バイナリ・レコードは ASCII レコードに変換されます。このコマンドのオプションについての詳細は fwtmp(8) を参照してください。

ヘッダ・ファイル /usr/include/utmp.h のレコードを手動で入力する場合は、表 10-6 に示されているように、utmp ASCII 構造体メンバが使用する順に、9 つのフィールドのそれぞれにデータを入力しなければなりま

せん。キーボードから入力するすべてのレコード・フィールド・エントリは、スペースで区切ります。また、ブランク文字を使用してすべての文字列フィールドを (必要ならば、最大の文字列サイズまで) 指定しなければなりません。すべての小数值は、空の桁にはゼロ (0) を指定し、必要な数の小数点以下の桁数を持たなければなりません。

次の例は、バイナリ・ファイル `/var/adm/wtmp` のレコードを ASCII 形式のレコードに変換します。

```
# /usr/sbin/acct/fwtmp < /var/adm/wtmp

          system boot  0 20000 0000 652547412 Jan 5 11:10:12 1994
          system boot  0 10062 0123 652547412 Jan 5 11:10:12 1994
bcheck   bl           6 80000 0000 652547413 Jan 5 11:10:13 1994
cat      cr          16 80000 0000 652547414 Jan 5 11:10:14 1994
rc       rc          17 80000 0000 652547485 Jan 5 11:11:25 1994
hoffman  co console  147 70000 0001 652547495 Jan 5 11:11:35 1994
hoffman  p4 pty/ttyp4 2156 80000 0002 652650095 Jan 6 15:41:35 1994
LOGIN    p4 pty/ttyp4 2140 60000 0000 652649075 Jan 6 15:24:35 1994
LOGIN    p4 pty/ttyp4 2140 80000 0000 652649086
Jan 6 15:24:46 1994
```

`/var/adm/wtmp` ファイルを修正する手順は次のとおりです。

1. 作業ディレクトリを `/var/adm/acct/nite` に変更します。
2. `fwtmp` コマンドを使用して、`wtmp` ファイルの ASCII バージョンを作成します。

```
# fwtmp < wtmp.0617 > wtmp_temp
```

3. 一時ファイルを編集し、不正なレコードを削除します。
4. `fwtmp` コマンドを使用して `wtmp` ファイルを作成し直します。

```
# fwtmp -ic < wtmp_temp > wtmp.0617
```

10.4.3 acctwtmp コマンド

`acctwtmp` コマンドを使用することにより、コマンドを起動した理由を示す文字列、現在の時刻、および日付を `utmp.h` 構造体形式のファイル (通常は `/var/adm/wtmp` ファイル) に追加することができます。 `runacct`、`startup`、および `shutacct` シェル・スクリプトが `acctwtmp` コマンドを実行し、`runacct` スクリプトの実行時期およびシステム課金機能を起動または停止した時期を記録します。

`acctwtmp` コマンドの構文は次のとおりです。

/usr/sbin/acct/acctwtmp *reason*

reason 変数は最大の `sizeof(ut_line)` 文字を持ち, " " (引用符) で囲まなければならない。

10.4.4 ac コマンド

ac コマンドは, 表 10-6 に記載されている utmp ファイル構造体を持つファイルの接続セッション・レコードを表示します。このコマンドを使用して, システム診断や, ユーザへの請求金額の決定ができます。ac コマンドは, すべてのユーザの接続時間の合計, または指定したユーザの接続時間の合計を表示します。接続時間は, 最も近似の小数点以下第 2 位に丸めた時間で表示されます。

総ユーザ接続セッション・ファイルを自動的に作成するには, ac コマンドを `/usr/spool/cron/crontab/adm` ファイルに記述するか, または `runacct` シェル・スクリプトを変更して ac コマンドが含まれるようにします。自動課金機能の設定についての詳細は, 10.2.4 項を参照してください。

ac コマンドの構文は次のとおりです。

/usr/sbin/ac [-d] [-p] [-w *filename*] [*username*...]

このコマンドのオプションの詳細は, `ac(8)` を参照してください。

オプションを省略した場合は, すべてのユーザのシステム接続時間の合計を表示します。次に例を示します。

```
# /usr/sbin/ac
"total 48804.26"
```

次のコマンドは, ユーザ名ごとの合計接続時間を表示します。

```
# /usr/sbin/ac -p
buckler      61.44
fujimori     530.94
newsnug      122.38
dara          0.10
root         185.98
buchman      339.33
russell       53.96
hoff         200.43
hermi        157.81
total        1968.02
```

最後の行には、リストされているすべてのユーザの接続時間の合計が表示されます。

10.4.5 acctcon1 コマンド

acctcon1 コマンドは、ヘッダ構造体 utmp.h を持つファイルのバイナリ・セッション・レコードを ASCII 形式に変換します。個々の接続セッションに対して、1つのレコードが作成されます。runacct シェル・スクリプトは、acctcon1 コマンドを使用して、lineuse および reboots ファイルを作成します。これらのファイルは日次レポート /var/adm/acct/sum/rprtmmda に含まれます。

acctcon1 コマンドの構文は次のとおりです。

/usr/sbin/acct/acctcon1 [-l *file*] [-o *file*] [-pt]

コマンドへの入力としてファイルを指定する必要があります。このコマンドのオプションの詳細は、acctcon1(8) を参照してください。

次に示すのは、/var/adm/acct/nite/lineuse ファイルの例です。このコマンドは、指定されたファイルに ASCII 回線使用量形式でレコードを書き込みます。この形式は、回線使用量を追跡し、不良回線を識別し、ユーザがログインしたポートの参照指定、および、現在実行中の接続セッションの日付およびタイプ・スタンプをいれるのに役立ちます。

```
# acctcon1 -l line_file < /var/adm/wtmp | more line_file
TOTAL DURATION IS 57 MINUTES
LINE           MINUTES      PERCENT    # SESS    # ON     # OFF
pty/ttyp4       37          64         3         3         7
console         26          45         2         2         4
pty/ttyp5        7          11         1         1         3
pty/ttyp6        0           0         0         0         2
TOTALS          69          -         6         6        16
```

上の例で、ASCII 回線使用量形式は、次のことを指定します。

- システムがマルチユーザ状態にあった合計時間 (分)
- 回線名
- 課金期間中に使用されたセッション時間 (分)
- 合計期間に対する使用中になっていた時間 (分) の比率
- ポートがアクセスされた回数 (4 番目と 5 番目の欄)
- 回線上でのログアウトおよびその他の割り込みの回数

最後の欄を 4 番目の欄と比較して、回線が不良であるかどうかを判定することができます。

次は、`/var/adm/acct/reboots` ファイルの例です。ASCII オーバオール・レコード形式でファイルにレコードを書き込みます。ASCII オーバオール・レコード形式では、開始時間、終了時間、再起動の回数、日付変更の回数が指定されます。

```
# acctcon1 -o overall_file < /var/adm/wtmp | more overall_file
from Thu Jun 13 17:20:12 2002 EDT
to   Fri Jun 14 09:56:42 2002 EDT
2   date changes
2   acctg off
0   run-level S
2   system boot
2   acctg on
1   acctcon1
```

オーバーオール・レコード形式には、`from` および `to` フィールドが含まれています。このフィールドには、最後の課金レポートが作成された時間および現在のレポートの時間が指定されます。これらのフィールドの後には、`/var/adm/wtmp` ファイルのレコードのリストが続きます。

10.4.6 acctcon2 コマンド

`runacct` シェル・スクリプトは `acctcon2` コマンドを起動し、`acctcon1` コマンドによって作成される接続セッション・ファイル `/var/adm/acct/nite/ctmp` を ASCII 形式からバイナリ形式に変換します。

10.4.7 prctmp シェル・スクリプト

`prctmp` シェル・スクリプトは、`acctcon1` コマンドによって作成される `/var/adm/acct/nite/ctmp` ファイルなどの、`utmp.h` ヘッダ・ファイル構造体を持っている接続セッション・データベース・ファイルに欄見出しを書き込みます。`prctmp` シェル・スクリプトの構文は次のとおりです。

`/usr/sbin/acct/prctmp` [*filename*]

詳細は、`prctmp(8)` を参照してください。

10.4.8 lastlogin シェル・スクリプト

lastlogin シェル・スクリプトは、ユーザがシステムに最後にログインした日付を /var/adm/acct/sum/loginlog ファイルに書き込みます。このスクリプトは、printpw コマンドを実行して、/etc/passwd ファイルのログイン名とユーザ識別番号にアクセスします。

CMS 状態において runacct シェル・スクリプトが lastlogin シェル・スクリプトを起動します。lastlogin シェル・スクリプトを手動で実行して、日次レポート /var/adm/acct/sum/rprtmdd に含まれている /var/adm/acct/sum/loginlog ファイルを更新することもできます。

lastlogin シェル・スクリプトの構文は次のとおりです。

/usr/sbin/acct/lastlogin

10.4.9 last コマンド

last コマンドは、/var/adm/wtmp ファイル内のすべてのログイン・レコードを入力と逆の順序で表示します。各ログイン・セッションについて、次の情報が示されます。

- セッションが開始された時刻
- セッションの期間
- セッションが発生した端末

状況によっては次の情報も含まれます。

- リブート時の終了
- 連続セッション

last コマンドの構文は次のとおりです。

/usr/bin/last [-#] [*username...*] [*tty...*]

省略時の設定ではすべてのレコードが表示されます。ユーザ名あるいは端末を指定することもできます。

最近の 3 つの root ログインに関する情報を表示させる場合は、次のように入力します。

```
# last -3 root
root    ttypl    shout    Fri Jun 21 10:56    still logged in
root    ttypl    raven    Fri Jun 21 08:59 - 09:00    (00:00)
```

10.5 プロセス課金

コマンド、シェル・スクリプト、またはプログラムがシステムで実行されると、プロセス課金が発生します。プロセスが終了すると、カーネルはプロセス課金レコードをデータベース・ファイル `pacct` に書き込みます。プロセス課金レコードによって、プログラムの実行にかかわる統計情報を監視することができます。プロセスの実行に関する情報は、`ps` コマンドを使用して取得します。`accton` コマンドは `/var/adm/pacct` ファイルを作成し、プロセス課金をオンにします。

`pacct` ファイルは常に成長します。`ckpacct` コマンドは `pacct` ファイルのサイズを調べ、`pacct` ファイルが指定されたサイズよりも大きい場合には、`pacctn` ファイルを作成します。

データベース・ファイル `pacct` には、次のようなプロセスに関する情報が含まれています。

- プロセス・タイプ (たとえば子プロセス)
- プロセスの終了状態
- ユーザ識別番号
- グループ識別番号
- プロセスの起動端末
- 開始時間、ユーザ時間、システム時間、および CPU 時間
- 使用されたメモリの量
- 転送された I/O 文字数
- 読み書きされた 1024 バイトのブロックの数
- プロセスの起動に使用したコマンド名

プロセス課金ファイルのレコードは `tacct` 形式であり、ヘッダ・ファイル構造体 `acct` によって設定されます。ヘッダ・ファイル構造体 `acct` はヘッダ・ファイル `/usr/include/sys/acct.h` に定義され、最高 18 欄の課金情報を含んでいます。構造体メンバ `tacct` は、プライベート・ヘッダ・ファイル `tacct.h` に定義されます。

表 10-7 に、tacct 形式のファイルの欄番号、見出し、および説明を示します。

表 10-7: tacct ファイル・フォーマット

| 欄 | 見出し | 説明 |
|----|--------------|---|
| 1 | UID | /etc/passwd ファイルから得られるユーザ識別番号。 |
| 2 | LOGNAME | /etc/passwd ファイルから得られるユーザ・ログイン名。 |
| 3 | PRI_CPU | 就業時間内の CPU 実行時間。これは、就業時間内の CPU 実行時間のうちユーザに課金された合計時間 (秒単位) である。 |
| 4 | NPRI_CPU | 非就業時間内の CPU 実行時間。これは、非就業時間内の CPU 実行時間のうちユーザに課金された合計時間 (秒単位) である。 |
| 5 | PRI_MEM | 就業時間内のメモリ占有時間。これは、総 CPU 時間 (分単位) に、使用されたメモリの平均サイズを掛けたものである。 |
| 6 | NPRI_MEM | 非就業時間内のメモリ占有時間。これは、総 CPU 時間 (分単位) に、メモリの平均サイズを掛けたものである。 |
| 7 | PRI_RD/WR | 就業時間の動作時に転送された合計文字数。 |
| 8 | NPRI_RD/WR | 非就業時間の動作時に転送された合計文字数。 |
| 9 | PRI_BLKIO | 就業時間内の読み取りおよび書き込み動作時に転送された総 I/O ブロック数。I/O ブロックのバイト数は、実装方法によって異なる。 |
| 10 | NPRI_BLKIO | 非就業時間内の読み取りおよび書き込み動作時に転送された合計 I/O ブロック数。I/O ブロックのバイト数は、実装方法によって異なる。 |
| 11 | PRI_CONNECT | 接続が行われていた総就業時間数 (秒)。 |
| 12 | NPRI_CONNECT | 接続が行われていた総非就業時間数 (秒)。 |
| 13 | DSK_BLOCKS | 使用された総ディスク・ブロック数。 |
| 14 | PRINT | システムのプリンタにキュー登録された総ページ数。 |
| 15 | FEES | 課金対象のユニット数。この値は、シェル・スクリプト /usr/sbin/acct/chargefee で指定される。 |

表 10-7: tacct ファイル・フォーマット (続き)

| 欄 | 見出し | 説明 |
|----|-----------|---|
| 16 | PROCESSES | 課金期間中にユーザが生成した総プロセス数。 |
| 17 | SESS | 課金期間中のユーザの総ログイン回数。 |
| 18 | DSAMPS | DSK_BLOCKS 欄に指定されている総ディスク・ブロック数を調べるためにディスク課金コマンドが使用された回数の合計。DSK_BLOCKS 欄の値を DSAMPS 欄の値で除算することにより、課金期間中に使用された平均ディスク・ブロック数を求めることができる。 |

プロセス課金のシェル・スクリプトおよびコマンドを使用することによって、コマンド群およびそれらのコマンドを処理するために使用されたりソースに関する情報を統合することができます。以降の各項では、プロセス課金のシェル・スクリプトおよびコマンドについて説明します。

10.5.1 accton コマンド

accton コマンドは、プロセス課金を開始したり終了したりします。accton コマンドの構文は次のとおりです。

/usr/sbin/acct/accton [*filename*]

filename 変数を省略すると、プロセス課金は使用不能になります。
filename 変数を指定すると、プロセス課金が有効になり、カーネルは指定したファイルにプロセス課金レコードを書き込みます。通常、このファイルには `pacct` ファイルを指定しますが、別のプロセス課金データベース・ファイルを指定することもできます。指定するファイルは、`/var/adm` ディレクトリに存在し、ユーザ `adm` によって所有され、ログイン・グループ `adm` のメンバでなければなりません。

注意

シェル・スクリプト `runacct` および `turnacct` はプロセス課金データベース・ファイル `pacct` を使用するため、`pacct` ファイル以外のプロセス課金データベース・ファイルを指定するとシェル・スクリプト `runacct` および `turnacct` が影響を受けます。

10.5.2 turnacct シェル・スクリプト

turnacct シェル・スクリプトは、プロセス課金機能を制御し、プロセス課金ファイルを作成します。このシェル・スクリプトを使用するには、スーパーユーザでなければなりません。turnacct スクリプトの構文は次のとおりです。

turnacct [on | off | switch]

turnacct on シェル・スクリプトは、引数として pacct ファイルを指定して accton シェル・スクリプトを実行し、プロセス課金を使用可能に設定します。

turnacct off シェル・スクリプトは、引数を指定しないで accton コマンドを実行し、プロセス課金を使用不能に設定します。

turnacct switch シェル・スクリプトは、pacct ファイルの内容を pacctn ファイルに移動して、新しい pacct ファイルを作成します。

10.5.3 ckpacct シェル・スクリプト

pacct ファイルは拡張できます。指定した上限を超えて pacct ファイルが大きくなり、かつディスク・スペースが十分にある場合、ckpacct スクリプトは turnacct switch シェル・スクリプトを実行して pacct ファイルの内容を pacctn ファイルに移動し、新しい pacct ファイルを作成します。

ckpacct スクリプトを定期的に行うように、cron デーモンを設定することができます。詳細は、10.2.4 項を参照してください。

ckpacct シェル・スクリプトの構文は次のとおりです。

ckpacct [*blocksize*]

blocksize 変数には、pacct ファイルの上限サイズをディスク・ブロック単位で指定します。省略時のサイズは 500 ディスク・ブロックです。

ckpacct シェル・スクリプトを実行すると、/var/adm ディレクトリで利用できるディスク・ブロックの数が報告されます。使用可能なブロックの数が上限サイズより少ないと、プロセス課金はシェル・スクリプト turnacct off の呼び出しにより使用不能になります。また、診断メッセージが表示され、そのメッセージが環境変数 MAILCOM で指定されているアドレスに送

られます。環境変数 MAILCOM を下記のコマンドに設定するには、putenv 関数を使用します。

```
mail root adm
```

次の診断メッセージは、/var/adm ディレクトリに 224 のディスク・ブロックが残っていることを示しています。

```
ckpacct: /var/adm too low on space (224 blocks)
        "turning acctg off"
```

ckpacct シェル・スクリプトは、/var/adm ディレクトリに十分なスペースが確保されるまで、診断メッセージを表示し続けます。

10.5.4 acctcom コマンド

acctcom コマンドは、プロセス課金レコードの要約を表示します。コマンド・オプションを使用すれば、出力のタイプと形式を指定することができます。acctcom コマンドを使用するには、スーパーユーザである必要はありません。

acctcom コマンドは、終了したプロセスに関する情報だけを表示します。実行中のプロセスについての情報を表示するには、ps コマンドを使用します。acctcom コマンドの構文は次のとおりです。

```
/usr/bin/acctcom [ option... ] [ filename... ]
```

filename 変数を省略すると、pacct ファイルを使用してプロセス課金レコードを取得します。*filename* 変数を指定することにより、ヘッダ・ファイル acct.h を持つ別のプロセス課金ファイルを指定することができます。2 つ以上の *filename* 変数を指定すると、acctcom コマンドは入力順にファイルを読み取ります。

コマンド・オプションを指定しない場合は、省略時の出力として、欄ヘッダ形式の次のような情報が含まれます。

- 課金機能が開始された日付と時間
- コマンド名
- ユーザ名
- 端末名
- プロセス起動時間

- プロセス終了時間
- 実時間 (秒)
- CPU 時間 (秒)
- 平均メモリ・サイズ (単位 K バイト)

このコマンドのオプションについての詳細は、`acctcom(8)` を参照してください。

省略時のプロセス課金の要約の出力例は次のとおりです。

```
# /usr/bin/acctcom /var/adm/pacct1
ACCOUNTING RECORDS FROM: Mon Jun 17 02:00:00 2002
COMMAND          START      END      REAL    CPU    MEAN
NAME      USER   TTYNAME  TIME      TIME      (SECS) (SECS) SIZE(K)
#sa       root   ttypl    11:59:00  11:59:00   0.77   0.01   0.00
ls        root   ttypl    11:59:04  11:59:04   0.11   0.01   0.00
uugetty   root   ?        11:58:39  11:59:48  69.53   0.01   0.00
#ls        root   ttypl    11:59:55  11:59:55   0.30   0.01   0.00
uugetty   root   ?        11:59:49  12:00:58  69.48   0.01   0.00
cp         adm    ?        12:05:01  12:05:01   0.33   0.01   0.00
chmod     adm    ?        12:05:01  12:05:01   0.27   0.01   0.00
#df        adm    ?        12:05:02  12:05:02   0.38   0.01   0.00
awk        adm    ?        12:05:02  12:05:02   0.58   0.01   0.00
sed        adm    ?        12:05:02  12:05:02   0.56   0.01   0.00
```

10.5.5 sa コマンド

`sa` コマンドは、プロセス課金情報を要約します。このコマンドを使用すれば、大量の課金情報を管理しやすくなります。`sa` コマンドで作成されたファイルには、有効なプロセス課金情報がすべて含まれています。`sa` コマンドの構文は次のとおりです。

```
/usr/sbin/sa [ options... ] [ filename ]
```

filename 変数には、`acct.h` ヘッダ・ファイル構造体を持つプロセス課金ファイルを指定します。*filename* 変数を省略すると、`pacct` ファイルが使用されます。

オプションを指定しないで `sa` コマンドを実行すると、省略時の出力は、見出しのない6つの欄で構成されることになります。あるコマンド・オプションは、欄が6つに増えてより多くの情報を含むことができます。オプションを指定して、フォーマットを変更したり、識別接尾語を含む補足情報を

出力することができます。このコマンドのオプションの詳細は、sa(8)を参照してください。

次に示すのは、sa コマンドの出力の省略時の形式の例です。

| # | /usr/sbin/sa | | | | |
|-----|--------------|---------|-------------|----|-----------|
| 798 | 277.24re | 0.08cpu | 3248790avio | 0k | |
| 7 | 33.42re | 0.08cpu | 103424avio | 0k | csh |
| 14 | 0.08re | 0.00cpu | 127703avio | 0k | mv |
| 40 | 0.34re | 0.00cpu | 159968avio | 0k | cp |
| 2 | 0.01re | 0.00cpu | 132448avio | 0k | acctwtmpt |
| 34 | 0.13re | 0.00cpu | 133517avio | 0k | chmod |
| 23 | 0.10re | 0.00cpu | 139136avio | 0k | chgrp |
| 25 | 0.11re | 0.00cpu | 144768avio | 0k | chown |
| 36 | 0.15re | 0.00cpu | 133945avio | 0k | dspmsg |
| 32 | 0.18re | 0.00cpu | 134206avio | 0k | cat |
| | | | | | 6 |
| | | | | 5 | |
| | | | 4 | | |
| | | 3 | | | |
| | 2 | | | | |
| 1 | | | | | |

1 コマンドの実行回数

-c オプションを指定すると、全コマンドの実行回数に占める、そのコマンドの実行回数の比率を示す欄が、さらに追加されます。

2 実時間の合計

-c オプションを指定すると、実時間の比率を示す欄が、さらに追加されます。

3 CPU 時間

オプションを指定することによって、システム CPU 時間とユーザ CPU 時間の合計、ユーザ CPU 時間、システム CPU 時間、あるいはシステム CPU 時間に対するユーザ CPU 時間の比率を、それぞれ欄に表示できます。

-c オプションを指定すると、実時間の比率を示す欄がさらに追加されます。また、-t オプションを指定すると、ユーザ CPU 時間とシステム CPU 時間の合計に対する実時間の比率を示す欄が、さらに追加されます。

④ ディスク入出力動作についての情報 (I/O 動作の平均回数または総数)

- 5 メモリ時間全体で使用されたブロック数

K ブロック単位 (ブロック数に 1024 を積算した値)。

- ## 6 コマンド名

次の例では、下記のような各比率を表示する 3 つの欄が、省略時の形式に追加されています。

```
# /usr/sbin/sa -c
645 100.00% 324.10re 100.00% 0.02cpu 100.00% 6171050avio 0k
  2   0.31%  25.70re   7.93% 0.02cpu 100.00%  107392avio 0k  csh
  6   0.93%   0.04re   0.01% 0.00cpu   0.00%  132928avio 0k  mv
 38   5.89%   0.33re   0.10% 0.00cpu   0.00%  163357avio 0k  cp
  2   0.31%   0.01re   0.00% 0.00cpu   0.00%  132992avio 0k  cat
 26   4.03%   0.11re   0.03% 0.00cpu   0.00%  136832avio 0k  chmod
 24   3.72%   0.10re   0.03% 0.00cpu   0.00%  139824avio 0k  chgrp
```

追加の欄では、次の情報を表示します。

- ① 全コマンドの総実行回数に対する各コマンドの実行回数の比率を示す。
- ② 欄 1 に指定されている回数だけ、そのコマンドの実行に要した総実時間合計の、全コマンドの実行に要した総実時間に対する比率を示す。
- ③ 全コマンドの実行に要した総 CPU 時間に対して、欄 1 で指定されている回数だけの当該コマンドの実行に要した総 CPU 時間の比率を示す。

10.5.6 acctcms コマンド

acctcms コマンドは、プロセス課金レコードから、ASCII 形式およびバイナリ形式のコマンド要約ファイルを作成します。pacct ファイルなどの、`/usr/include/sys/acct.h` ヘッダ・ファイル構造体を持っているプロセス課金ファイルを指定します。acctcms コマンドは、レコードをソートして、課金期間中に使用された個々のコマンドについての統計情報を 1 つのレコードに統合します。このレコードから、最も頻繁に使用されたコマンドおよび最も多くのシステム時間を使用したコマンドが分かります。

CMS 状態において `runacct` シェル・スクリプトが `acctcms` コマンドを実行します。このコマンドを手動で実行して、コマンド要約レポートを作成することもできます。

acctcms コマンドの構文は次のとおりです。

/usr/sbin/acct/acctcms [-acjnopst] *filename...*

オプションを指定しないで **acctcms** コマンドを実行すると、メモリ占有時間の多い順に出力されます。メモリ占有時間は、プロセスによって使用されたメモリの K バイト数に、使用されたバッファ時間を積算した値です。省略時の設定は、バイナリ出力です。メモリ占有時間は、次の計算によって求められます。

メモリ占有時間 = [(秒単位の CPU 時間) * (平均メモリ・サイズの k バイト数)] / 60

このコマンドのオプションの詳細は、**acctcms(8)** を参照してください。

注意

acctcms コマンドを使用して要約ファイルを ASCII 形式で作成すると、個々のコマンド・レコードは 81 文字以上で構成され、10 文字/インチの固定幅フォントが指定されている場合は全体幅が 8.5×11 インチの用紙を使用することができます。レコードの一部が欄の幅を超えると、その部分は次の行に移動されます。

次に示すのは、COMMAND NAME 欄で *****other** が指定されている行に、1 回しか実行されなかったコマンド群についての統計情報が表示されている ASCII 出力の例です。

```
# acctcms -a -j /var/adm/pacct1
TOTAL COMMAND SUMMARY
COMMAND NUMBER TOTAL TOTAL TOTAL MEAN MEAN HOG CHARS BLOCKS
NAME CMDS KCOREMIN CPUMIN REALMIN SIZEK CPUMIN FACTOR TRNSFD READ
TOTALS 9377 0.00 0.36 26632.67 0.00 0.00 0.00 17768213 100529
chmod 34 0.00 0.00 .15 0.00 0.00 0.07 5785856 64
ln 4 0.00 0.00 0.01 0.00 0.00 0.78 422016 16
xterm 9 0.00 0.03 537.41 0.00 0.00 0.00 22948288 536
getcons 8 0.00 0.00 0.14 0.00 0.00 0.07 26636992 102
cfe2.20 4 0.00 0.00 0.09 0.00 0.00 0.12 182464 155
dump 22 0.00 0.00 14.91 0.00 0.00 0.00 69402112 128
whoami 4 0.00 0.00 0.03 0.00 0.00 0.36 7405952 27
restore 40 0.00 0.00 49.16 0.00 0.00 0.00 34247488 1316
***other 25 0.00 0.00 3546.88 0.00 0.00 0.00 35904984 737
hostname 2 0.00 0.00 0.01 0.00 0.01 0.94 223104 14
```

HOG FACTOR (使用効率) は、総 CPU 時間を実時間合計で除算した値です。

10.5.7 acctprcl コマンド

acctprcl コマンドは、**/usr/include/sys/acct.h** ヘッダ・ファイル構造体を持つファイルからプロセス課金レコードを読み取り、ユーザ識別番

号と対応するログイン名を追加し、ASCII 形式でレコードを表示します。ログイン・セッション・レコードは、ユーザ識別番号およびログイン名に従ってソートされます。

使用しているシステムに、同じユーザ識別番号を持つ複数のユーザが存在する場合は、`pacct` ファイルではなく、`/var/adm/acct/nite` ディレクトリ内のプロセス課金ファイルを使用しなければなりません。

PROCESS 状態において `runacct` シェル・スクリプトが `acctprcl` コマンドを実行します。また、このコマンドは手動で実行することもできます。`acctprcl` コマンドの構文は次のとおりです。

`/usr/sbin/acct/acctprcl` [*filename*]

filename 変数には、`/usr/include/utmp.h` ヘッダ・ファイル構造体で定義された形式でログイン・セッションのリストが入っているファイルを指定します。*filename* 変数を省略すると、ログイン名は `/etc/passwd` ファイルから得られます。

このコマンドの出力は、見出しのない次の 7 つの欄を持つ形式で情報を指定します。

- ユーザ識別番号
- ログイン名
- プロセスが就業時間中に使用した CPU 時間 (秒)
- プロセスが非就業時間中に使用した CPU 時間 (秒)
- 転送された文字の総数
- 読み取りおよび書き込みが行われたブロックの総数
- 平均メモリ・サイズ (K バイト単位)

次に示すのは、`acctprcl` コマンドとその出力例です。

```
# /usr/sbin/acct/acctprcl < /usr/adm/pacct
0   root      0    1   17228   172    6
4   adm       0    6   46782   46    16
0   root      0   22  123941  132    28
9261 hoffmann  6    0   17223   22    20
9   lp        2    0   20345   27    11
9261 hoffmann  0  554  16554   20   234
```


10.5.8 acctprc2 コマンド

acctprc2 コマンドは、acctprc1 コマンドが作成したレコードを読み取り、ユーザ識別番号とログイン名に従ってそれらのレコードを要約し、ソートされた要約結果を総合課金バイナリ・レコードとして tacct ファイル・フォーマットで表示します。acctprc2 コマンドによって作成されたバイナリ・ファイルは、acctmerg コマンドを使用して日次要約課金レコード・ファイルを作成することにより、他の総合課金ファイルとマージすることができます。

PROCESS 状態において runacct シェル・スクリプトが acctprc2 コマンドを実行します。このコマンドは手動でも実行することができます。

10.5.9 lastcomm コマンド

lastcomm コマンドは、pacct ファイルにあるコマンド実行情報を、時間の新しい順に表示します。

各プロセスについて、次の情報が表示されます。

- コマンド名
- S フラグまたは F フラグ
S フラグは、当該コマンドがスーパーユーザによって実行されたことを示します。F フラグは、当該コマンドが子プロセスの作成後に実行され、後続のシステム・コール exec がなかったことを示します。
- コマンドを発行したユーザの名前
- コマンドを発行した端末
- 使用した CPU 時間 (秒数)
- プロセスを起動した時間

lastcomm コマンドの構文は次のとおりです。

/usr/bin/lastcomm [*command*] [*username*] [*tty*]

次の例は、root によって実行された sed コマンドに関する情報を示したものです。

```
# lastcomm sed root
sed      S   root      tty0      0.01 secs Fri Jan 21 11:34
sed      S   root      tty0      0.01 secs Fri Jan 21 11:34
```

10.6 ディスク使用量課金機能

ディスク使用量課金機能は、シェル・スクリプト `dodisk` によって実行されます。`dodisk` シェル・スクリプトは、`diskusg` コマンドまたは `acctdusg` コマンドを使用して、中間 ASCII ファイル `/var/adm/dtmp` に情報を書き込みます。`dodisk` スクリプトは、中間ファイルを `acctdisk` コマンドの入力として使用してバイナリ総合課金データベース・ファイル `/var/adm/acct/nite/dacct` を作成します。`dodisk` スクリプトは、すべてのファイル・システムあるいはファイル・システム・データベース・ファイル `/etc/fstab` で指定されているファイル・システムでディスク課金機能を実行します。

`/var/adm/acct/nite/dacct` ファイルに他の課金情報を組み込むことにより、完全な課金レポートを作成することができます。次に例を示します。

```
# /usr/sbin/acct/dodisk
# /usr/sbin/acct/prtacct /var/adm/acct/nite/dacct
```

10.6.1 `dodisk` シェル・スクリプト

ディスク使用量課金情報を取得するには、`dodisk` シェル・スクリプトを使用します。`dodisk` スクリプトは、自動的に実行されるように `cron` デーモンに対して設定することも、手動で実行することもできます。`dodisk` シェル・スクリプトの構文は次のとおりです。

```
/usr/sbin/acct/dodisk [-o] [ filesystem...]
```

```
/usr/sbin/acct/dodisk [ device special file...]
```

-o オプションを使用することにより、UFS ファイル・システムあるいは AdvFS ファイルセットのマウント・ポイントに対してディスク使用量課金機能を実行できるように `filesystem` 変数を指定することができます。-o オプションを指定しない場合は、次の例のように、この変数には `raw` デバイス特殊ファイルあるいはキャラクタ・デバイス特殊ファイルを指定する必要があります。

```
# /usr/sbin/acct/dodisk /dev/rdisk/dsk3c
```

引数の指定を省略すると、ディスク課金機能はデータベース・ファイル `/etc/fstab` に記述されている UFS デバイス特殊ファイルに対して実行されます。詳細については、`fstab(4)` を参照してください。

注意

スワップ領域が `/etc/fstab` ファイルに指定されていると、`dodisk` シェル・スクリプトは正しく実行されません。この場合、特定のファイル・システムだけを使用するように `dodisk` シェル・スクリプトを編集するか、`dodisk` シェル・スクリプトを実行して課金情報が必要なファイル・システムを指定します。

-o オプションを指定すると、シェル・スクリプト `dodisk` は、`diskusg` コマンドではなく `acctdusg` コマンドを使用して、より完全であるが低速なバージョンのディスク課金機能を実行します。-o オプションを指定して `filesystem` 変数を指定する場合は、デバイス特殊ファイル名ではなくマウント・ポイントを指定します。

10.6.2 diskusg コマンド

`diskusg` コマンドは、ディスク課金レコードを表示します。`diskusg` コマンドは、`/etc/passwd` ファイルからユーザ・ログイン名と識別番号を取得します。`diskusg` コマンドの構文は次のとおりです。

`/usr/sbin/acct/diskusg` [-options] [*filesystems*]

このコマンドのオプションの詳細は、`diskusg(8)` を参照してください。

`diskusg` コマンドは、ASCII 形式ファイル `/var/adm/dtmp` を出力として生成します。このファイルは、ASCII 形式レコードをバイナリ形式の総合課金レコードに変換し `/var/adm/acct/nite/dacct` ファイルを生成する `acctdisk` コマンドの入力として使用されます。これらのレコードを他の課金レコードとマージすることにより日次総合課金レポートが作成されます。

`diskusg` コマンドによって作成される各出力レコードには、ユーザ識別番号、ログイン名、およびユーザに割り当てられたディスク・ブロックの総数が含まれます。`diskusg` コマンドはユーザ `i` ノード・レコードを検査するため、空のディレクトリを含め、すべてのディスク・スペースが課金の対象となります。

次に示すのは `diskusg` コマンドの実行例です。

```
# /usr/sbin/acct/diskusg /dev/rdisk/dsk3c
0  root          63652
1  daemon        84
2  bin           71144
```

| | | |
|------|----------|------|
| 4 | adm | 976 |
| 5 | uucp | 3324 |
| 322 | homer | 2 |
| 521 | whistler | 2 |
| 943 | cellini | 363 |
| 1016 | pollock | 92 |
| 1098 | hopper | 317 |

ファイル・システムに対して raw デバイス特殊ファイルを指定する必要があります (たとえば /dev/rdisk/dsk3c)。ファイル・システムはターゲット・デバイス上になければなりません。

10.6.3 acctdusg コマンド

acctdusg コマンドは、diskusg コマンドよりも完全なディスク課金情報を取得します。dodisk シェル・スクリプトを -o オプション付きで実行すると、acctdusg コマンドが呼び出されて /var/adm/dtmp ファイルが作成されます。

acctdusg コマンドの構文は次のとおりです。

acctdusg [-u *filename*] [-p *filename*]

このコマンドのオプションの詳細は、acctdusg(8) を参照してください。

このコマンドの出力ファイルにはディスク使用量バイナリ・ファイル (通常は /var/adm/dtmp) を使用します。acctdusg コマンドは、dodisk シェル・スクリプトによって実行されると、dodisk スクリプトで指定されたファイル・システムを入力として使用します。

acctdusg コマンドへの入力には、通常、find / -print コマンドから引き渡されたファイルのリストです。このコマンドは、ファイル・パス名とユーザのログイン・ディレクトリ (\$HOME) を比較します。ファイル・パス名がユーザのログイン・ディレクトリと同じであれば、当該ユーザはそのファイルについて課金されます。したがって、ファイルが置かれているディレクトリは、ディスク・スペースに対してユーザに課金する際の決定要素となります。ログイン・ディレクトリ以外のディレクトリに存在するファイルが使用したディスク・ブロック数は、-u オプションを使用して表示することができます。

acctdusg コマンドは、各ファイルごとに、ファイルに割り当てられたディスク・ブロック (隠しブロックや間接ブロックを含む) の数をハード・リンクの数で除算した値を計算します。2 人以上のユーザが同じファイルとのリン

クを持っている場合は、acctdusg コマンドは、当該ファイルの総ディスク・スペースの料金をユーザ数で割った値で各ユーザに課金します。

acctdusg コマンドの出力は、ユーザ識別番号、ユーザ名、およびユーザが所有しているすべてのファイルの計算値の合計を、3つの欄に表示し、ユーザ識別番号の前に0(ゼロ)を付けます。acctdusg コマンドは、空のディレクトリのディスク・ブロック数は表示しません。

10.6.4 acctdisk コマンド

acctdisk コマンドは、バイナリ形式の総合課金ファイルを作成します。acctdisk コマンドがdodisk スクリプトから実行されると、diskusg コマンドまたはacctdusg コマンドで作成された /var/adm/dtmp ファイルを読み取り、変換されたバイナリ・レコードを一時ファイルに書き込みます。この一時ファイルは、その後、/var/adm/acct/nite/dacct ファイルに改名されます。

通常、acctdisk コマンドで作成されたディスク使用量課金レコードは他の課金レコードとマージされて、総合課金レポートが作成されます。

10.7 システム管理サービス課金機能

スーパーユーザは、システム管理の各サービス料をユーザに請求することができます。たとえば、次のサービスに対して課金することができます。

- ディスクへのファイル・バックアップ
- ディスクからのファイルの復元
- テープへのファイル・バックアップ
- テープからのファイルの復元
- 電話によるソフトウェア技術支援
- 係員によるソフトウェア技術支援

chargefee シェル・スクリプトを使用すれば、ユーザへのサービスに対して課金することができます。この場合、各サービスの課金額を決定する必要があります。また、サービスの実行に要した時間に応じて異なる課金率を設定することができます。

課金単位は、fee ファイルに収集されます。各ユーザ名に対する課金単位数によって、システム管理作業の料金を決定します。chargefee シェル・ス

クリプトは、必要に応じて `fee` ファイルを作成し、ユーザ識別番号、ユーザ名、および課金単位で構成されるレコードを書き込みます。

`chargefee` シェル・スクリプトの構文は次のとおりです。

```
/usr/sbin/acct/chargefee user_name units
```

`units` 変数にダッシュ (-) を指定することによって課金単位を減算することができます。

次の例では、7 課金単位を `josh` というユーザに請求しています。

```
# chargefee josh 7
```

上記のコマンドが発行されると、次のレコードが `/fee` ファイルに書き込まれます。

```
1114 josh 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0
```

10.8 プリンタ課金機能

課金機能が使用可能になっているプリンタを使用すると、プリンタ課金ファイルにレコードが書き込まれます。プリンタ課金レコードは、固有の形式を持っており、次の情報が含まれています。

- プリント要求を出したホストおよびユーザの名前
- 出力した用紙のページ数またはフィート数
- プリンタを使用した回数
- プリント出力の単位あたりの料金

プリンタ課金レコードによって、スーパーユーザは、システム・プリント・リソースを使用したユーザに課金し、プリンタ使用状況を追跡することができます。

2 つのプリンタ課金ファイルが、ディレクトリ `/var/adm` または `/var/adm/printer` に置かれます。プリンタ・ユーザ・ファイル `printer.acct` は、マシンおよびユーザごとに、使用したプリント用紙の量と費用をリストします。プリンタ要約ファイル `printer.acct_sum` は、マシンおよびユーザごとに、使用された用紙についての要約をリストします。`printer` 変数には、プリンタ名を指定します。プリンタ課金ファイルの作成に関しては、第 8 章を参照してください。

プリンタ使用量に関するレポートの作成には `pac` コマンドを使用します。ただし、`pac` コマンドは、課金機能が使用可能になっているプリンタに関する情報のみを取得します。`pac` コマンドの構文は次のとおりです。

```
pac [-cmrs] [-p price] [-P printer] [ user...]
```

このコマンドのオプションの詳細は、`pac(8)` を参照してください。

10.9 日次、要約、月次の各レポート・ファイルの作成

日次レポート・ファイル、要約レポート・ファイル、月次レポート・ファイルを `/var/adm/acct/nite`、`/var/adm/acct/sum`、および `/var/adm/acct/fiscal` の各ディレクトリに作成するために使用できる 4 つのシェル・スクリプトおよび 1 つのコマンドがあります。

| コマンド | 説明 |
|-----------------------|---|
| <code>runacct</code> | このシェル・スクリプトは、ディレクトリ <code>/var/adm/acct/nite</code> および <code>/var/adm/acct/sum</code> に、日次ファイルと要約ファイルを作成する。 |
| <code>acctmerg</code> | このコマンドは、総合課金レコード・ファイルをマージするコマンドで、これによってユーザは、ファイル、プロセス接続時間、料金、ディスク使用量、プリント・キュー課金レコードを統合して、指定したフォーマットでファイルを作成することができる。出力は、省略時の設定であるバイナリ形式か、または ASCII 形式で行うことができ、最高 18 欄の課金情報を含めることができる。 |
| <code>prtacct</code> | このシェル・スクリプトは、 <code>/usr/include/sys/acct.h</code> ヘッダ・ファイル構造体を持つ課金ファイルをフォーマットして表示する。個々のレコードには、ユーザ識別番号、接続時間、プロセス時間、ディスク使用量、およびプリンタ使用量に関する情報が含まれる。 |
| <code>prdaily</code> | このシェル・スクリプトは、前日からの課金データを内容とする ASCII 形式のファイルを作成する。このスクリプトは、 <code>runacct</code> スクリプトから実行されると、 <code>/var/adm/acct/sum/rprtmmdd</code> ファイルを作成する。 |
| <code>monacct</code> | このシェル・スクリプトは、 <code>/var/adm/acct/fiscal</code> ディレクトリに累積プロセス課金ファイルおよび総合課金ファイルを作成する。 |

以降の各項で、それぞれのシェル・スクリプトおよびコマンドについて詳しく説明します。

10.9.1 runacct シェル・スクリプト

runacct シェル・スクリプトは、課金シェル・スクリプトおよびコマンドを使用して、接続時間、料金、ディスク使用量、キュー、およびプロセス課金の各データベース・ファイル进行处理し、ディレクトリ `/var/adm/acct/nite` および `/var/adm/acct/sum` に、日次ファイルと要約ファイルを作成します。

`/var/adm/acct/nite` ディレクトリには、runacct スクリプトによって毎日再使用されるファイルが含まれています。これらのファイルのうちのいくつかには、バイナリ形式のファイルが `/var/adm/acct/sum` ディレクトリにあります。このディレクトリには、累積要約ファイルもあります。累積要約ファイルは、runacct シェル・スクリプトによって作成および更新されるほか、月次レポートを作成するために monacct シェル・スクリプトによって使用されます。

cron デーモンを設定して runacct シェル・スクリプトを毎日実行させることも、runacct シェル・スクリプトを手動で実行することもできます。runacct シェル・スクリプトが完了しなかったり、このスクリプトで作成されたファイルが破壊されたり失われたりすると、このコマンドを手動で実行しなければならない場合があります。

runacct シェル・スクリプトを実行すると一時ファイル `/var/adm/acct/nite/lock` が作成されます。
`/var/adm/acct/nite/lock` ファイルが存在する場合、runacct シェル・スクリプトは実行されません。

runacct シェル・スクリプトは、次の表に示された順序で、13 の実行状態があり、どの状態でも再起動することができます。

| 状態 | 説明 |
|----------|--|
| SETUP | 課金ファイルのうちのいくつかを設定する。 |
| WTMPFIX | acctcon1 などのコマンドの異常終了の原因になる不正な日付およびタイム・スタンプ・エントリを修正する。 |
| CONNECT1 | 接続セッション・レコードを記述する。 |
| CONNECT2 | 接続セッション・レコードを使用して、バイナリ総合課金レコードを作成する。バイナリ課金レコードは、他のレコードとマージされて、日次レポートになる。 |
| PROCESS | プロセス課金レポート・ファイルを作成する。 |

| 状態 | 説明 |
|-----------|---|
| MERGE | acctmerge コマンドを使用して、バイナリ総合課金ファイルを作成する。 |
| FEES | acctmerge コマンドを使用して、fee ファイルのレコードをバイナリ総合課金ファイルにマージする。 |
| DISK | acctmerge コマンドを使用して、ディスク使用量レコードをバイナリ総合課金ファイルにマージする。 |
| QUEUEACCT | acctmerge コマンドを使用して、プリント・キュー課金レコードをバイナリ総合課金ファイルにマージする。 |
| MERGEACCT | バイナリ総合課金ファイルを、日次総合課金ファイルにコピーする。この日次ファイルは、累積日次総合課金ファイルを作成するために、acctmerge コマンドへの入力として使用される。 |
| CMS | コマンド使用量要約を作成する。 |
| USEREXIT | システム固有のシェル・スクリプトを実行する。 |
| CLEANUP | 一時ファイルを削除する。 |

10.9.1.1 runacct シェル・スクリプト・エラーの修正

runacct シェル・スクリプト・エラーが発生すると、メッセージがコンソール・デバイスに書き込まれ、ロック・ファイルが削除されて、診断ファイルとエラー・メッセージが保存された後、処理が停止します。runacct シェル・スクリプト・エラーが発生したかどうかは、次の情報から判定できます。

- スクリプトが正常終了すると、/var/adm/acct/nite/active ファイルが作成される。

runacct シェル・スクリプトは、メッセージをこのファイルに記録します。このファイルを使用してどのタスクが正常終了したかを判定することができます。次に示すのは active ファイルの例です。

```
Fri Jul 5 11:02:56 EST 2002
-rw-r--r-- 1 adm adm      0 Jul 01 03:00 /var/adm/acct/nite/dacct
-rw-rw-r-- 1 root system 924 Jun 05 10:45 /var/adm/wtmp
-rw-rw-r-- 1 adm adm      0 Jun 08 13:46 fee
-rw-rw-r-- 1 adm adm      0 Jun 07 02:00 pacct
-rw-rw-r-- 1 adm adm    8904 Jun 02 11:02 pacct1
files setups complete
wtmp processing complete
connect acctg complete
process acctg complete for /var/adm/Spacct1.1101
process acctg complete for /var/adm/Spacct2.1101
all process acctg complete for 1101
tacct merge to create daytacct complete
no fees
no disk records
```

```
no queueing system records
updated sum/tacct
command summaries complete
system accounting completed at Fri
```

- スクリプトが正常終了しない場合、`/var/adm/acct/nite/activemmdd` ファイルが作成される。

このファイルには、スクリプトの実行に関する情報が含まれており、これを使用してスクリプトの実行ができなかった箇所を調べることができます。

- `/var/adm/acct/nite/statefile` ファイルには、`runacct` シェル・スクリプトが実行した最後の状態の名前が記述されている。

`runacct` シェル・スクリプトはこの状態を正常に終了しなかった可能性があります。

- `/var/adm/acct/nite/lastdate` ファイルには、最後に `runacct` シェル・スクリプトが実行された日付が記述されている。

このファイルに指定されている日付が現在の日付である場合は、このシェル・スクリプトは実行されません。

`runacct` シェル・スクリプトが中断または異常終了した場合は、前回正常終了したときの状態でスクリプトを再起動する必要があります。

`/var/adm/acct/nite/statefile` ファイルには、最後に実行された状態の名前が記述されています。

`runacct` シェル・スクリプトの構文は次のとおりです。

`/usr/sbin/acct/runacct [mmdd] [state]`

`mmdd` 変数には、`runacct` シェル・スクリプトを実行する日付を指定します。`state` 変数には、`runacct` スクリプトの実行状態を指定します。

2 日以上連続して `runacct` シェル・スクリプトが実行できない場合は、`SETUP` 状態に設定し、スクリプトを手動で実行してください。

`runacct` シェル・スクリプトを再起動する前に、`/var/adm/acct/nite/lock` ファイルと `/var/adm/acct/nite/lastdate` ファイルを削除する必要があります。

次の例では、1 月 26 日からの課金データベース・ファイルを使用して、`runacct` シェル・スクリプトが `MERGE` 状態で実行されます。

```
# runacct 0126 MERGE > /var/adm/nite/fd2log&
```

次の例では、1月26日からの課金データベース・ファイルを使用してシェル・スクリプト `nohup` コマンドを実行し、シグナル、ハングアップ、ログアウト、および取り消し終了が無視されるように指定します。その実行中に生成されたエラー・メッセージは、`fd2log` ファイルに書き込まれます。

```
# nohup runacct 0126 > /var/adm/acct/nite/fd2log&
```

10.9.1.2 エラーの例と対策

以下に、エラーの例と、その問題を解決するための対策を示します。

```
ERROR: locks found. run aborted
```

`/var/adm/acct/nite/lock` ファイルが存在します。このファイルを削除して、最後に終了したときの状態で `runacct` シェル・スクリプトを再起動してください。

```
ERROR: acctg already run for Fri : check Jan
```

現在の日付が、`/var/adm/acct/nite/lastdate` ファイルに指定されている日付と同じになっています。このファイルを削除し、最後に終了したときの状態で `runacct` シェル・スクリプトを再起動してください。

```
ERROR: runacct called with invalid arguments
```

`runacct` シェル・スクリプトで、誤った引数を指定しています。

```
ERROR: turnacct switch returned rc=?
```

`turnacct switch` シェル・スクリプトから `accton` コマンドを実行したときに異常終了しました。`accton` コマンドの保護を調べ、`adm` ユーザがこのコマンドを実行できるように設定してください。

```
ERROR: Spacct?.mmdd already exists run setup manually
```

`runacct` シェル・スクリプトを `MERGE` 状態で手動で実行しなければなりません。疑問符 (?) は、1文字のワイルドカードを示します。実際のファイル名には `var/adm/Spacct1.1101` のように、バージョン番号と日付が入っています。

```
ERROR: wtmpfix errors see nite/wtmperror
```

`WTMPFIX` 状態のときに、修復不能な `wtmp` ファイルが検出されました。`fwtmp` コマンドを使用してファイルを修正してください。

```
ERROR: invalid state, check /usr/var/adm/nite/active
```

処理中に、runacct シェル・スクリプトが壊れた active ファイルを検出した可能性があります。/var/adm/acct/nite/active* ファイルおよび statefile ファイルを調べてください。

10.9.2 acctmerg コマンド

acctmerg コマンドは、プロセス、接続時間、料金、ディスク使用量、およびキューの各総合課金レコードを tacct ファイル形式で統合します。たとえば、特定のログイン名およびユーザ識別番号を持つ複数の総合課金レコードをマージして、そのログイン名とユーザ識別番号を持つ 1 つのレコード・グループを作成することができます。ファイル・レコードは通常、ユーザ識別番号またはユーザ・ログイン名の順にマージされます。

省略時のコマンド出力はバイナリ形式ですが、ASCII 形式の出力も生成することができます。省略時の acctmerg コマンド出力は、/usr/include/sys/acct.h ヘッダ・ファイル構造体を持ち、最高 18 欄の情報を含みます。倍精度の 2 つの要素の配列として指定されているデータ型を含む /usr/include/sys/acct.h ヘッダ・ファイル構造体を持つレコードは、就業時間の値と非就業時間の値の両方を含むことができます。

acctmerg コマンドは、runacct シェル・スクリプトによって実行されます。また、このコマンドを手動で実行してレポートを作成することもできます。acctmerg コマンドの構文は次のとおりです。

```
/usr/sbin/acct/acctmerg [-ahiptuv] [#] [ file...]
```

最高 9 つの総合課金レコード・ファイルを指定することができます。ファイルを指定しない場合は、レコードは標準入力から読み取られます。

このコマンドのオプションの詳細は、acctmerg(8) を参照してください。

次の例は、ASCII 形式のディスク課金ファイル /var/adm/acct/nite/dacct から、UID、LOGNAME、DSK_BLOCKS、および DSAMPS の欄エントリを読み取って、総合課金ファイル /var/adm/acct/sum/tacct のバイナリ・レコードにマージします。

```
# acctmerg -il-2, 13, 18 < nite/dacct | sum/tacct
```

acctmerg コマンドを使用して、`/var/adm/sum/tacct` ファイルのエラーを修正することができます。このファイルで発生する可能性のあるエラーは、負数と、ユーザ識別番号の重複です。

現在の `/var/adm/sum/tacct` ファイルのエラーを修正するには、次の手順に従ってください。

1. 作業ディレクトリを `/var/adm/sum` に変更します。
2. `prtacct` コマンドを使用して、`/var/adm/sum/tacctprev` ファイルを表示します。

`/var/adm/sum/tacctprev` ファイルの内容が正しい場合は、
`/var/adm/sum/tacctmdd` ファイルに問題がある可能性があります。
この例では、`/var/adm/sum/tacctmdd` ファイルを修正することで問題が解決することを想定しています。
3. 次のコマンドを使用して、`/var/adm/sum/tacctmdd` ファイルの ASCII バージョンを取得します。


```
# acctmerg -v < tacct.0617 > tacct_temp
```
4. 一時ファイルを編集し、必要に応じてレコードを修正します。
5. 次のコマンドを使用して、`/var/adm/sum/tacctmdd` ファイルを作成し直します。


```
# acctmerg -i < tacct_temp > tacct.0617
```
6. 次のコマンドを使用して、`/var/adm/sum/tacct` ファイルを作成し直します。


```
# acctmerg tacctprev < tacct.0617 > tacct
```

10.9.3 prtacct シェル・スクリプト

`prtacct` シェル・スクリプトは、`tacct` ファイル形式のバイナリ総合課金ファイルを ASCII 形式で表示します。このスクリプトを使用すれば、接続時間、プロセス時間、ディスク使用量、またはプリンタ使用量のレポート・ファイルを作成することができます。

`prtacct` シェル・スクリプトは、`monacct` シェル・スクリプトおよび `prdaily` シェル・スクリプトによって実行されます。`prdaily` シェル・スクリプトは、`CLEANUP` 状態の `runacct` シェル・スクリプトによって実行されます。`prtacct` シェル・スクリプトの構文は次のとおりです。

/usr/sbin/acct/prtacct [-f *column*] [-v] *file*

このコマンドのオプションの詳細は、`prtacct(8)` を参照してください。

10.9.4 **prdaily** シェル・スクリプト

`prdaily` シェル・スクリプトは、前日からの課金データのレポートを ASCII 形式で作成します。`prdaily` シェル・スクリプトは、`CLEANUP` 状態の `runacct` シェル・スクリプトによって実行され、`/var/adm/acct/sum/rprtmmdd` ファイルを作成します。手動でこのコマンドを実行して、レポートを作成することもできます。

`prdaily` スクリプトは、次の 6 つの課金ファイルの情報を統合します。

- `/var/adm/acct/nite/reboots`
- `/var/adm/acct/nite/lineuse`
- `/var/adm/acct/sum/tacctmmdd`
- `/var/adm/acct/nite/daycms`
- `/var/adm/acct/nite/cms`
- `/var/adm/acct/sum/loginlog`

`prdaily` シェル・スクリプトの構文は次のとおりです。

prdaily [-l[*mmdd*]] | [-c]

このコマンドのオプションに関する詳細は、`prdaily(8)` を参照してください。

10.9.5 **monacct** シェル・スクリプト

`monacct` シェル・スクリプトは、バイナリ課金ファイルを使用して、累積要約ファイルを `/var/adm/acct/fiscal` ディレクトリに作成します。要約ファイルが作成された後、このコマンドによって古い課金ファイルが `/var/adm/acct/sum` ディレクトリから削除され、新しいファイルが作成されます。

通常、`monacct` スクリプトは、毎月 1 回実行されて、月次レポート・ファイルを作成します。`cron` デーモンを設定して、シェル・スクリプトを自動的に実行することもできます。詳細は、10.2.4 項を参照してください。

`monacct` シェル・スクリプトの構文は次のとおりです。

/usr/sbin/acct/monacct [*number*]

number 変数には、1 ~ 12 の範囲の整数を指定し、要約レポートを作成したい月を指定します。省略時の値は現在の月です。

monacct シェル・スクリプトは、次のファイルを /var/adm/acct/fiscal ディレクトリに作成します。

tacctmm *mm* 変数で指定されている月の前月のバイナリ総合課金ファイルを指定します。

cmsmm *mm* 変数で指定されている月の前月のバイナリ累積コマンド一覧ファイルを指定します。

fiscrptmm ASCII 形式の総合月次課金レポート・ファイルを指定します。このファイルは、レポート・ファイル /var/adm/acct/sum/rprtmmdd と同じ形式で、次のファイルから作成されます。

- /var/adm/acct/fiscal/tacctmm
- /var/adm/acct/fiscal/cmsmm
- /var/adm/acct/sum/loginlog



システムのモニタリングとテスト

この章では、以下の項目について説明します。

- 基本的なモニタリング・ガイドラインとユーティリティの概要、および関連するトピックの参照先 (11.1 節)
 - 一部のモニタリング・ユーティリティについての詳細 (11.2 節)
 - 温度や、冷却ファンが動作しているかどうかなど、システムのハードウェア・ステータスを監視する、環境モニタリング。この機能を使用するためには、このようなモニタリングをサポートするセンサをハードウェアが備えている必要があります。すべてのシステムがこの機能をサポートしているわけではありません (11.3 節)。
 - システム・コンポーネントであるテスト・ユーティリティの使用方法。システム・ハードウェアにもテスト・ルーチンが用意されています。詳細については、オーナーズ・マニュアルを参照してください (11.4 節)。
- システム・デバイス (ディスクやテープなど) の特性についての詳細は、『ハードウェア管理ガイド』の `hwmgr` コマンドを参照してください。

11.1 モニタリングとテストの概要

システムのモニタリングには、動作状況のパラメータ (CPU 負荷や I/O スループットなど) のベースラインを取得するための、基本的なコマンドおよびオプション・ユーティリティを使用します。稼働中のシステムの動作状況をモニタリング、記録、比較するためにこれらのベースラインを使用することで、システムが動作要件からかけ離れないようにできます。

システムのモニタリングを行うと、ユーザがシステムやその周辺機器を利用できなくなるような問題を事前に見つけ、防止することもできます。モニタリング・ユーティリティの情報により、システム・パニックやディスク・クラッシュなどの予期しないイベントに対し素早く対処し、問題を素早く解決してシステムをオンラインに戻すことができます。

モニタリングのトピックは、テクニカル・サポート上のニーズに深く関わっています。この章で説明するユーティリティには、二重の機能を持つものがあります。リアルタイムのシステム・モニタリングとは別に、これらのユーティリティはテクニカル・サポート担当者が使用する履歴データやイベント固有データも収集します。このデータは、オペレーティング・システムやハードウェアの障害後、迅速にシステムを立ち上げて稼働させるために必要になることがあります。少なくとも、11.1.1 項のモニタリング・ガイドラインに従ってください。

テストでは、システムの部品や周辺デバイス（ディスクなど）を試験するツールやユーティリティを使用します。利用可能なテスト・ユーティリティについては、この章で説明します。システム・ハードウェアにも、コンソール・プロンプトで実行できるテスト・ユーティリティが用意されています。ハードウェア・テスト・コマンドについては、オーナーズ・ガイドを参照してください。

11.1.1 項では、システムのモニタリングの一般的なガイドラインについて説明します。11.1.2 項では、オペレーティング・システムに用意されているすべてのユーティリティについて概要を説明します。

11.1.1 システムのモニタリングの概要

目的の動作に合わせてシステムを構成した後で、次の手順を実行してください。

1. 毎日使用するモニタリング・ユーティリティを選択する。

11.1.2 項にある、モニタリング・ユーティリティの概要を参照してください。システム構成に基づいて、構成およびモニタリングの要件に合うユーティリティを選択します。たとえば、グラフィック・ターミナルがあり、分散システムのモニタリングを行う場合は、SysMan Station をセットアップすることを考慮してください。単一のローカル・サーバのモニタリングを行う場合は、dxsysinfo ウィンドウで十分です。

必要に応じて、警告やメッセージを表示するトリガーとなる属性を設定します。たとえば、ファイル・システムが満杯になってデータが紛失するのを防ぐには、85% で満杯になるような制限をすべてのファイル・システムに設定できます。

注意

多くのオプション・サブシステムには、独自のモニタリング・ユーティリティが用意されています。これらのインタフェースを十分に理解し、一般のユーティリティよりも適しているかどうかを判断してください。

2. ベースラインを確定する。

`sys_check` ユーティリティを `-all` オプションを指定して実行し、次の処理を行います。

- 負荷のない状態でのベースラインを確定する。
- チューニングが必要なシステム属性がないか調べる。

必要であれば、`sys_check` ユーティリティの情報を使用して、システムの属性をチューニングします。システムのチューニングについては、『システムの構成とチューニング』を参照してください。このベースラインのデータを、別のシステム上など、後で簡単にアクセスできる場所に保管します。また、このレポートのコピーを印刷することもできます。

3. 負荷のある状態で `sys_check` ユーティリティを実行する。

システムに適度な負荷がかかっている状態で、`sys_check` ユーティリティを実行します。モニタリングしたいオプション (`-perf` など) だけを選択します。この操作は、システムの性能に多少影響を与えることがあるため、エンドユーザの使用がピークになる時間帯を避けて実行してください。

`sys_check` ユーティリティの出力を分析し、動作要件に合わせるために変更が必要であれば追加で変更します。この操作では、システム属性のチューニングがさらに必要となる場合や、クラス・スケジューラなどのユーティリティによるシステム・リソースの再配置などの構成の変更が必要となる場合もあります。`sys_check` ユーティリティの使用方法については、11.2.2 項を参照してください。

4. Event Managerをセットアップする。

モニタリング方式を選択すると同時に、システムのイベント管理ログおよびレポート方式を構成します。Event Manager の構成方法については、第 13 章および第 12 章を参照してください。

5. モニタリング・ユーティリティを構成する。

使用するモニタリング・ユーティリティが他にあれば、セットアップします。たとえば、次の処理を行います。

- 11.2.2 項で説明しているように、`cron` ユーティリティを `runsyscheck` スクリプトとともに使用して、`sys_check` ユーティリティがオフ・ピーク時に定期的に行われるように構成します。定期的に更新されるレポートは、システムの障害時に、問題の分析およびトラブルシューティングに役立ちます。

注意

システムの問題の診断には、クラッシュ・ダンプ・データが必要になる場合もあります。クラッシュ・ダンプ環境の構成については、第 14 章を参照してください。

- オプションの性能ユーティリティをインストールして構成します。ターゲット・システムでサポートされている場合は、11.3 節で説明しているように、環境モニタリングも構成してください。

11.1.2 コマンドおよびユーティリティの概要

オペレーティング・システムには、多数のモニタリング・コマンドおよびユーティリティが用意されています。システム・データの単純なスナップショットを数値形式で返却するコマンドもあれば、情報を選択したりフィルタリングするオプションを多数持つコマンドもあります。また、リアル・タイムでシステム・データをフィルタリングして追跡し、そのデータをグラフィック・ターミナルに表示する、複雑なグラフィカル・ユーザ・インタフェースもあります。

ローカルな環境やモニタリングの要件に最適なモニタリング・ユーティリティを選択してください。また、次の事項を考慮してください。

- モニタリング・ユーティリティを使用すると、システムの性能に影響を与えることがあります。
 - 入出力のボトルネックなどの性能上の問題の診断に利用する場合は、`iostat` などの簡単なコマンドが適しています。
 - シングルユーザ・システム上のリソースを素早く視覚的に検査する場合は、X11 システム情報インタフェース (`dxsysinfo`) が適しています。

- root ユーザ専用のユーティリティと、すべてのシステム・ユーザがアクセスできるユーティリティがあります。
- エンタプライズ・レベルのモニタリングでは、SysMan Station を使用すると、多数のシステムの状態を同時に 1 つの画面に表示できます。
- エンタプライズ・レベルで資産を追跡管理したり、どのオプションがどのシステムにインストールされているかを確認する（およびそのオプションが正しく動作するかどうかを確認する）には、UNIX サーバおよびクライアント PC システムの両方で、Web ベースの HP Insight Manager ユーティリティを使用できます。
- 問題の診断時に、モニタリング・ユーティリティからの出力をテクニカル・サポート・サイトに渡さなければならないことがあります。問題が発生する前に、システムのベースラインを調査し、定期的なモニタリングとデータ収集のスケジュールを決めておくと、システムのダウン時間を大幅に短縮できます。

以降の項では、モニタリング・ユーティリティについて説明します。

11.1.2.1 コマンド行ユーティリティ

各種のシステム統計情報のスナップショットを表示するには、以下のコマンドを使用します。

vmstat

vmstat コマンドは、仮想メモリ、プロセス、トラップ、および CPU 動作のシステム統計情報を表示します。vmstat の出力例を次に示します。

```
bigrig> vmstat
Virtual Memory Statistics: (pagesize = 8192)
procs  memory      pages                      intr      cpu
r  w  u act  free wire fault cow zero react pin pout in  sy  cs us sy id
2  97 20 8821  50K 4434 653K 231K 166K 1149 142K    0 76 250 194  1  1 98
```

詳細については、vmstat(1) を参照してください。

iostat

iostat コマンドは、端末およびディスクの入出力情報と、各種の動作の実行に使用された CPU 時間の割合を報告します。iostat の出力例を次に示します。

```
bigrig> iostat
          tty          floppy0          dsk0          cpu
tin tout    bps    tps    bps    tps  us ni sy id
0      1      0      0      3      0  0  0  0  1 98
```

詳細については、`iostat(1)` を参照してください。

`who`

`who` コマンドは、ローカル・システムのユーザとプロセスの情報を報告します。`who` の出力例を次に示します。

```
bigrig>who
# who
root      console   Jan  3 09:55
root      :0          Jan  3 09:55
root      pts/1       Jan  3 09:55
bender    pts/2       Jan  3 14:59
root      pts/3       Jan  3 15:43
```

類似のコマンド `users` があります。これは、ログインしているユーザについて、簡潔なリストを表示します。

詳細については、`who(1)` および `users(1)` を参照してください。

`uptime`

`uptime` コマンドは、システムが稼働した時間を報告します。

```
bigrig> uptime
16:20 up 167 days, 14:33,  4 users,  load average: 0.23, 0.24, 0.24
```

詳細は、`uptime(1)` を参照してください。

`uptime` と同じ情報を表示する、類似のコマンド `w` がありますが、`w` はログインしているユーザの情報も表示します。詳細は、`w(1)` を参照してください。

`netstat`

`netstat` コマンドは、ネットワーク関連の統計情報を、各種のフォーマットで表示します。

ネットワークのモニタリングの詳細は、`netstat` コマンドと『ネットワーク管理ガイド：接続編』を参照してください。

11.1.2.2 SysMan Menu のモニタリング作業およびチューニング作業

SysMan Menu には、いくつかのモニタリング作業のオプションがあります。SysMan Menu の使用方法についての一般的な説明は、第 1 章を参照してください。[モニタリング/チューニング] メニュー項目下には、以下のオプションがあります。

イベントの参照 [event_viewer]

このオプションは、EVM イベント・ビューア (第 13 章を参照) を起動します。

HP Insight Manager の設定 [imconfig]

HP Insight Manager を構成して HP Insight Manager デーモンを開始するインタフェースを起動します。HP Insight Manager の構成については、第 1 章を参照してください。

仮想メモリ (VM) 統計の参照 [vmstat]

11.1.2.1 項で説明した `vmstat` コマンドへの SysMan Menu インタフェースです。

入出力 (I/O) 統計の参照 [iostat]

11.1.2.1 項で説明した `iostat` コマンドへの SysMan Menu インタフェースです。

UPTIME 統計の参照 [uptime]

11.1.2.1 項で説明した `uptime` コマンドへの SysMan Menu インタフェースです。

この他、[サポートとサービス] メニュー項目の下には以下のオプションがあります。

エスカレーション・レポートの作成 [escalation]

このオプションによって、`sys_check` ユーティリティのエスカレーション・レポート機能を起動します。エスカレーション・レポートは診断サービスの際にのみ使用され、テクニカル・サポート部門がこのレポートを必要とします。`sys_check` ユーティリティのエスカレーション・オプションの使用方法についての詳細は、11.2.2 項を参照してください。

構成レポートの作成 [config_report]

このオプションによって、`sys_check` ユーティリティのシステム構成レポート機能を起動します。このオプションを使用すると、システム構成のベースラインを記録し、定期的にベースラインを更新できます。

このオプションを使用すると、省略時の完全なレポートが作成されますが、完了までに長い時間がかかり、システムの性能に影響を与える可能性があります。 `sys_check` ユーティリティの使用方法についての詳細は、11.2.2 項を参照してください。

11.1.2.3 SysMan Station

SysMan Stationは1つ以上のシステムをグラフィカルに表示でき、どのシステム上でも管理作業を行うためのアプリケーションを起動することができます。SysMan Stationの使用方法については、第1章を参照してください。

11.1.2.4 X11 準拠のグラフィカル・ユーザ・インタフェース

オペレーティング・システムには、省略時の共通デスクトップ環境 (CDE) ウィンドウ環境下で一般的に使用されるいくつかのグラフィカル・ユーザ・インタフェース (GUI) があります。これは「システム管理」フォルダにあります。

これらのインタフェースは、CDE フロント・パネルの「SysMan アプリケーション」パネルから起動できます。「SysMan アプリケーション」パネルの詳細は、図 1-5 を参照してください。ここには、「モニタリング/チューニング」フォルダ、「ツール」フォルダ、および「日常管理」フォルダを指すアイコンがあります。

「モニタリング/チューニング」フォルダ

このフォルダには、以下の SysMan Menu の項目を起動するアイコンがあります。

構成レポート

このアイコンは、`sys_check` ユーティリティのシステム構成レポート機能のグラフィカル・ユーザ・インタフェースを起動します。

エスカレーション・レポート

このアイコンは、`sys_check` ユーティリティのエスカレーション・レポート機能のグラフィカル・ユーザ・インタフェースを起動します。

HP Insight Manager

このアイコンは、HP Insight Manager を構成して HP Insight Manager デーモンを開始するインタフェースを起動します。

このフォルダ内の残りのアプリケーションは、システムのチューニングに関連するアプリケーションです。プロセス・チューナ (nice コマンドへのグラフィカル・ユーザ・インタフェース) およびカーネル・チューナ (dxkerneltuner) を使用するチューニングについては、『システムの構成とチューニング』を参照してください。

「ツール」フォルダ

「ツール」フォルダには、`vmstat` などのコマンドへのグラフィカル・ユーザ・インタフェースが用意されています。これらのインタフェースは、CDE フロント・パネルから「アプリケーション・マネージャ」アイコンを選択して「アプリケーション・マネージャ」フォルダから、「システム管理」アイコン、「ツール」アイコンの順で選択します。このフォルダには、以下のインタフェースが用意されています。

「I/O 統計」

`iostat` コマンド (11.1.2.1 項参照) へのグラフィカル・ユーザ・インタフェースです。

「ネットワーク統計」

`netstat` コマンドへのグラフィカル・ユーザ・インタフェースです。ネットワークのモニタリングについては、『ネットワーク管理ガイド：接続編』を参照してください。

「システム・メッセージ」

`/var/adm/messages` ログ・ファイルへのグラフィカル・ユーザ・インタフェースです。このファイルには、システム・イベント管理の現在の設定内容に応じて特定のシステム・メッセージが格納されます。イベント、イベントが生成するメッセージ、メッセージ・ログ・ファイルについては、第 12 章および第 13 章を参照してください。

「仮想メモリ統計」

`vmstat` コマンド (11.1.2.1 項参照) へのグラフィカル・ユーザ・インタフェースです。

Who?

`who` コマンド (11.1.2.1 項参照) へのグラフィカル・ユーザ・インタフェースです。

「日常管理」フォルダ

この他の X11 準拠モニタリング・アプリケーションは、「システム情報」です。これは「アプリケーション・マネージャー-日常管理」フォルダにあります。

「システム情報」

このインタフェースを起動するには、「システム情報」(dxsysinfo) アイコンをクリックします。このインタフェースを使用すると、以下のシステム・リソースおよびデータを素早く参照することができます。

- プロセッサ (CPU) の数とタイプの簡単な説明
- UNIX オペレーティング・システムのバージョンと、利用可能なシステム・メモリの量
- CPU 動作、使用中のメモリ、および使用中の仮想メモリ (スワップ) の量の概算を示す、3つの表示。この情報は、`vmstat` などのコマンドを使用しても取得できます。
- ファイルおよびスワップ用の、2つの警告インジケータ。いずれかのファイル・システムがほぼ満杯になるか、スワップ領域の量が少なすぎる場合は、これらのインジケータの色が変わります。
- すべてのローカルおよびリモート・マウントのファイル・システムの、現在利用可能なスペースの状況。ここでは、利用可能なスペースが一定の割合を下回った場合に警告インジケータを表示するように、限界値を割合で設定できます。利用可能なファイル・システム・スペースを増やす方法については、第 6 章および第 9 章を参照してください。

11.1.2.5 高度なモニタリング・ユーティリティ

以下のユーティリティには、多数の動作パラメータを参照し記録するオプションがあります。

`collect`

`collect` ユーティリティを使用すると、事前に決めたサンプリング期間の間、多くの種類のシステム・データやプロセス・データを同時にサンプリングできます。情報をデータ・ファイルに収集し、端末でそのファイルを再生することができます。

`collect` ユーティリティを使用すると、性能上の問題の診断に役立ちます。また、テクニカル・サポート・サービスがシステムの問題解決の手助けをする際に、このユーティリティの出力結果を必要とすることがあります。`collect` ユーティリティの使用方法については、11.2.1 項を参照してください。

詳細は、インストレーション・キットにある『*Collect User's Guide*』を参照してください。

`sys_check` ユーティリティ

`sys_check` ユーティリティは、システム構成および多数のシステム属性の現在の設定を恒久的に記録するために使用する、コマンド行インタフェースです。このユーティリティの詳細は、11.2.2 項で説明しています。

MPH (Monitoring Performance History) ユーティリティ

MPH (Monitoring Performance History) ユーティリティは、オペレーティング・システムやハードウェア環境の信頼性や可用性の情報 (クラッシュ・データ・ファイルなど) を収集するシェル・スクリプト群です。このユーティリティの詳細は、11.2.3 項で説明しています。

11.1.3 関連ドキュメント

本書の以下のトピックは、システムのモニタリングおよびテストに密接に関連しています。

- プリンタなどのリソースへのアクセスをモニタリングして記録することが可能なシステム課金サービスの管理については、第 10 章を参照してください。
- 基本的な `binlogd` および `syslogd` イベント・チャネルを使用した基本システム・イベント・ロギングの構成および使用の手順については、第 12 章を参照してください。この章では、イベントやエラーが記録されているシステム・ログ・ファイルへのアクセス方法も説明しています。
- 特定のタイプのイベントへの自動応答など、高度なシステム・イベント管理が可能なイベント・マネージャ (EVM) の構成および使用については、第 13 章を参照してください。

次のマニュアルにも、システム・モニタリングとテストについての追加情報が含まれています。

- システムのモニタリングおよびテストで収集した情報に従って、システムをチューニングする方法については、『システムの構成とチューニング』を参照してください。
- システムのネットワーク構成要素のモニタリングについての詳細は、『ネットワーク管理ガイド：接続編』を参照してください。

11.2 モニタリング・ユーティリティの構成および使用

以降の項では、いくつかのモニタリング・ユーティリティを紹介し、セットアップ方法と使用方法について説明します。詳細については、各アプリケーションに付属のドキュメントおよびリファレンス・ページを参照してください。グラフィック環境を備えたシステムのモニタリング用に SysMan Station を構成して使用方法については、第 1 章を参照してください。

イベント管理とエラー・ロギングは、密接に関連するトピックです。これらのトピックについては、第 12 章および第 13 章を参照してください。

11.2.1 collect ユーティリティによるシステム・データの記録

`/usr/sbin/collect` コマンド行ユーティリティは、現在のシステム状態を示すデータを収集します。このユーティリティを使用すると、多数のパラメータから選択してソートすることや、データ収集期間を指定することができます。データはリアルタイムに表示できますが、後で分析したり再生するためにファイルに記録することもできます。`collect` ユーティリティを使用したために発生する CPU のオーバヘッドは少しだけです。これは、記録するシステム動作だけに焦点を置くことができるので、システムの性能に悪影響を与えないためです。

何も指定しないで `/usr/sbin/collect` コマンドを実行すると、出力は `vmstat`、`iostat`、または `netstat` などのモニタリング・コマンドを実行した場合と同じようになります。

コマンド構文の詳細は、`collect(8)` に定義されています。`collect` ユーティリティの重要な機能は、以下のとおりです。

- データのサンプルを取得する期間と頻度を制御します。プロセッサの使用量に従って、出力をソートします。

- データ・レコード・ファイルから、データのタイム・スライスを抽出します。たとえば、使用量が最も多い時間帯でのシステム・パラメータをいくつかを参照したい場合、`-c` オプションを使用すると、データ・ファイルからデータを抽出できます。
- デバイス特殊ファイル名を使用して、特定のデバイスを指定します。たとえば、次のコマンドでは、指定したデバイスからデータを収集するように指示しています。

```
# collect -sd -Ddsk1,dsk10
```

- CPU やネットワークなど特定のサブシステムを指定します。たとえば、次のコマンドでは、CPU のデータだけを収集することを指定しています。また、データのサンプルをコマンドの下に示します。

```
# collect -e cf
CPU SUMMARY
USER SYS IDLE WAIT INTR SYSC CS RUNQ AVG5 AVG30 AVG60 FORK VFORK
  13  16   71    0 149  492 725   0 0.13 0.05  0.01 0.30 0.00
SINGLE CPU STATISTICS
CPU USER  SYS IDLE WAIT
   0   13   16   71    0
```

- `-H` (履歴) オプションを使用して、一連のデータ・ファイルを記録し保存します。
- 記憶域を節約するために、データ・ファイルは圧縮されます。
- データのサンプルを取得するユーザ、グループ、およびプロセスを指定します。
- `-p` オプションを使用して、データ・ファイルを再生します。`-f` オプションを使用すると、複数のバイナリ入力ファイルを、1 つのバイナリ出力ファイルにまとめることができます。

`collect` ユーティリティはページ・ロック関数 `plock()` を使用して、自分自身をメモリ内にロックします。このためシステムは、このユーティリティをスワップ・アウトすることはできません。またこのユーティリティは、優先順位関数 `nice()` を使用して、自身の優先順位を上げます。必要であれば、`-ol` コマンド・オプションを使用してロックを無効にでき、`-on` コマンド・オプションを使用して優先順位の設定を無効にできます。ただし、負荷の高いシステムで `collect` ユーティリティを使用しても最小限の影響しかありません。

11.2.2 sys_check ユーティリティの使用

sys_check ユーティリティを使用すると、次の処理ができます。

- ソフトウェアとハードウェアの両方のシステム構成情報のベースラインを確定し、Web ブラウザで簡単にアクセスできる HTML 形式のレポートとして記録することができます。このレポートは定期的に再作成するか、システム構成が変化したときに再作成することができます。
- 多数のシステム属性 (チューニング・パラメータなど) の自動検査を実行することができます。現在のシステムの使用状況にもっと適した設定についてのフィードバックを受け取ることができます。

sys_check は、推奨される保守作業 (パッチ・キットのインストールやスワップ領域の保守など) についても、検査しレポートします。

- テクニカル・サポート・サービスがシステムの問題の診断や修正に使用する、問題のエスカレーション・レポートを作成できます。

現在のハードウェアおよびソフトウェアの構成を記録する他に、sys_check ユーティリティはシステム性能状況のパラメータの拡張ダンプを作成できます。この機能を使用すると、多数のシステム属性値を記録でき、システム・データの有用なベースラインを得ることができます。このようなベースラインは、大規模な変更や、トラブルシューティングを行う際には、特に役立ちます。

sys_check ユーティリティを実行すると、HTML ドキュメントが標準出力に出力されます。-escalate フラグを指定してこのスクリプトを実行すると、特に指定しなければ \$TMPDIR/escalate* 出力ファイルが作成されます。環境変数 \$TMPDIR によって、現在の一時ディレクトリが決まります。これらのファイルをテクニカル・サポート部門に転送し、システムの問題やエラーの診断に使用することができます。

コマンド・オプションの完全なリストを表示するには、次のコマンドを実行します。

```
# /usr/sbin/sys_check -help
```

sys_check ユーティリティが生成する出力は、通常 0.5 MB ~ 3 MB のサイズで、検査の完了までに 30 分 ~ 1 時間かかります。各種のコマンド・オプションの詳細は、sys_check(8) を参照してください。いくつかの項目を除外して実行すると、実行時間を大幅に短縮できます。たとえば、sys_check ユーティリティはインストールされているソフトウェアを記録す

るために `setld` を実行します。 `setld` 動作を除外すると、 `sys_check` の実行期間を大幅に短縮できます。

標準の `sys_check` 実行タスクは、次の方法で起動することができます。

- CDE を使用して、CDE フロント・パネルから「アプリケーション・マネージャ」をオープンします。「システム管理」を選択してから、「モニタリング/チューニング」を選択します。標準の `sys_check` 実行タスクのアイコンが 2 つ（「構成レポート」と「エスカレーション・レポート」）表示されます。
- SysMan Menu を使用して、[サポートとサービス] メニュー項目を展開し、次のオプションから選択します。
 - エスカレーション・レポートの作成
 - 構成レポートの作成

SysMan Menu の使用方法については、第 1 章を参照してください。

`root` の `crontabs` ファイルでオプションを有効にすることで、 `sys_check` タスクを自動的に実行することができます。 `/var/spool/cron/crontabs` ディレクトリの `root` ファイルには、 `cron` が定期的に起動する省略時のタスクのリストが入っています。次の行からコメント（#）コマンドを削除してください。

```
#0 3 * * 0 /usr/share/sysman/bin/runsyscheck
```

このオプションを有効にすると、結果のレポートを HP Insight Manager が参照します。そのため、HP Insight Manager のデバイス・ホーム・ページの「Tru64 UNIX 構成レポート」アイコンを選択して、参照することができます。HP Insight Manager の詳細は、第 1 章を参照してください。

11.2.3 Monitoring Performance History ユーティリティの使用

MPH (Monitoring Performance History) ユーティリティは、オペレーティング・システムとそのハードウェア環境の信頼性や可用性の情報（クラッシュ・データ・ファイルなど）を収集するシェル・スクリプト群です。この情報は、インターネット・メールまたは DSN リンク（利用可能な場合）によって、システムのベンダに自動的にコピーされます。このデータを使用して性能分析レポートが作成され、開発グループおよびサポート・グループに配布されます。この情報は、信頼性の高い高可用性のシステムの設計を改善するために、システム・ベンダの内部でのみ使用されます。

MPH の処理は自動的に実行されるため、ユーザの操作は不要です。初回の構成に、約 10 分かかります。MPH は、システムの性能に影響を与えたり、低下させることはありません。これは、MPH がわずかな CPU リソースを使用して、バックグラウンド・タスクとして実行されるためです。収集データおよびアプリケーションに必要なディスク・スペースは、1 システムあたり約 300 ブロックです。この値は、エラーの数が多いと若干大きくなることがあります。またこの値は、初回の実行でベースラインを確立する際には、かなり大きくなります (ただし、これは 1 回だけです)。

MPH ユーティリティは、次のように動作します。

- 10 分ごとに、システムが動作中であることを示すタイムスタンプを記録します。
- 毎日午前 2:00 に、省略時のイベント・ログ `/var/adm/binary.errlog` から新たに追加されたイベント・レコードを抽出します。
- 毎日午前 3:00 に、イベントおよびタイムスタンプのデータ、`/var/adm/crash` 内の新しい `crashdc` データ・ファイルを、システム・ベンダに転送します。平均の転送量は、データ 150 ブロックです。

MPH を実行する前に、次の情報について確認してください。

- Standard Programmer Commands (Software Development) の OSFPGMR400 サブセットがインストールされていなければなりません。 `setld -i` コマンドを使用すると、このサブセットがインストールされているか確認できます。
- MPH ソフトウェア・キットは、必須のベース・ソフトウェア・サブセット OSFHWBASE400 に含まれています。このサブセットは、オペレーティング・システムのインストール時に自動的にインストールされます。完全なドキュメントは、`/usr/field/mph/unix_installation_guide.ps` にあります。テキスト・ファイルのドキュメントも利用できます。
- MPH ソフトウェア・サブセットで必要となるディスク・スペースは、約 100 ブロックです。

MPH をシステムに構成する人は、root ユーザでなければならず、ターゲット・システムの管理者でなければなりません。MHP を構成する際

には、名前、電話番号、e-mail アドレスを入力する必要があります。次の手順で実行します。

1. ターゲット・システムのシリアル番号 (SN) を調べます。この番号は通常、システム本体の背面に記録されています。インストール・スクリプトを完了させるには、この番号が必要です。

2. 次のコマンドを入力して、MPH スクリプトを実行します。

```
# /usr/field/mph/MPH_UNIX***.CSH
```

ここで、*** はバージョン番号 (025 など) です。

3. スクリプトが要求する情報を入力します。スクリプトの実行が完了したら、MPH が自動的に開始されます。

何らかの理由でオペレーティング・システムをシャットダウンする必要がある場合は、正常なシャットダウン処理に従わなければなりません。従わなかった場合は、MPH のドキュメントで説明されているように、MPH スクリプトを再起動しなければなりません。詳細については、mph(1) を参照してください。

11.3 環境モニタリングと envmond/envconfig

どのようなシステムでも、換気の悪さ、過熱状態、あるいはファンの故障のために温度が上昇することがあります。このような状態が検出されなければ、予期しないシャットダウンのためにシステムのデータが失われたり、システム自体が損傷する恐れがあります。環境モニタリング機能を使用すると、前もって警告されるため、システムを回復したり、通常の段階的なシャットダウンを実行することができます。

コマンド行からの環境モニタリング

次のように sysconfig コマンド行ユーティリティを使用して、システム環境の状態のモニタリングを行えます。

```
# /sbin/sysconfig -q envmon
```

このコマンドを使用すると、現在の温度、ファンの状態、その他の一般的な環境情報がレポートされます。

環境センサのモニタリング

少数の最新のハードウェア・プラットフォームには、プロセッサの温度やファンの回転速度しきい値など、温度環境をモニタリングする機能が備わっています。root で次のコマンドを実行して、ご使用中のハードウェア・プラットフォームにこの機能が備わっているかどうかを調べてください。

```
# /sbin/hwmon view hier | grep sensor
52:      sensor systhermal-cpu0
53:      sensor systhermal-cpu1
54:      sensor systhermal-cpu2
55:      sensor systhermal-pci_zone-1
56:      sensor systhermal-pci_zone-2
57:      sensor systhermal-pci_zone-3
58:      sensor sysfan-pci_zone-1/2
59:      sensor sysfan-power_supply-3/4
60:      sensor sysfan-cpu_memory-5/6
61:      sensor syspower-ps-0
62:      sensor syspower-ps-1
63:      sensor syspower-ps-2
```

このコマンドの出力には、個々のアクティブなセンサが表示されます。

出力がなかった場合、これらの機能はご使用中のプラットフォームには備わっていません。

これらのセンサは、しきい値に達した場合、Event Manager にイベントをポストします。Event Manager ログ・ファイルを参照して Hardware Manager コマンド行プログラムを使用すると、どの構成要素がクリティカルになっているかがわかります。Event Manager の詳細は、第 13 章を参照してください。Hardware Manager コマンド行ユーティリティの詳細は、『ハードウェア管理ガイド』と hwmon(8) を参照してください。

HP Insight Manager

HP Insight Manager でも「回復->環境」画面によって、環境条件のモニタリングが行えます。詳細は、『*HP Management Agents for AlphaServers for Tru64 UNIX*』を参照してください。

envmond/envconfig フレームワーク

環境モニタリングのフレームワークは、4 つのコンポーネントから構成されています。

- ロード可能カーネル・モジュールおよびその関連 API
- Server System MIB サブエージェント・デーモン

- `envmond` デーモン

`envmond` デーモンはシステム環境のモニタリングのために使用します。詳細は、`envmond(8)` を参照してください。

- `envconfig` ユーティリティ

`envconfig` ユーティリティを使用すると、`envmond` デーモンをカスタマイズできます。詳細は、`envconfig(8)` を参照してください。

これらのコンポーネントについて、以下の各項で説明します。

11.3.1 ロード可能カーネル・モジュール

ロード可能カーネル・モジュールおよびその関連 API には、システムのしきい値レベルをモニタして状態を返すために必要なパラメータが含まれています。カーネル・モジュールは、11.3.1.1 項で説明するように、カーネル構成マネージャ (CFG) インタフェースを介してのみサーバ管理属性をエクスポートします。これは、サーバ管理をサポートするすべてのプラットフォームで動作するとともに、開発中の他のサーバ管理システムとの互換性を提供します。

ロード可能カーネル・モジュールには、プラットフォーム固有のコード (状態レジスタの位置など) は含まれていません。カーネル・モジュールは、どのオプションがプラットフォームでサポートされているのかを意識する必要がありません。つまり、カーネル・モジュールおよびプラットフォームは、オプションがサポートされている場合には有効なデータを返し、サポートされないオプションに対しては一定の値またはヌルを返すように設計されています。

11.3.1.1 ロード可能カーネル属性の指定

ロード可能カーネル・モジュールは、表 11-1 に示すパラメータをカーネル構成マネージャ (CFG) にエクスポートします。

表 11-1: カーネル・モジュールで定義されているパラメータ

| パラメータ | 用途 |
|----------------------|--|
| env_current_temp | システムの現在の温度を指定します。システムに KCRCM モジュールを構成している場合、返される温度は摂氏です。システムが温度の読み取りをサポートしておらず、温度がしきい値を超えていない場合は、-1 が返されます。システムが温度の読み取りをサポートしておらず、温度がしきい値を超えている場合には、-2 が返されます。 |
| env_high_temp_thresh | システム固有の動作温度のしきい値を指定します。返される値は、ハードコードされたプラットフォーム固有の温度です。単位は摂氏です。 |
| env_fan_status | 危険ではないファンの状態を指定します。返される値はゼロ (0) のビット値です。この機能に対してハードウェア・サポートがある場合には、この値は異なります。 |
| env_ps_status | 冗長電源の状態を指定します。冗長電源の故障のために割り込みを行うプラットフォームでは、対応するエラー状態ビットを読み込んでリターン値を決定します。エラーの場合には値 1 が返され、それ以外の場合にはゼロ (0) が返されます。 |
| env_supported | プラットフォームがサーバ管理および環境モニタリングをサポートしているかどうかを示します。 |

11.3.1.2 プラットフォーム固有の機能の取得

ロード可能カーネル・モジュールは、照会されているプラットフォームに基づいて環境状態を返さなければなりません。環境状態を取得するには、`get_info()` 関数を使用します。`get_info()` 関数の呼び出しは、`platform_callsw[]` テーブルを通してフィルタにかけられます。

`get_info()` 関数は、表 11-2 に示す関数タイプを使用して、動的な環境データを取得します。

表 11-2: `get_info()` 関数タイプ

| 関数タイプ | 関数の使用 |
|----------------|--|
| GET_SYS_TEMP | KCRCM モジュールが構成されているプラットフォーム上のシステムの内部温度を読み取ります。 |
| GET_FAN_STATUS | エラー・レジスタからファンの状態を読み取ります。 |
| GET_PS_STATUS | エラー・レジスタから冗長電源の状態を読み取ります。 |

get_info() 関数は、HIGH_TEMP_THRESH 関数タイプを使用して静的なデータを取得します。この関数タイプは、プラットフォーム固有の動作温度の上限しきい値を読み取ります。

11.3.2 Server System MIB サブエージェント

Server System MIB エージェント (eSNMP サブエージェント) は、Server System MIB で指定された環境モニタリング・パラメータのサブセットをエクスポートするために使用されます。Server System MIB は、インストールされているオペレーティング・システムに応じて、すべてのサーバ・プラットフォームに共通するハードウェア固有パラメータ・セットをエクスポートします。

表 11-3 は、環境モニタリングをサポートする Server System MIB 変数サブセットのカーネル・パラメータ (11.3.1.1 項で説明) へのマッピングを示しています。

表 11-3: サーバ・サブシステム変数のマッピング

| Server System MIB 変数名 | カーネル・モジュール・パラメータ |
|-----------------------|----------------------|
| svrThSensorReading | env_current_temp |
| svrThSensorStatus | env_current_temp |
| svrThSensorHighThresh | env_high_temp_thresh |
| svrPowerSupplyStatus | env_ps_temp |
| svrFanStatus | env_fan_status |

MIB 記述をコンパイルして骨組みサブエージェント・デーモン用のコードにするには、SNMP MIB コンパイラおよび他のユーティリティを使用します。サブエージェント・デーモンとマスタ・エージェント eSNMP デーモン (snmpd) の間の通信は、eSNMP シェアード・ライブラリ (libesnmp.so) 内のインタフェースで処理されます。サブエージェント・デーモンは、システムをブートして eSNMP デーモンが起動された後に起動する必要があります。

サブエージェント・デーモンには、表 11-3 に示した Server System MIB の各変数に対する処理が含まれています。このデーモンは、CFG インタフェースを介してカーネル・モジュールから適切なパラメータにアクセスします。

11.3.3 環境モニタリング・デーモン

環境モニタリング・デーモン `envmond` を使用することにより、しきい値レベルをチェックして、システムに障害が発生する前に修復作業を行うことができます。 `envmond` デーモンは、次の処理を行います。

- システムにしきい値レベルを照会します。
- 冷却ファンが故障したときに、システムのシャットダウンを開始します。

AlphaServer 1000A の冷却ファンが故障すると、カーネルはエラーのログを取り、ディスクの同期を取ったのち、システムの電源をオフにします。

その他のファンが故障した場合には、ハード・シャットダウンが行われます。

- 高温のしきい値状態が解決すると、ユーザに通知します。
- 回復が不可能な場合には、通常のシャットダウンを行っていることをすべてのユーザに通知します。

システムに照会するため、`envmond` デーモンはベース・オペレーティング・システムの `/usr/sbin/snmp_request` コマンドを使用して、Server System MIB に指定されている環境変数の現在の値を取得します。

環境モニタリングを有効にするためには、`envmond` デーモンをシステム・ブート中に起動する必要がありますが、eSNMP および Server System MIB エージェントが起動された後でなければなりません。

詳細は、`envmond(8)` を参照してください。

11.3.4 `envconfig` を使用した `envmond` デーモンの構成

`envconfig` ユーティリティを使用して、`envmond` デーモンが環境を照会する方法をカスタマイズすることができます。これらのカスタマイズは、`/etc/rc.config` ファイルに格納されます。このファイルは、スタートアップ時に `envmond` デーモンによって読み込まれます。

`envconfig` ユーティリティを使用すると、次の処理を行うことができます。

- システム・ブート中に環境モニタリングを有効または無効に設定します。
- システム・ブート後に `envmond` デーモンを起動または停止させます。

- envmond デーモンによるシステム照会の頻度を指定します。
- envmond デーモンによって温度イベントが通知される前に検出される最も高いしきい値レベルを設定します。高いしきい値レベルが検出された場合に envmond デーモンに実行させるユーザ定義スクリプトのパスを指定します。
- シャットダウン・メッセージがブロードキャストされた場合にユーザがデータを保存するための、シャットダウンまでの猶予期間を指定します。
- 環境モニタリング変数の値を表示します。

詳細は、envconfig(8) を参照してください。

11.3.5 ユーザ定義可能なメッセージ

環境モニタリング・ユーティリティでブロードキャストされたりログに記録されるメッセージは、すべて変更できます。メッセージは、次のファイル内にあります。

```
/usr/share/sysman/envmon/EnvMon_UserDefinable_Msg.tcl
```

このファイルを編集するには、root でなければなりません。中カッコ ({}) 中のメッセージは、すべて編集することができます。このファイルの各セクションの編集方法については、# で始まるコメント・フィールドに説明があります。

たとえば、次のメッセージは、高温状態の原因を示すサンプルです。

```
set EnvMon_Ovstr(ENVMON_SHUTDOWN_1_MSG){System has reached a \
high temperature condition. Possible problem source: Clogged \
air filter or high ambient room temperature.}
```

このメッセージ・テキストを、次のように変更できます。

```
set EnvMon_Ovstr(ENVMON_SHUTDOWN_1_MSG) {System \
has reached a high temperature condition. Check the air \
conditioning unit}
```

中カッコ ({}) 内のテキスト文字列以外は、このファイル内のデータを変更しないでください。

11.4 システム・エクササイズの使用

オペレーティング・システムには、システムのトラブルシューティングに使用できるシステム・エクササイズがあります。エクササイズは、ファイル・

システムやシステム・メモリなど、システムの特定領域をテストします。以下の節で、システム・エクササイズについて説明します。

- システム・エクササイズの実行 (11.4.1 項)
- エクササイズ診断の使用 (11.4.2 項)
- `fsx` コマンドを使用した、ファイル・システムのエクササイズ (11.4.3 項)
- `memx` コマンドを使用した、システム・メモリのエクササイズ (11.4.4 項)
- `shmx` コマンドを使用した、共用メモリのエクササイズ (11.4.5 項)
- `cmx` コマンドを使用した、通信システムのエクササイズ (11.4.6 項)

また、`diskx` コマンドを使用してディスク・ドライブのエクササイズと、`tapex` コマンドを使用してテープ・ドライブのエクササイズを行うことができます。詳細は、それぞれ `diskx(8)` および `tapex(8)` を参照してください。

この章で説明するエクササイズの他に、システムにはエクササイズと同様の機能を持つ Verifier and Exerciser Tool (VET) もサポートされています。VET についての詳細は、最新のファームウェア CD-ROM に付属のドキュメントを参照してください。

11.4.1 システム・エクササイズの実行

システム・エクササイズを実行するには、スーパーユーザとしてログインし、現在のディレクトリを `/usr/field` にしなければなりません。

システム・エクササイズを起動するコマンドにはオプションが用意されており、エクササイズが作業を終了したときに診断出力を保存するファイルを指定できます。

ほとんどのエクササイズ・コマンドには、そのエクササイズの使用方法についての説明が表示されるオンライン・ヘルプ・オプションがあります。オンライン・ヘルプにアクセスするには、コマンドに `-h` オプションを指定します。たとえば、`diskx` エクササイズのヘルプにアクセスするには、次のようにコマンドを入力します。

```
# diskx -h
```

エクササイズは、フォアグラウンドでもバックグラウンドでも実行することができ、フォアグラウンドで `Ctrl/c` を入力することにより、いつでも取り消すことができます。また、一度に複数のエクササイズを実行することもできます。ただし、実行するプロセス数が多くなるに従って、システムの実行速

度は遅くなります。このため、広範囲に渡ってシステムのエクササイズを行うときには、システム上に他のユーザがいないことを確認してください。

システム・エクササイズの実行を、NFS リンクを経由したり、ディスクレス・システム上で行う場合には、いくつかの制約があります。ファイル・システムに書き込みが必要な `fsx` のようなエクササイズの場合は、`root` によるターゲット・ファイル・システムへの書き込みが可能でなければなりません。また、実行されるエクササイズが置かれているディレクトリも、一時ファイルがそのディレクトリに書き込まれるため、`root` による書き込みが可能でなければなりません。

NFS ファイル・システムは `root` による書き込みを防止する方法でマウントされることが多いため、これらの制約に従うのは難しいことがあります。これらの問題のいくつかは、エクササイズを別のディレクトリにコピーしてから実行することで解決できます。

11.4.2 エクササイズ診断の使用

`Ctrl/c` を入力するかタイム・アウトによってエクササイズが停止すると、診断が表示され、またエクササイズの最新のログ・ファイルに保存されます。この診断によって、テスト結果が報告されます。

エクササイズが起動されるたびに、新しいログ・ファイルが `/usr/field` ディレクトリに作成されます。たとえば、`fsx` コマンドが最初の実行されるときには、`#LOG_FSX_01` という名前のログ・ファイルが作成されます。ログ・ファイルには、各エクササイズの実行結果レコードが格納されます。このレコードには、開始/終了時刻や、エラー情報および統計情報が含まれます。開始/終了時刻は、省略時のシステム・エラー・ログ・ファイル `/var/adm/binary.errlog` にも記録されます。このファイルには、デバイス・ドライバやシステムによって報告されたエラー情報も格納されます。

ログ・ファイルは診断の記録を提供します。ただし、エクササイズは9個のログ・ファイルしか持てないため、ログ・ファイルを読み終わった後は必ず削除してください。ログ・ファイルが9個あるエクササイズを実行しようとすると、古いログ・ファイルをいくつか削除して新しいファイルが作成できる状態にするように、エクササイズがユーザに通知します。

エクササイズがエラーを検出した場合は、`dia` コマンド (推奨) または `uerf` コマンドのいずれかを使用して `/var/adm/binary.errlog` ファイルをチェックすることにより、システムのどのデバイスまたは領域に問題がある

かを判断することができます。エラー・ロガーについての詳細は、12.1 節を参照してください。エラー番号およびシグナル番号の意味については、`intro(2)` および `sigvec(2)` を参照してください。

11.4.3 ファイル・システムのエクササイズ

ローカル・ファイル・システムのエクササイズを行うには、`fsx` コマンドを使用します。`fsx` コマンドは、複数のプロセスを開始することにより、指定されたローカル・ファイル・システムのエクササイズを行います。各プロセスでは、ランダムなデータのテスト・ファイルの作成、書き込み、クローズ、オープン、読み取り、検証、およびリンク解除を行います。

注意

NFS ファイル・システムは、`fsx` コマンドではテストしないでください。

`fsx` コマンドの構文は次のとおりです。

fsx [-f *path*] [-h] [-o *file*] [-p *num*] [-t *min*]

コマンド・オプションの詳細については、`fsx(8)` を参照してください。

次の `fsx` コマンドの例は、バックグラウンドで 5 つの `fsxr` プロセスを 60 分間実行することにより、`/usr` ファイル・システムをテストします。

```
# fsx -p5 -f/usr -t60 &
```

11.4.4 システム・メモリのエクササイズ

システム・メモリのエクササイズを行うには、`memx` コマンドを使用します。`memx` コマンドは、複数のプロセスを開始して、システム・メモリのエクササイズを行います。省略時の設定では、各プロセスのサイズは、全システム・メモリのバイト数を 20 で割った値として定義されています。各プロセスの許容最小バイト数は 4,095 です。`memx` コマンドは、オール 1 とオール 0、オール 0 とオール 1、およびランダム・データ・パターンで、割り当てられたメモリ内をテストします。

`memx` エクササイズの実行に必要なファイルには、次のものがあります。

- `memx`
- `memxr`

memx コマンドは、利用可能なスワップ領域の量によって制約を受けます。スワップ領域のサイズと利用可能な内部メモリにより、システムで同時に実行できるプロセス数が決まります。たとえば、スワップ領域が 16 MB で、メモリが 16 MB の場合には、プロセスが 20 個 (省略時の設定) 開始されてすべて同時に動作すると、すべてのスワップ領域が使用されます。これによって、他のプロセスの実行が妨げられます。このため、メモリの量が多くてスワップ領域の量が少ない場合は、`-p` オプション、`-m` オプション、またはその両方を指定して、memx プロセスの数を制限するか、テストするメモリのサイズを制限してください。

memx コマンドの構文は、次のとおりです。

```
memx -s [-h] [-msize] [-ofile] [-pnum] [-tmin]
```

コマンド・オプションの詳細は、memx(8) を参照してください。

次の memx コマンドの例は、4,095 バイトのメモリをテストする 5 つの memxr プロセスを開始して、バックグラウンドで 60 分間実行します。

```
# memx -m4095 -p5 -t60 &
```

11.4.5 共用メモリのエクササイズ

共用メモリ・セグメントのエクササイズを行うには、shmx コマンドを使用します。shmx コマンドは、shmxsb と呼ばれるバックグラウンド・プロセスを作成します。shmx コマンドは、セグメント内の shmxsb のデータに対して、書き込みと読み取りを行います。shmxsb プロセスは、セグメント内の shmx のデータに対して、書き込みと読み取りを行います。

shmx を使用することにより、メモリ・セグメントおよび shmxsb プロセスの数とサイズをテストすることができます。shmx エクササイズは、プロセスが強制終了されるか、`-t` オプションで指定された時間が経過するまで、実行されます。

memx コマンドに `-s` オプションを指定しなかった場合は、memx エクササイズを起動すると、shmx エクササイズが自動的に起動されます。shmx エクササイズは、手動で起動することもできます。shmx コマンドの構文は以下のとおりです。

```
/usr/field/shmx [-h] [-ofile] [-v] [-ttime] [-msize] [-sn]
```

コマンド・オプションの詳細は、shmx(8) を参照してください。

次の例は、省略時のセグメント・サイズのメモリ・セグメントを、省略時の数だけテストします。

```
# shmx &
```

次の例は、100,000 バイトのメモリ・セグメントを 3 つ、180 分間テストします。

```
# shmx -t180 -m100000 -s3 &
```

11.4.6 ターミナル通信システムのエクササイズ

ターミナル通信システムのエクササイズを行うには、`cmx` コマンドを使用します。`cmx` コマンドは、指定された通信回線上でランダムなデータおよびパケット長の、書き込み、読み取り、および検証を行います。

エクササイズを行う回線では、分配パネルまたはケーブルにループバック・コネクタが接続されていなければなりません。また、回線は、`/etc/inittab` ファイルで使用不能に設定され、非モデム回線である必要があります。つまり、`CLOCAL` オプションにオンが設定されていなければなりません。オンが設定されていない場合、時間切れになるか `Ctrl/c` が入力されるまで、`cmx` コマンドは、エラー・メッセージを繰り返しターミナル画面上に表示します。

擬似デバイス回線や `lta` デバイス回線をテストすることはできません。擬似デバイスの名前は、`tty` の直後の文字が、`p, q, r, s, t, u, v, w, x, y, z` のいずれかになっています。たとえば、`ttyp3` です。

`cmx` コマンドの構文は次のとおりです。

```
/usr/field/cmx [-h] [-o file] [-t min] [-l line]
```

コマンド・オプションの詳細は、`cmx(8)` を参照してください。

次の例では、通信回線 `tty22` および `tty34` のエクササイズを、バックグラウンドで 45 分間行います。

```
# cmx -l 22 34 -t45 &
```

次の例では、回線 `tty00` から `tty07` までのエクササイズを、`Ctrl/c` が入力されるまで行います。

```
# cmx -l 00-07
```

基本システム・イベント・チャネルの管理

この章では、システム・イベントのロギング方法と、基本システム・イベント・ロギング・チャネルの構成方法について説明します。また、ログ・ファイルの管理についても説明します。

この章では、次の事項について説明します。

- システム・イベントをモニタリングするためのオプション (12.1 節)
- イベントのモニタリングの設定 (12.2 節)
- システム・クラッシュ後にイベント・ログを回復し参照する方法 (12.3 節)
- ログ・ファイルを管理するためのオプション (12.4 節)

12.1 基本イベント・ロギング機能

本オペレーティング・システムでは、次の 3 つの基本的なメカニズムを使用してシステム・イベントのログを取ります。

- システム・イベント・ロギング機能 (syslogd)。初期化オプションの詳細は、`syslogd(8)` を参照してください。また、構成オプションの詳細は、`syslog.conf(4)` を参照してください。リモート・ロギングの詳細は、`syslog.auth(4)` を参照してください。
- バイナリ・イベント・ロギング機能 (binlogd)。初期化オプションの詳細は、`binlogd(8)` を参照し、`binlog.conf` ファイルについては `binlog.conf(4)` を参照してください。リモート・ロギングの詳細は、`binlog.auth(4)` を参照してください。
- Event Manager。システムのイベントとエラーを管理するための統合的な手段を提供します。Event Manager の導入については、`EVM(5)` を参照してください。また、Event Manager の構成と使用の詳細は、第 13 章を参照してください。

syslogd および binlogd により検出され記録されるイベントは、Event Manager、DECevent、またはエラー報告フォーマット uerf を使用して参照することができます。

システム・イベントの管理方法としては、Event Manager をお勧めします。Event Manager の構成方法についての詳細は、第 13 章を参照してください。Event Manager ビューア evmviewer には、システム・イベントの選択、フィルタ処理、表示を行うためのグラフィカル・ユーザ・インタフェースがあります。詳細は、リファレンス・ページの EVM(5) および evmviewer(8) を参照してください。

システム・イベントは、バイナリ形式で返されることがよくあります。このようなイベントを読み取り可能なテキスト形式で表現するには、次のような変換ツールを使う必要があります。

- サービス契約で提供されるサービス・ツールには、Compaq Analyze のようなイベント分析ツールがあります。詳細は、<http://www.support.compaq.com/svctools/webes/index.html> を参照してください。最新のプロセッサ・モデルでは、以前のプラットフォームで生成された形式と異なるヘッダ形式を使って binlogd イベントが生成されます。新しい形式のイベントは、CEH (Common Event Header) イベントと呼ばれます。CEH イベントを生成しないシステムでは、Compaq Analyze で変換を行うことはできません。このため、DECevent フォーマット・ユーティリティ /usr/sbin/dia をインストールしなければなりません。
- DECevent の制限付き使用ライセンスは、『インストレーション・ガイド』の説明のように、ディストリビューション・キットで提供しています。詳細は、『DECevent Translation and Reporting Utility』と dia(8) を参照してください。
- uerf コマンド・ユーティリティ。uerf(8) を参照してください。

注意

uerf コマンド・ユーティリティは、CEH イベントをサポートしておらず、将来のリリースでは廃止される予定です。できるだけ早く、イベント管理プロシージャを Event Manager に移行してください。

イベント・ロギング機能で作成されるログ・ファイルは保護されており，所有者は `root`，グループは `adm` です。これらのログ・ファイルをチェックするには，適切な特権が必要です。

以降の各項で，イベント・ロギング機能について説明します。

12.1.1 システム・イベント・ロギング

ASCII フォーマットでシステム全体のイベントのログを取るための主要なイベント・ロギング機能として，`syslog` 関数があります。`syslog` 関数は，カーネル，コマンド，ユーティリティ，アプリケーションなどのさまざまなプログラムによって記録されるロギング・メッセージを，`syslogd` デーモンを使用して収集します。`syslogd` デーモンは，メッセージを `/etc/syslog.conf` ファイルに指定されているローカル・ファイル，あるいはリモート・システムに送ります。

オペレーティング・システムをインストールすると，`/etc/syslog.conf` ファイルが作成され，省略時のイベント・ロギング設定が指定されます。`/etc/syslog.conf` ファイルは，ASCII 形式のイベント・メッセージの宛先となるファイル名を指定します。`/etc/syslog.conf` ファイルについては，12.2.1.1 項で説明しています。詳細は，`syslog.conf(4)` を参照してください。

`/etc/syslog.auth` ファイルは，ローカル・ホストへの `syslog` メッセージの転送を許可するリモート・ホストを定義します。システムのセキュリティ上の理由から，このファイルにリストされたりリモート・ホストからのメッセージのみが，`syslogd` デーモンに記録されます。`/etc/syslog.auth` ファイルが存在しない場合は，すべてのリモート・ホストからのイベント転送が可能になります。

`/etc/syslog_evm.conf` ファイルは，Event Manager イベントの形式で `syslogd` デーモンから Event Manager に転送される `syslogd` メッセージを定義します。`syslogd` 転送機能が `-e` オプションでオンになっていれば，これらの `syslogd` メッセージは，`syslogd` によって Event Manager デーモン `evmd` にポストされます。イベント転送は常に，省略時はオンです。`-E` オプションを使い，必要に応じてオフにします。イベントは，`sys.unix.syslog.facility` という Event Manager 名でポストされます。

詳細は，`syslog.auth(4)` と `syslog_evm.conf(4)` を参照してください。

12.1.2 バイナリ・イベント・ロギング

バイナリ・イベント・ロギング機能は、カーネル内のハードウェア・イベントおよびソフトウェア・イベントを検出し、詳細情報をバイナリ形式のレコードで記録します。このようなイベントの一部は、syslog 関数を使用してもログを取ることができます。syslog 関数による情報はバイナリ・イベント・ロギング機能によるログほど詳しくはありません。

バイナリ・イベント・ロギング機能は、binlogd デーモンを使用して、さまざまなイベント・ログ・レコードを収集します。binlogd デーモンはこれらのイベント・ログ・レコードを、システムのインストール時に作成される省略時の構成ファイル /etc/binlog.conf に指定したローカル・ファイル、またはリモート・システムに送ります。12.2.1.3 項では、/etc/binlog.conf ファイルについて説明しています。

DECevent (Compaq Analyze) は、バイナリ・イベントをシステムのバイナリ・イベント・ログ・ファイルから ASCII 形式のレポートに変換するために使います。DECevent は、コマンド行で dia コマンドを入力すると起動されます。オプションを付けずにコマンドを入力すると、DECevent は直ちにイベント・ログ・ファイルにアクセスして、内容を変換し、例 12-1 に示すようにイベントを表示します。イベントは、すべてのイベントが表示されるか、Ctrl/c キーが押されるまで、上にスクロールし続けます。

例 12-1: 変換されたイベントの例

```
**** V3.3 ***** ENTRY 4
***** [1]
Logging OS                2[OS] [2]
System Architecture       2.
Alpha Event sequence number 440.
Timestamp of occurrence   22-AUG-2002 18:24:31 [3]
Host name                 Host Name

System type register      x0000001B      AlphaServer 800 or 1000A
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000

Event validity [4]        1. O/S claims event is valid
Event severity           5. Low Priority
Entry type               301. Shutdown ASCII
Message Type             -1. - (minor class)
SWI Minor class          9.
ASCII Message SWI Minor sub class 2. Shutdown ASCII Message
System halted by root:   System going down @ 6:24PM on 22 Aug
```


例 12-1: 変換されたイベントの例 (続き)

Please log off in good time 5

- ❶ 変換されたログ内のイベントの数です。この数値はイベントの選択またはフィルタリングに基づいていることがあります。
- ❷ オペレーティング・システム ([OS]) とシステム・アーキテクチャを示す文字列です。
- ❸ イベントの発生時間を示すタイムスタンプ (日付とシステム・クロックの時刻) と発生したシステムの名前 (<host name>) です。
- ❹ イベントの有効性, 重大度, タイプについての情報です。この例では, システムがシャットダウンされたという, 情報的なメッセージです。
- ❺ イベントによって実際に記録されたメッセージです。イベントの発生時に端末またはコンソールにも表示されていることがあります。

DECevent ユーティリティの管理についての詳細は, 次のドキュメントを参照してください。

- 『*DECevent Translation and Reporting Guide*』
- dia(8)

Compaq Analyze は, ルール・ベースのハードウェア障害管理の診断アプリケーションであり, エラー・イベント解析と変換を行います。Compaq Analyze の多重イベント相関解析機能により, バイナリ・システム・イベント・ログまたはその他の特定のバイナリ・ログ・ファイルに格納されたイベントを解析する機能を提供しています。Compaq Analyze がインストールされると, SysMan Station から直接 GUI インタフェースを起動することができます。起動するには, 「Host Icon」を選択し, [Tool] メニューから [Compaq Analyze] を選択します。

12.2 イベント・ロギングの構成

この節の説明に従って構成ファイルを変更すると, 省略時の構成を変更できます。たとえば, 重要な, システムにとってクリティカルなイベントの

みを記録し、情報通知イベントは無視するように構成を変更できます。メール・サービスやプリント・サービスなどの一部のサブシステムに限定して、イベント・メッセージが記録される方法や位置を制御することもできます。システム・イベント監視の最適な方式は、第 13 章で示すように Event Manager を使用することです。Event Manager は、イベントの統合とフィルタ処理を可能にしています。

システムおよびバイナリのイベント・ロギングを使用可能にするには、特殊ファイルが存在し、イベント・ロギング・デーモンが実行されていなければなりません。詳細については、12.2.3 項および 12.2.4 項を参照してください。

ファイル `/var/adm/syslog.dated` と、`/var/adm` ディレクトリにあるその他のファイルは、コンテキスト依存のシンボリック・リンク (CDSL) であり、1 つのシステムと複数のクラスタを結合する機能があります。syslog ディレクトリの CDSL は `/var/cluster/members/member0/adm/syslog.dated` です。これらのファイルを操作する際は、シンボリック・リンクを切断しないように注意してください。CDSL についての詳細は、第 6 章を参照してください。

12.2.1 構成ファイルの編集

省略時のシステム・イベント・ロギングの設定またはバイナリ・イベント・ロギングの設定を使用したくない場合は、構成ファイル `/etc/syslog.conf` または `/etc/binlog.conf` を編集して、システムでのイベント・ロギングの方法を指定することができます。これらのファイルには、次のデータを指定します。

- ファシリティ (メッセージのソース、またはメッセージを生成するシステムの構成要素)
- 優先順位 (メッセージの重大度)
- メッセージの記録先

以降の各項では、構成ファイルの編集方法を説明します。

12.2.1.1 syslog.conf ファイルの編集

syslogd デーモンが省略時の設定以外の構成ファイルを使用するように設定するには、次のように、syslogd コマンドでファイル名を指定します。

```
# syslogd -f config_file
```

次に、省略時の `/etc/syslog.conf` ファイルの例を挙げます。

```
#
# syslogd config file
#
# facilities: kern user mail daemon auth syslog lpr binary
# priorities: emerg alert crit err warning notice info debug
#
#1  2                                3
kern.debug                /var/adm/syslog.dated/kern.log
user.debug                /var/adm/syslog.dated/user.log
daemon.debug              /var/adm/syslog.dated/daemon.log
auth.crit;syslog.debug    /var/adm/syslog.dated/syslog.log
mail,lpr.debug            /var/adm/syslog.dated/misc.log
msgbuf.err                /var/adm/crash.dated/msgbuf.savecore
kern.debug                /var/adm/messages
kern.debug                /dev/console
*.emerg                   *
```

`/etc/syslog.conf` ファイルの各エントリは次の情報を表します。

❶ メッセージを生成するシステムの一部であるファシリティ

❷ 重大度

syslogd デーモンは、指定された重大度以上のメッセージはすべて記録します。たとえば `err` レベルを指定すると、レベル `err`、`crit`、`alert`、`emerg`、`panic` のすべてのメッセージのログが作成されます。

❸ メッセージを記録する場所

ログ・ファイルまたは、`/dev/console` のようなデバイスです。

syslogd デーモンは、空白行と `#` (番号記号) で始まる行を無視します。このため、行の最初に `#` を指定して `/etc/syslog.conf` ファイルにコメントを記述したり、エントリを無視させたりすることができます。

ファシリティおよび重大度は、1 つ以上のタブ文字または空白によって記録先と区切ります。

セミコロンで区切れば、2 つ以上のファシリティおよびその重大度を指定することができます。上記の例では、重大度 `crit` およびそれ以上の `auth` ファシリティからのメッセージと、重大度 `debug` およびそれ以上の `syslog` ファシリティからのメッセージは、`/var/adm/syslog.dated/syslog.log` ファイルにログが取られます。

コンマで区切ることによって、2 つ以上のファシリティを指定することができます。上記の例では、mail からのメッセージと、重大度が debug 以上のレベルの lpr ファシリティは、/var/adm/syslog.dated/misc.log ファイルに記録されます。

ファシリティ

次のファシリティを指定することができます。

| | |
|--------------------------------|---|
| kern | カーネルによって生成されるメッセージ。これらのメッセージはユーザ・プロセスでは生成できない。 |
| user | ユーザ・プロセスによって生成されるメッセージ。省略時の設定。 |
| mail | メール・システムによって生成されるメッセージ。 |
| daemon | システム・デーモンによって生成されるメッセージ。 |
| auth | 認証システムによって生成されるメッセージ (login , su , getty など)。 |
| lpr | ライン・プリンタ・スプーリング・システムによって生成されるメッセージ (lpr , lpc , lpd など)。 |
| local0 , local1 ~ local7 | ローカルでの使用のために確保されている。 |
| mark | 20 分ごとに優先順位 info のメッセージを受け取る。メッセージを受け取る間隔を変更する場合は、syslogd -m コマンドを使用する。 |
| msgbuf | システム・クラッシュから回復したカーネル syslog メッセージ・バッファ。savecore コマンドと syslogd デーモンは、msgbuf ファシリティを使用して、システム・イベント・メッセージをクラッシュから回復する。 |
| * | システムのすべての部分で生成されるメッセージ。 |

重大度

次の重大度を指定することができます。重大度の高い順にリストしてあります。

| | |
|------------------|--|
| emerg または panic | パニック状態。このメッセージはすべてのユーザにブロードキャストできる。 |
| alert | システム・データベースが破壊された場合など、ただちに修復しなければならない状態。 |
| crit | ハード・デバイス・エラーなどのクリティカルな状態。 |
| err | エラー・メッセージ。 |
| warning または warn | 警告メッセージ。 |
| notice | エラー状態ではないが、特殊なケースとして扱われる状態。 |
| info | 情報メッセージ。 |
| debug | プログラムのデバッグに使用する情報を含んでいるメッセージ。 |
| none | 特定のファシリティのメッセージを使用不能にするメカニズム。 |

記録先

次のメッセージ記録先を指定することができます。

| | |
|-------|--|
| 絶対パス名 | 指定されたファイルにメッセージを追加する。個々のファシリティのメッセージの宛先として、それぞれ別々のファイルを指定する必要がある。たとえば、kern.log、mail.log、または lpr.log などを指定する。 |
|-------|--|

| | |
|-------------------|---|
| ホスト名 (先頭に @ を付ける) | 指定されたホスト上の <code>syslogd</code> デーモンに、メッセージを転送する。 <code>syslogd</code> デーモンの起動時に <code>-R</code> オプションを指定した場合、メッセージは転送されない。詳細は、12.2.2 項を参照。 |
| コンマで区切られたユーザのリスト | 指定したユーザがログインしている場合に、それらのユーザにメッセージを送る。 |
| * | ログインしているすべてのユーザにメッセージを送る。 |

日次ログ・ファイル

`syslogd` デーモンが日次ログ・ファイルを作成するように指定することができます。日次ログ・ファイルを作成するには、次の構文を使用して、メッセージの記録先のパス名を指定してください。

```
/var/adm/syslog.dated/{file}
```

file 変数には、ログ・ファイルの名前を指定します。たとえば、`mail.log` または `kern.log`などを指定します。

記録先のパス名 `/var/adm/syslog.dated/file` を指定すると、`syslogd` デーモンは、次の構文で `/var/adm/syslog.dated` ディレクトリの下にサブディレクトリを毎日作成して、そのサブディレクトリにログ・ファイルを作成します。

```
/var/adm/syslog.dated/date/file
```

- *date* 変数には、ログ・ファイルが作成された日、月、および時刻を指定します。
- *file* 変数には、`/etc/syslog.conf` ファイルに指定したログ・ファイルの名前を指定します。

`syslogd` デーモンは、24 時間ごと、システムのブート時、または `syslogd` デーモンの再起動時または再構成後に、新しい *date* ディレクトリを自動的に作成します。最新のログは `/var/adm/syslog.dated/current` ディレクトリにあります。`current` ディレクトリは、最新の *date* ディレクトリへのシンボリック・リンクです。

たとえば、info 以上のレベルのすべてのメール・メッセージの日次ログ・ファイルを作成するには、/etc/syslog.conf ファイルを編集して、次のような行を追加します。

```
mail.info                                /var/adm/syslog.dated/mail.log
```

/etc/syslog.conf に上記の行を指定すると、syslogd デーモンは、次のような日次ディレクトリおよびファイルを作成します。

```
/var/adm/syslog.dated/11-Jan-12:10/mail.log
```

12.2.1.2 Event Manager を使用するための syslog の構成

省略時の設定では、syslogd は -e オプションを使用してイベントを Event Manager に転送するように構成されます (12.2.4 項を参照)。syslog_evm.conf ファイルを変更すると、どの syslog イベントが Event Manager に転送されるかを指定できます。このファイルが存在しないか、存在しても登録エントリがない場合は、syslog メッセージは Event Manager にポストされません。

省略時の syslog_evm.conf ファイルには、例 12-2 に示すようなエントリが含まれています (情報用のファイル・ヘッダは除く)。

例 12-2: syslog_evm.conf ファイルのエントリの例

```
1 2
*.emerg
# above forwards all emergency events to EVM 3
kern.info+ 4
user.notice+
mail.notice+
daemon.notice+
auth.notice+
syslog.notice+
```

- 1 各行エントリの最初の部分は、カーネルの kern のように、メッセージを生成するファシリティを指定します。アスタリスク (*) は、ファシリティをすべて選択することを示します。この例では、*.emerg により、優先順位が緊急のメッセージがすべて Event Manager に転送されます。

ファシリティに対するエントリを作成するか、既存のエントリを削除することで、転送されるイベントが選択できます。エントリは、12.2.1.1 項のファシリティの表のキーワードに基づいています。

- ② 各エントリの 2 番目の部分は、12.2.1.1 項の重大度レベルの表のキーワードに基づくメッセージ優先順位を示します。
- ③ 番号記号 (#) を前に付けてコメントを付加できます。ただし、転送エントリとコメントを同じ行に混在させることはできません。
- ④ 優先順位に正符号 (+) を付けると、指定した優先順位とそれ以上の優先順位を持つメッセージが転送されます。1 つのファシリティでの警告、クリティカル、緊急などの重大度を個別に選択したい場合は、それぞれの優先順位に対して 1 行を記述します。

イベントは、`sys.unix.syslog.facility` という Event Manager 名でポストされます。

詳細は、`syslog_evm.conf(4)` および 第 13 章を参照してください。

12.2.1.3 binlog.conf ファイルの編集

省略時の設定以外の構成ファイルを `binlogd` デーモンに使用させる場合は、ファイル名を指定して `binlogd -fconfig_file` コマンドを実行します。`binlogd` デーモンは、すべてのイベントを Event Manager へ転送します。第 13 章の説明のように、Event Manager ユーティリティを使用すると `binlog` イベントのフィルタ処理と選択ができます。

`binlogd` イベントをリモート・ホストに送ることができます。リモート・ロギング・オプションの詳細は、`binlogd(8)` を参照してください。特に、リモート・アクセス用の `inet` ポートの作成を制御する `-R` と `-r` には注意してください。

`/etc/binlog.conf` ファイルの例は次のとおりです。

```
#
# binlogd configuration file
#
# format of a line:  event_code.priority          destination
#
# where:
# event_code  - see codes in binlog.h and man page, * = all events
# priority    - severe, high, low, * = all priorities
# destination - local file pathname or remote system hostname
#
#
*. *                /usr/adm/binary.errlog
dumpfile           /usr/adm/crash/binlogdumpfile
102.high           /usr/adm/disk.errlog
```

12-12 基本システム・イベント・チャネルの管理

1 2

3

/etc/binlog.conf ファイルの各エントリには、`dumpfile` イベント・クラスのエントリを除き、次の 3 つのフィールドがあります。

- 1 イベントを生成しているシステムの部分を示すイベント・クラス・コード。
- 2 イベントの重大度。イベント・クラスとして `dumpfile` を指定する場合、重大度は指定できない。
- 3 バイナリ・イベント・レコードの記録先。

`binlogd` デーモンは、空白行と # (番号記号) で始まる行を無視します。このため、行の最初に # を指定してファイルにコメントを記述したり、エントリを無視させたりすることができます。

イベント・クラスと重大度は、1 つ以上のタブ文字または空白によって記録先と区切ります。

次のイベント・クラス・コードを指定することができます。

| クラス・コード | 説明 |
|-----------------------|--|
| 汎用 | |
| * | すべてのイベント・クラスを指定する。 |
| <code>dumpfile</code> | クラッシュ・ダンプからのカーネル・バイナリ・イベント・ログ・バッファの復元を指定する。重大度は指定できない。 |
| ハードウェア検出イベント | |
| 100 | CPU マシン・チェックと例外または、一般例外フォールト |
| 101 | メモリ |
| 102 | ディスク |
| 103 | テープ |
| 104 | 制御装置 |
| 105 | アダプタ |
| 106 | バス |
| 107 | ストレイ割り込み |
| 108 | コンソール・イベント |
| 109 | スタック・ダンプ |

| クラス・コード | 説明 |
|------------------|-------------------------------------|
| 110 | 一般マシン状態 |
| 113 | ダブル・エラー停止 |
| 115 | 修正可能 (不可) 環境 |
| 120 | 修正報告の使用不可 |
| 195 | SWCC (StorageWorks Command Console) |
| 196 | I2O ブロック・ストレージ |
| 198 | SWXCR RAID 制御装置 |
| 199 | SCSI CAM |
| ソフトウェア検出イベント | |
| 201 | CI ポート・ツー・ポート・ドライバ |
| 202 | システム通信サービス |
| 203 | LSM 注意 |
| 204 | LSM 警告 |
| 205 | LSM 継続 |
| 206 | AdvFS ドメイン・パニック |
| ASCII 形式の情報メッセージ | |
| 250 | 汎用 ASCII 形式情報メッセージ |
| 動作上のイベント | |
| 300 | スタートアップ ASCII メッセージ |
| 301 | シャットダウン ASCII メッセージ |
| 302 | パニック ASCII メッセージ |
| 310 | タイム・スタンプ |
| 350 | 診断状況 ASCII メッセージ |
| 351 | 修復および保守 ASCII メッセージ |
| 400 | フィルタ・ログ・イベント (filterlog でのみ使用) |

重大度

次の重大度を指定することができます。

* 全ての重大度

| | |
|--------|---|
| severe | システム動作に致命的な影響を及ぼすような回復不能なイベント |
| high | 回復可能なイベント，またはシステム動作に致命的な影響を及ぼさない回復不能なイベント |
| low | 情報イベント |

記録先

次の記録先を指定することができます。

絶対パス名 binlogd デモンがバイナリ・イベント・レコードを追加するファイルの名前を指定する。

@hostname binlogd デモンがバイナリ・イベント・レコードを送るホストの名前を指定する。先頭にアットマーク (@) を付ける。イベント・クラスとして `dumpfile` を指定した場合，ホストへレコードを送ることはできない。

操作上のタイム・スタンプ (310) イベントは，自動的に転送されません。

12.2.2 リモート・メッセージと **syslog** のセキュリティ

リモート・ホストのドメイン・ホスト名がローカルの `/etc/syslog.auth` ファイルに登録されていなければ，ローカル・システムはこのリモート・ホストからの `syslog` メッセージは何も記録しません。システム上で `syslogd` の保護を行うようにし，そのシステムへ `syslog` メッセージを転送するように他のホストを構成する場合は，次の手順を実行してください。

1. `su` コマンドを使用してスーパーユーザ (root) になります。
2. テキスト・エディタを使用して `/etc/syslog.auth` ファイルを作成します。
3. ローカル・システムの `/etc/syslog.auth` ファイルへの `syslog` メッセージの転送を許可するリモート・ホストの名前を追加します。このホスト名は，次の基準を満たさなければなりません。

- 各リモート・ホスト名は、`/etc/syslog.auth` ファイル内で別々の行に記述すること。# 文字で始まる行はコメントなので、無視されます。
- ホスト名は完全なドメイン名でなければならない (`trout.fin.huk.com` など)。
- ドメイン・ホスト名を指定する場合、そのホスト名がローカルの `/etc/hosts` ファイルに記述されているか、ローカル・システムがネーム・サーバ (BIND など) によってそのホスト名を解釈できなければならない。
- ホスト名には、`/sys/include/sys/param.h` ファイルの `MAXHOSTNAMELEN` 定数で定義されている文字数まで使用できる。ただし、`/etc/syslog.auth` ファイルの各行に使用できる文字は 512 文字までである。
- 正符号 (+) だけを記述すると、すべてのホストからのイベント転送を許可します。また、負符号 (-) をホスト名の前に付けて、そのホストからのイベント転送を明示的に禁止することもできます。
`/etc/syslog.auth` ファイルがシステム上に存在しない場合、すべてのホストからの転送が許可されます。

4. `/etc/syslog.auth` の所有者を `root` にします。

```
# chown root /etc/syslog.auth
```

5. `/etc/syslog.auth` のパーミッションを `0600` にします。

```
# chmod 0600 /etc/syslog.auth
```

リモート・ホストが送信するイベントをリッスンする `inet` ポートを `syslogd` デーモンが作成しないようにするには、デーモン開始時に `-R` オプションを指定します。この動作を省略時の操作モードにするには、`/sbin/init.d/syslog` ファイル内のスタートアップ・コマンド行を編集します。`-R` オプションを使うと、`syslogd` デーモンが、別のシステムにイベントを転送できなくなります。

詳細は、`syslog.auth(4)` および `syslogd(8)` を参照してください。

12.2.3 特殊ファイルの作成

文字型特殊ファイル `/dev/klog` が存在しないと、`syslogd` デーモンはカーネル・メッセージのログを取ることができません。`/dev/klog` ファイルが存在しない場合は、次のコマンドで作成してください。

```
# /dev/MAKEDEV /dev/klog
```

同様に、文字型特殊ファイル `/dev/kbinlog` が存在しないと、`binlogd` デーモンは、ローカル・システム・イベントのログを取ることができません。`/dev/kbinlog` がない場合は、次のコマンドで作成してください。

```
# /dev/MAKEDEV /dev/kbinlog
```

詳細は `MAKEDEV(8)` を参照してください。

12.2.4 イベント・ロギング・デーモンの起動と停止

`syslogd` および `binlogd` デーモンは、システム・スタートアップ時に `init` プログラムによって自動的に起動されます。この2つのデーモンは必ず起動されるようにしなければなりません。デーモンを起動するコマンドにオプションを指定することもできます。

12.2.4.1 `syslogd` デーモン

`init` プログラムによって `syslogd` デーモンが必ず起動されるように設定しなければなりません。`syslogd` デーモンが起動されない場合、または `syslogd` デーモンを起動するコマンドにオプションを指定する場合には、`/sbin/init.d/syslog` ファイルを編集する必要があります。ファイルを編集するとき、`syslogd` コマンド行を追加または変更します。このコマンドは、手動でも起動できます。

`syslogd` デーモンを起動するコマンドの構文は、次のとおりです。

```
/usr/sbin/syslogd [-b rcvbufsz] [-d] [-e | -E] [-f config_file] [-m  
mark_interval] [-p path] [-r | -R] [-s]
```

省略時の設定では、デーモンの初期化で使われるのは `-e` オプションだけです。`-e` オプションは、デーモンが自動的にイベントを Event Manager に転送するように構成します。次のように `ps` コマンドを使用すると、現在の `syslogd` の構成が確認できます。

```
# /sbin/ps agx | grep syslogd
```

```
261 ??      S      0:00:10  usr/sbin/syslogd -e
```

コマンド・オプションについての詳細は、`syslogd(8)` を参照してください。

注意

`/var/adm` ディレクトリがマウントされていることを確認してください。マウントされていない場合、`syslogd` デーモンは正しく動作しません。

`syslogd` デーモンは、次のものからメッセージを読み取ります。

- ドメイン・ソケット `/dev/log` ファイル
`syslogd` デーモンによって自動的に作成されます。
- インターネット・ドメイン (UDP) ソケット
`/etc/services` ファイルに指定されています。セキュリティ上の理由から、`-R` オプションを使ってこのソケットを使用不可にするか、許可するホストを `/etc/syslog.conf` ファイル内に指定することもできます。
- デバイス特殊ファイル `/dev/klog`
カーネル・メッセージのみを記録します。

他のプログラムからのメッセージは、`openlog`、`syslog`、および `closelog` 呼び出しを使用します。

`syslogd` デーモンが起動されると `/var/run/syslog.pid` ファイルを作成し、そのプロセス ID 番号をこのファイルに格納します。システムをシャット・ダウンする場合は、その前に、`syslogd` プロセス ID 番号を使用してデーモンを停止させてください。

正常なシステム動作中に、物理メモリに置かれているカーネルの `syslog` メッセージ・バッファにデータがいれられると、`syslogd` デーモンが実行されます。`syslogd` デーモンは、`/dev/klog` ファイルを読み取って、カーネル `syslog` メッセージ・バッファのコピーを取得します。`syslogd` デーモンは、バッファの先頭から検索を開始して、検出した各メッセージをシーケンシャルに処理します。個々のメッセージの前には、`/etc/syslog.conf` ファイルに指定されているのと同じファシリティ・コードと優先順位コードが付けられます。次に、`syslogd` デーモンは、`/etc/syslog.conf` ファイルに指定されている記録先にメッセージを送信します。

イベント・ロギング・デーモン `syslogd` を停止するには、次のコマンドを使用します。

```
# kill `cat /var/run/syslog.pid`
```

次のコマンドを使用すれば、デーモンを再起動せずに、構成ファイル `/etc/syslog.conf` に対する変更を有効にすることができます。

```
# kill -HUP `cat /var/run/syslog.pid`
```

12.2.4.2 binlogd デーモン

`init` プログラムが `binlogd` デーモンを必ず起動するように設定しなければなりません。 `binlogd` デーモンが起動されない場合、または `binlogd` デーモンを起動するコマンドにオプションを指定する場合は、`/sbin/init.d/binlog` ファイルを編集し、`binlogd` コマンド行を追加または変更しなければなりません。このコマンドは、手動でも起動できます。 `binlogd` コマンドは、次のオプションをサポートしています。

```
/usr/sbin/syslogd [-d] [-f config_file] [-r | -R]
```

コマンド・オプションについては、`binlogd(8)` を参照してください。

`binlogd` デーモンは、次のものからバイナリ・イベント・レコードを読み取ります。

- インターネット・ドメイン・ソケット (`binlogd` , 706/udp)
`/etc/services` ファイルに指定されています。セキュリティ上の理由から、`-R` オプションを使ってこのソケットを使用不可にすることもできます。また、許可するホストを `/etc/binlog.conf` ファイル内に指定することもできます。
- 特殊ファイル `/dev/kbinlog`

`binlogd` デーモンが起動されると `/var/run/binlogd.pid` ファイルを作成して、そのプロセス ID 番号をこのファイルに格納します。 `binlogd` デーモンを停止または再設定する場合はこのプロセス ID 番号を使用します。

正常なシステム動作中に、データがカーネルのバイナリ・イベント・ログ・バッファにいれられるか、またはインターネット・ドメイン・ソケット上で受け取られると、`binlogd` デーモンが実行されます。次に、`binlogd` デーモンは、特殊ファイル `/dev/kbinlog` またはソケットからデータを読み取ります。個々のレコードには、イベント・クラス・コードと重大度コードが含まれています。 `binlogd` デーモンは、各バイナリ・イベント・レ

コードを処理して、それを `/etc/binlog.conf` ファイルに指定されている記録先にログを取ります。

`binlogd` デーモンを停止するには、次のコマンドを使用します。

```
# kill `cat /var/run/binlogd.pid`
```

次のコマンドを使用すれば、デーモンを再起動せずに、構成ファイル `/etc/binlog.conf` に対する変更を有効にすることができます。

```
# kill -HUP `cat /var/run/binlogd.pid`
```

12.2.5 カーネルのバイナリ・イベント・ロガーの設定

カーネルのバイナリ・イベント・ロガーを設定するには、省略時のキーワードを変更してカーネルを再構築します。次の設定ができます。

- カーネルのバイナリ・イベント・ログ・バッファを、システムに必要な大きさに調節することができます。
- バイナリ・イベント・ロガー、およびバイナリ・イベント・ログへのカーネル ASCII メッセージのロギング機能を、使用可能に設定したり、使用不能に設定することができます。

`/sys/data/binlog_data.c` ファイルは、バイナリ・イベント・ロガーの設定を定義します。省略時の設定では、バッファ・サイズは 24 K バイト、バイナリ・イベント・ロギングは使用可能、カーネル ASCII メッセージのロギングは使用不能に設定されます。設定を変更するには、ファイル内のキーワード `binlog_bufsize` および `binlog_status` の設定を変更します。

`binlog_bufsize` キーワードは、バイナリ・イベント・ロガーによって使用されるカーネル・バッファのサイズを指定します。バッファのサイズは、8 K (8,192) ~ 1 M (1,048,576) バイトが設定可能です。ワークステーションなどの小規模なシステム構成では、バッファ・サイズが小さくても構いません。多数のディスクを使用する大規模なサーバ・システムでは、大きいバッファが必要になる場合があります。

`binlog_status` キーワードは、バイナリ・イベント・ロガーの動作を指定します。`binlog_status` キーワードには、次の値を指定することができます。

| | |
|--------|---------------------------|
| 0 (ゼロ) | バイナリ・イベント・ロガーを使用不能に設定します。 |
|--------|---------------------------|

BINLOG_ON

バイナリ・イベント・ロガーを使用可能に設定します。

BINLOG_ASCIIION

バイナリ・イベント・ロガーが使用可能になっている場合に、バイナリ・イベント・ログへのカーネル ASCII メッセージのロギングを可能にします。この設定は、次のように BINLOG_ON とともに指定しなければなりません。

```
int binlog_status = BINLOG_ON | BINLOG_ASCII;
```

/sys/data/binlog_data.c ファイルを変更した後、新しいカーネルを再構築してブートしなければなりません。

12.3 システム・クラッシュ後のイベント・ログの回復

システム・クラッシュ時の未処理のメッセージおよびバイナリ・イベント・ログ・レコードは、システムをリブートすることによって回復することができます。

/etc/syslog.conf ファイルの msgbuf.err エントリで、ダンプ・ファイルから回復されるカーネル syslog メッセージ・バッファ msgbuf の記録先を指定します。カーネル syslog メッセージ・バッファ・ファイルに関する、/etc/syslog.conf の省略時のファイル・エントリは、次のとおりです。

```
msgbuf.err /var/adm/crash/msgbuf.savecore
```

/etc/binlog.conf ファイルの dumpfile エントリで、ダンプ・ファイルから回復されるカーネル・バイナリ・イベント・ログ・バッファの記録先となるファイル名を指定します。カーネル・バイナリ・イベント・ログ・バッファ・ファイルに関する、/etc/binlog.conf の省略時のファイル・エントリは、次のとおりです。

```
dumpfile /usr/adm/crash/binlogdumpfile
```

クラッシュが発生すると、syslogd および binlogd デーモンは、/dev/klog および /dev/kbinlog 特殊ファイルを読み取ることができず、メッセージおよびバイナリ・イベント・レコードの処理ができません。システムをリブートすると savecore コマンドが実行され、ダンプ・ファイルが存在すれば、savecore コマンドはダンプ・ファイルからカーネル syslog

メッセージおよびバイナリ・イベント・ログ・バッファを回復します。
savecore の実行後に、syslogd および binlogd デーモンが起動されます。

syslogd デーモンは、syslog メッセージ・バッファ・ファイルを読み取り、そのデータが有効であるかどうかを検査します。その後、
/etc/syslog.conf ファイル内の情報を使用して /dev/klog ファイルからのデータを処理するのと同じ方法で、データを処理します。

binlogd デーモンは、バイナリ・イベント・ログ・バッファ・ファイルを読み取り、そのデータが有効であるかどうかを検査します。その後、
/etc/binlog.conf ファイル内の情報を使用して特殊ファイル /dev/kbinlog からのデータを処理するのと同じ方法で、データを処理します。

syslogd および binlogd デーモンがバッファ・ファイルの使用を終了すると、このバッファ・ファイルは削除されます。

12.4 ログ・ファイルの管理

管理の行き届いたシステムでは、次の対処が行われているため、ログ・ファイルのサイズが問題になることはありません。

- 記録するイベントを注意深く選択している。
- 多数のポストを行うエラー状態のログをモニタリングしている。
- 重要なイベント・ログを定期的にアーカイブしバックアップしている。

/var/spool/cron/crontabs/root ファイルには、次のように、ログ・ファイルを管理するためのモデル・エントリが含まれています。

```
0 2 * * 0 /usr/sbin/logclean /var/adm/wtmp > /dev/null
```

また、cron デーモンを使用して、他のログ・ファイルを削除するように指定することもできます。ただし、ローカル・サイトの要件に応じて、重要なログ・ファイルは保存するかアーカイブしてください。

次の例は、/var/adm/syslog.dated ディレクトリ内の古いログをクリーンアップする crontab ファイル・エントリです。

```
40 4 * * * find /var/adm/syslog.dated/* -depth -type d -ctime +7 -exec rm -rf {} \;
```

このエントリにより、/var/adm/syslog.dated ディレクトリ下のすべてのディレクトリのうち 8 日以上前に作成されたものが、毎日 4 時 40 分に

その内容とともに削除されます。詳細は、第 3 章、および `crontab(1)` を参照してください。

12.5 /var/adm/messages 内の起動ログ・メッセージ

格納されるメッセージの数は、ブート・ログ・メッセージを格納するためのメッセージ・バッファの、`msgbuf_size` カーネル属性によって制御されるサイズに依存します。この属性の省略時の最小値は、物理メモリが 128 MB までのシステムでは 8KB です。物理メモリが 128 MB よりも大きいシステムでは、`msgbuf_size` の値は、1 MB のメモリごとに 64 バイトとして計算され、自動的に設定されます。たとえば、512 MB のシステムでは、この値は $512 \times 64 = 32,768$ (32 KB) です。

アダプタやデバイスが多数ある大規模なシステムでは省略時の値では不十分で、`/var/adm/messages` ファイルのメッセージが失われることがあります。デバイスの数が少ない、メモリ容量の大きいシステムでは、この値では大きすぎる場合があります。この場合は、バッファ・スペースを小さくしてメモリを解放できます。

システムのブート・ログ・レコードが不完全な場合や、割り当て値を小さくしてバッファ・スペースを解放したい場合は、次の手順で `msgbuf_size` 属性の値を変更します。

1. コマンド行から `dxkernel tuner` グラフィカル・ユーザ・インタフェースを起動します。
2. 「generic」サブシステムを選択して、[サブシステムの選択] を選択します。「generic」のラベルの付いた「Subsystem Attributes」ダイアログボックスが表示されます。
3. 「`msgbuf_size`」に、「ブート時の値」の新しい値を設定します。
4. [了解] ボタンをクリックして変更を確定し、ダイアログボックスを閉じます。
5. メイン・ウィンドウの [オプション] メニューから「すべてをリセット」を選択します。
6. メイン・ウィンドウの [ファイル] メニューから「終了」を選択します。

`sysconfig` コマンドおよび `sysconfigdb` コマンドを使用してもこの変更を行うことができます (第 4 章を参照)。



Event Manager の使用方法

Event Manager は、高機能のイベント管理システムです。これは、従来のイベント処理機能の他に、独自のイベントと他のチャネルからのイベントを統合して 1 つの情報ソースとすることで、システム動作のモニタリング作業を簡単にします。Event Manager には、グラフィカル・イベント・ビューアと、コマンド行ツールのフル・セットがあります。これらの機能は、SysMan Menu のアプリケーション群および SysMan Station に統合されています。

この章では、次のトピックについて説明します。

- Event Manager の概要 (13.1 節)
- Event Manager のセットアップおよびカスタマイズ方法 (13.2 節)
- システムの管理に Event Manager を役立てる方法 (13.3 節)
- Event Manager の一般的な問題のトラブルシューティング (13.4 節)

13.1 Event Manager の概要

UNIX のシステム管理者の仕事で重要なものとして、システムの状態を監視し、通常と異なる状態になったときに、適切な処置をとることがあります。通常と異なる状態の例には、ディスクが一杯になったときや、プロセッサがハードウェア障害を報告したときなどがあります。また、日常処理が毎日正常に実行されていることを確認することや、システム構成値を見直すことも重要です。このような状態や、タスクの完了を、システムのイベントといいます。

イベントとは、関心のある事象が発生したことを表します。たとえば、アクションが行われたこと、ある条件が満たされたこと、あるいはアプリケーションが使用できる状態かどうかを確認すべき時期であることなどがあります。管理者にとって興味の対象となるイベントもあれば、別のクラスのユーザにとって興味の対象となるイベントもあります。システム・イベントは、次のような他のシステム・エンティティにとっても重要です。

- システム監視ソフトウェア

- オペレーティング・システム・ソフトウェア
- エンドユーザ・アプリケーション・プログラム
- ハードウェア・コンポーネント

イベントに関心を持つエンティティは、ローカル・システムの一部であっても、リモート・システムの一部であっても構いません。

システム・コンポーネントは、レポートすべきものを検出したら、イベント・チャンネルを通してその情報を利用可能にします。イベント・チャンネルとは、イベント情報を公表したり取得したりするための任意の機能のことです。イベント・チャンネルの例としては、次のものがあります。

- 通常、テキスト・フォーマットでメッセージが格納されている、ログ・ファイル
- イベント管理システム
- ステータス情報のスナップショットを取得するために実行されるプログラム

イベント管理システムはアクティブなイベント・チャンネルであり、イベント情報の配布、格納、および取り出しのサービスを提供します。

Tru64 UNIX では、システム・コンポーネントがイベントと状態に関する情報を報告するために使用できるチャンネルを多数サポートしています。各チャンネルで得られる情報を定期的に検査し、システムが正常に動作していることを確かめる必要があります。システム・ロガー `syslog` とバイナリ・エラー・ロガー `binlog` は、イベント管理システムの身近な例です。これらのイベント管理システムでは、他のコンポーネントから使用できるように簡単なイベント配布機能を用意しています。またそのデーモンは、受け取ったイベント情報をアクティブに管理します。

逆に、`cron` デーモンのログ・ファイル `/var/adm/cron/log` は、受動的なイベント・チャンネルの例です。`cron` デーモンは新しいイベント情報をそのデーモン用のファイルの末尾に書き込み、書き込み時に関連エンティティに通知する動作は特に行いません。

`syslog` および `binlog` の他にも、システム上のさまざまな場所にさまざまなログ・ファイルが格納されています。これらのログ・ファイルの管理を容易にするために、Event Manager は、すべてのソースからのイベントを1つのイベント・ストリームに結合することで、複数のイベント・チャンネルを1

箇所で監視できるようにします。システム管理者は、結合されたストリームをリアルタイムで見ることできるし、記憶域から取り出したイベントの履歴として見ることもできます。Event Manager の表示機構には、グラフィカル・イベント・ビューアがあり、SysMan Menu と SysMan Station に統合されています。また、コマンド行ユーティリティのフル・セットが用意され、必要に応じてイベントをフィルタリングしたり、ソートしたり、フォーマットすることができます。選択した状況を自動的にユーザ (または他システム・エンティティ) へ通知するように、Event Manager を構成できます。

Event Manager は、syslog や binlog など置き換えるのではなく、これらをカプセル化します。これらの既存のチャンネルは同じ場所にあり、いままでと同じイベント・セットを扱っています。Event Manager では、他のチャンネルがよりアクセスしやすくなります。

13.1.1 Event Manager の機能

Event Manager には、次のような機能があります。

- ユーザやアプリケーションが、イベントをポストし監視する機能
- syslog や binlog など、他のイベント・チャンネルのサポート
- カスタム・イベント・チャンネルのカプセル化のサポート
- DECevent および Compaq Analyze との統合によるバイナリ・エラー・ログ・イベントの変換のサポート

Compaq Analyze は規則ベースのハードウェア障害管理診断ツールで、エラー・イベントの解析機能と変換機能を備えています。

- SysMan アプリケーション群とグラフィカル・イベント・ビューアの統合
- イベント・データの要約か、詳細かの選択 (オンラインの説明を含む)
- シェル・スクリプトまたはコマンド行からイベントのポストと処理に使用可能なコマンド行ユーティリティのフルセット
- どのイベントを記録し、同一のイベントが使用する記憶領域を最適化するかを完全に制御できる、構成可能なイベント・ロガー
- 選択したイベントを他システム・エンティティに自動的に通知できる、構成可能なイベント転送
- ログ・ファイルのアーカイブおよびパージを毎日自動的に行う、ログ・ファイル管理

- アプリケーション・プログラミング・インタフェース (API) ライブラリのサポート
- イベント情報アクセスの集中化
- イベントのポストまたはアクセスの構成可能な権限

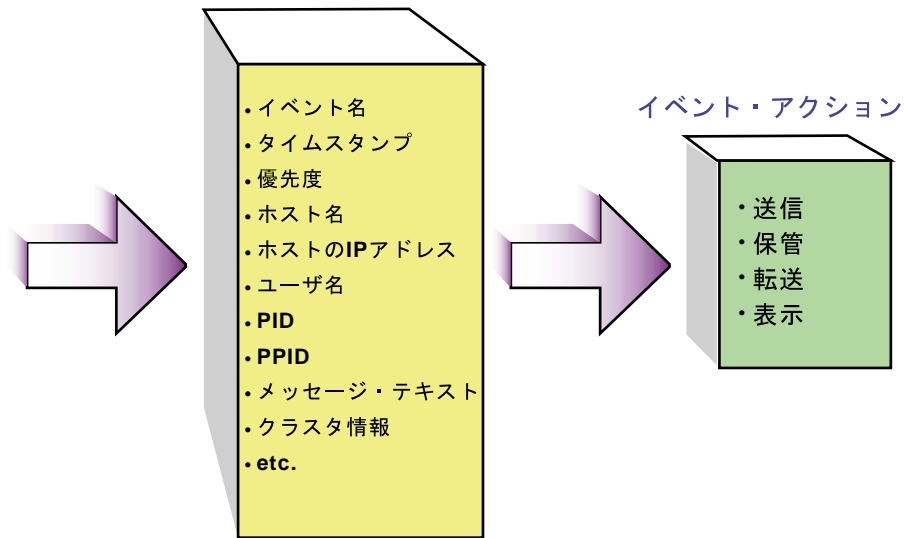
13.1.2 Event Manager イベントとは

Event Manager イベントは、バイナリ・データのパッケージであり、イベントの名前、タイムスタンプ、ポスタ (ポストする側のクライアント) についての情報などの標準データ項目が入っています。イベントには、ポスタが用意する可変データが入っていることがあります。たとえば、デバイスの障害を通知するイベントには、デバイスのパス名と型を含む変数が入っています。イベントは、Event Manager のポスト側クライアントにより生成されてポストされ、Event Manager デーモンによって他のクライアントに配布されます。受信したプロセスは、その後、イベントに含まれる情報を抽出したり処理したりできます。

Event Manager ロガーは、ポストされたイベントを捉え、システム・ログ・ファイルに格納しますが、ユーザは、自身のイベントのセットを捉えて自身のファイルに格納し後で分析に使うことも、容易にできます。evmwatch モニタリング・ユーティリティを使用するか、ロガーを再構成して、自身のイベントを捉えるようにします。

図 13-1では、イベントをモデル化して示します。「イベント内容」の箱がイベントに含まれている項目を示しています。この項目には、プロセス識別子 (PID)、イベントが生成され、イベントに含まれているホスト・システムの名前などがあります。「イベント・アクション」の箱は、イベントに対して実行される可能性があるアクションを示しています。

図 13-1: イベント・モデル
イベント内容



ZK-1549U-AIJ

Event Manager には、イベントのフォーマットを理解するコマンド行ユーティリティがあります。これを使って、コマンド・プロンプトやシェル・スクリプトから、基本的な操作を行うことができます。イベントはバイナリ・データのパッケージであるため、テキスト・ビューア (more コマンドなど) で直接見ることはできません。Event Manager コマンドを使用すると、次の操作が可能です。

- 記憶域からイベントを取り出し、好みの順序でソートし、表示形式に変換する。
- 新しいイベントがポストされないか待機する。
- 新しいイベントをポストする。

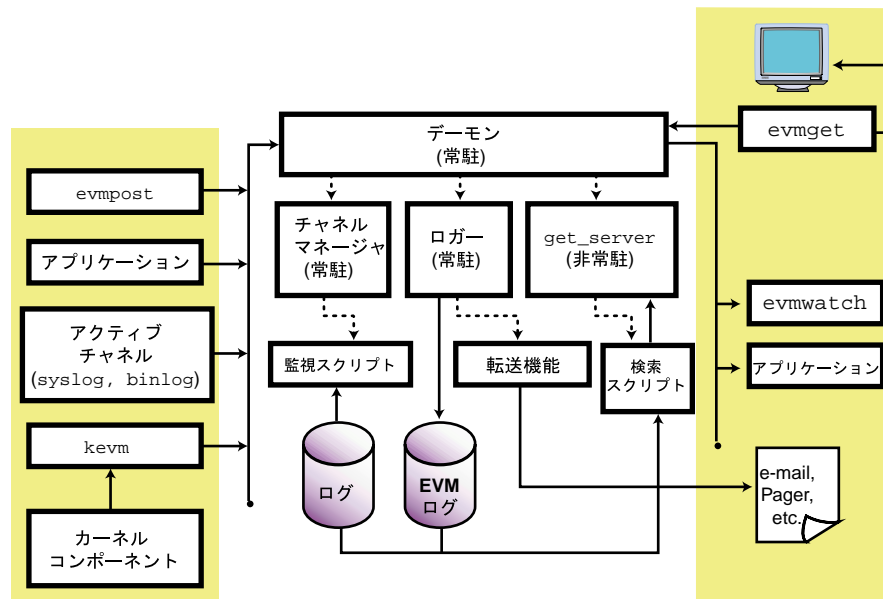
これらの Event Manager コマンド行ユーティリティは、パイプで結合して使えるようになっています。たとえば、ファイルからのイベントのセットをソート・ユーティリティにパイプし、その出力結果をフォーマット・ユーティリティにパイプし、その出力結果を more コマンドにパイプしたり、またはファイルにリダイレクトすることができます。13.3 節では、Event Manager コマンドを使用してイベント動作の監視および表示を行う例について説明します。

イベント・ファイルをテキスト形式に変換した後は、他の標準的なユーティリティを用いて分析することができます。たとえば、イベント名だけを表示させ、その表示を `sort -u` コマンドと `wc -l` コマンドにパイプして、ファイル内に異なるタイプのイベントがいくつあるかを調べることができます。

13.1.3 Event Manager のコンポーネント

この項では、Event Manager の各コンポーネントがどのようにやり取りするかを示します。また、Event Manager を実行するのに用いるシステム・ファイルと、通常の動作で Event Manager が生成するファイルについても説明します。システムのモデルを図 13-2 に示します。

図 13-2: Event Manager コンポーネントのモデル



ZK-1371U-AIJ

図 13-2 では、イベントをポストする側のクライアントのコンポーネントが左、Event Manager のシステム・コンポーネントが中央、イベントのサブスクライブおよび取り出しを行う側のクライアントのコンポーネントが右に示してあります。アクティブ・イベント・チャンネルは、Event Manager に直接イベントをポストします。パッシブ・イベント・チャンネルとは、イベントをポストせず、ポーリングによって情報が収集されるチャンネルのことです。これらのチャンネルは、監視スクリプトによって処理されるログ・ファイルに記録されます。

Event Manager の主要なコンポーネントは、`evmd` デーモンです。これは、システムがブートされ実行レベル 2 になるときに初期化されます。実行レベルについては、第 2 章 を参照してください。イベント管理がシステムのスタートアップ時に機能するように、デーモンとその子プロセスの初期化は次のように同期がとられます。

- システムのブート時にカーネルのコンポーネントのいくつかは、初期化シーケンスの一環としてイベントをポストします。Event Manager デーモンはまだ動作していないので、これらのイベントはデーモンが受け入れ可能な状態になるまでカーネルのメモリのキューにつながれます。
- Event Manager デーモンは、システム・スタートアップの実行レベル 2 の初期化シーケンスの早い段階で開始されます。システムの実行レベルについては、第 3 章 を参照してください。デーモンは、次に以下の処理を行います。
 - ロガーを起動する
 - チャネル・マネージャを開始する
 - クライアントからの接続要求をリッスンする
- ロガーが、リッスンする接続を確立しイベントを記録する準備ができると、デーモンは、カーネル・レベルとユーザ・レベルのポスタからポストされたイベントの受け入れを開始します。

Event Manager ロガー・プログラム `evmlogger` は、常駐プロセスとして動作します。このロガーは、選択した一連のイベントをサブスクライブし、後で取り出すことができるようにそれを管理対象のログ・ファイルに格納するように構成されています。ロガーはまた、省略時の動作として次のように構成されています。

- 高優先順位イベントをシステム・コンソールに表示する。
- 高優先順位イベントが発生すると、システム管理者宛てにメールを送る。

Event Manager ロガー `evmlogger` は他のシステム・コンポーネントの操作で必要とする必須のシステム・コンポーネントなので、システムの構成から外すべきではありません。

常駐のチャネル・マネージャ・プロセス `evmchmgr` は、チャネル内で注目すべき動作を検出した場合にイベントをポストする、チャネル監視スクリプト

を定期的に行うように構成されています。チャンネル・マネージャは、日常のログ・クリーンアップ機能も実行します。

get サーバ・プロセス `evmget_srv` は、非常駐 (デマンド) プロセスであり、さまざまなイベント・チャンネルに対してイベント検索スクリプトを実行します。ユーザが `evmget` コマンドを実行したときは必ず、`evmd` デーモンが `evmget_srv` のインスタンスを起動します。

モデルの左側のエンティティは、イベントをポストするために、デーモンへのポストのための接続 (posting connection) を確立します。ポスト側からイベントを受け取ると、デーモンは、テンプレート・データベースの対応するイベント・テンプレートとそれらをマージし、サブスクライブしているクライアントに配布します。

モデルの右側では、以下の動作が行われます。

- `evmwatch` とその他のアプリケーション・プログラムは、イベントが生成されたときにそのイベント情報を受け取る必要があり、デーモンへのサブスクライブを行うための接続 (subscribing connection) を確立し、イベントの通知要求を指定するフィルタ文字列 (filter string) をデーモンに渡します。
- `evmget` コマンドは、各種ログ・ファイルから履歴イベント情報を取得するためにユーザが実行することができ、サービスとの接続 (service connection) を確立して、検索するイベントのセットを指定するフィルタ文字列を渡します。次にデーモンは、`get` サーバのインスタンスを起動して要求を処理します。
- 電子メールおよびページャ (呼び出し) のアクションは、コマンドの転送の例であり、特定のイベントの生成に応答してロガーが実行します。

13.1.3.1 Event Manager のコマンド行ユーティリティ

Event Manager のコマンド行ユーティリティには、Event Manager システム自身の管理用のものと、イベントのポストおよび取得で使用するものがあります。表 13-1 では、一般ユーザ用のコマンドについて説明します。詳細は、それぞれのリファレンス・ページを参照してください。これらのコマンドを使用してイベント動作を監視して見直す方法の例については、13.3 節を参照してください。

表 13-1: Event Manager のコマンド行ユーティリティ

| コマンド | 説明 |
|----------|--|
| evmget | 構成済のログ・ファイルのセットとイベント・チャンネルから、格納されているイベントを、チャンネル固有の検索関数を使って取り出します。 |
| evmpost | テキスト・イベント・ソースのファイルまたはストリームを受け取り、Event Manager デーモンにポストします。デーモンはこれを配布します。 |
| evmshow | 1 つ以上の Event Manager イベントを受け取り、要求されたフォーマットで出力します。 |
| evmsort | イベントのストリームを読み、指定された基準に従ってソートします。 |
| evmwatch | 指定されたイベントをサブスクライブし、それらが到着すると出力します。 |

表 13-2では、通常、システムの初期化時に起動される、Event Manager 管理のためのコマンドを示します。個々のコマンドのリファレンス・ページでは、コマンドを使用する具体的な状況について説明しています。

表 13-2: Event Manager の管理ユーティリティ

| コマンド | 説明 |
|----------|---|
| evmchmgr | Event Manager チャンネル・マネージャは、Event Manager デーモンによって自動的に起動されます。これは、個々のチャンネルに定義された関数を定期的に実行します。 |
| evmd | <p>Event Manager デーモンは、ポスト側クライアントからイベントを受け取り、サブスクライブ側のクライアントにそれらを配布します。サブスクライブ側のクライアントとは、イベントの受信を希望していることを表明しているクライアントです。</p> <p>このデーモンは、システムの重要な機能であり、システムのブート時に自動的に起動されます。このデーモンは終了させないでください。</p> <p>Essential Services Monitor (ESM) デーモン esmd は、evmd を含む重要なシステム・デーモンを自動的に再起動することにより、その可用性を維持しています。ESM デーモンの詳細は、esmd(8) を参照してください。</p> |

表 13-2: Event Manager の管理ユーティリティ (続き)

| コマンド | 説明 |
|-----------|---|
| evmlogger | Event Manager ロガーは、Event Manager デーモンによって自動的に起動されます。ロガーは、デーモンからイベントを受け取り、そのイベントをフィルタ文字列が一致するログにそれぞれ記録します。evmlogger はまた、要求があれば動作を実行するように構成できる、イベント転送エージェントとして使うこともできます。 |
| evmreload | このコマンドは、Event Manager のコンポーネントに構成ファイルを再ロードするように指示する制御イベントをポストします。Event Manager 構成ファイルを変更した場合は、このコマンドを使用して新しい構成をロードしなければなりません。 |
| evmstart | このコマンドは、Event Manager デーモンを起動します。このコマンドは、システム・スタートアップ・スクリプト内で実行することを意図したものですが、何らかの理由で Event Manager が停止した場合には、再起動するために使うこともできます。 通常の運用では、esmd デーモンは、Event Manager デーモンを自動的に再起動します。 |
| evmstop | このコマンドで、Event Manager デーモンが停止し、エンティティはイベントのポストやサブスクライブができなくなります。このコマンドは、システム・シャットダウン・スクリプト内で使用することを意図しています。Event Manager は、多くのシステム関数が適切に動作するために必要です。このため、このコマンドは正常な運用状態では使用しないでください。 多くの場合、esmd デーモンは、Event Manager デーモンを自動的に再起動します。 |

13.1.3.2 Event Manager のアプリケーション・プログラミング・インタフェース

Event Manager API ライブラリ libevm.so には、幅広いイベント管理関数群があります。このライブラリを使用すると、プログラマは、Event Manager とのインタフェースを持つプログラムを設計できます。API 関数を使用すると、プログラムからイベントをポストしたり、デーモンに要求や通知を送信したり、デーモンから応答や情報を受信することができます。こ

これらのインタフェースの使用については、『プログラミング・ガイド』を参照してください。(個々の API リファレンス・ページのリストについては、EVM(5) を参照してください。)

13.1.3.3 Event Manager のシステム・ファイル

Event Manager は、以下のシステム・ファイルを作成または使用します。システム・ファイルには、実行可能ファイル、構成ファイル、およびログ・ファイルがあります。

実行可能ファイル

Event Manager 管理コマンドの実行可能ファイルは、`/usr/sbin` ディレクトリにあります。

一般的なコマンド (つまり、ユーザ・コマンド) の実行可能ファイルは、`/usr/bin` ディレクトリにあります。

初期化ファイルは、`/sbin/init.d` ディレクトリにあります。

構成ファイル

ベースとなる Event Manager 構成ファイルは、`/etc` ディレクトリにあります。これらのファイルについて、以下に示します。

`/etc/evmdaemon.conf`

このファイルは、Event Manager の構成と起動で使用されるコマンドを記述したテキスト・ファイルです。このファイルの詳しい説明は、13.2.2.1 項と `evmdaemon.conf(4)` を参照してください。

`/etc/evmchannel.conf`

チャンネル・マネージャ `evmchmgr` と `evmshow` コマンドが読むイベント・チャンネル構成ファイルです。このファイルには、イベントをポストおよび取り出すことができるすべてのチャンネルが記述されています。このファイルの詳しい説明は、13.2.2.2 項と `evmchannel.conf(4)` を参照してください。

`/etc/evmlogger.conf`

このファイルは、ロガー (`evmlogger`) の構成ファイルです。イベントの表示、転送、保存に関するコマンドが入っています。このファイルの詳しい説明は、13.2.2.3 項と `evmlogger.conf(4)` を参照してください。

/etc/evm.auth

このファイルは、イベントとイベント・サービスへのアクセスを制御するために使用します。このファイルの詳しい説明は、13.2.3.2 項と evm.auth(4) を参照してください。

ログ・ファイル，作業ファイル，およびローカル・インストレーション・ファイル

ログ・ファイル，作業ファイル，およびローカル・インストレーション・ファイルは、/var/evm の下の以下のサブディレクトリにあります。

/var/evm/sockets

この CDSL ディレクトリには、ドメイン・ソケット・ノード evmd と、関連するロック・ファイル evmd.lock があります。ローカル・クライアントはこのソケットを使って接続します。

/var/evm/evmlog

この CDSL ディレクトリには、省略時の Event Manager ロガー構成によって作成されたイベント・ログがあります。このディレクトリのログ・ファイルには、evmlog.yyyymmdd[_nn] という形式の名前がつけます。yyymmdd はログの日付、_nn は連続の生成番号です。ログのサイズが、設定されている 1 日の最大サイズに達した場合、またはロガーが現在のファイルにエラーを検出した場合、新しいログ・ファイルが作成されます。1 日の最初のログ・ファイルには生成番号は付きません。新しいログ・ファイルは、システム時間の午前 0 時を過ぎて最初にイベントを受け取った時点で作成されます。

このディレクトリにはまた、ロック・ファイル (evmlog.dated.lock) と、生成制御ファイル (evmlog.dated.gen) が置かれます。後者には、現在の生成番号に関する情報が記録されます。ログ・ファイルの管理については、13.2.4 項を参照してください。

/var/evm/adm/logfiles

この CDSL ディレクトリには、Event Manager の常駐のコンポーネントであるデーモン、ロガー、およびチャネル・マネージャが作成した出力メッセージ・ログが置かれます。Event Manager が開始されるたびに新しいファイルが作成されます。その際、古いファイルはファイル名に .old という接尾語を付加して名前が変更されます。古

いファイルがすでにある場合は、置き換えられます。これらのメッセージ・ログは、Event Manager の `misclog` イベント・チャンネルによってカプセル化され、その内容は、`evmget` とイベント・ビューアを通して見るすることができます。

`/var/evm/shared`

このディレクトリは、クライアントの認証に必要な一時ファイルを置く作業ディレクトリです。

`/var/evm/adm/templates`

このディレクトリは、ローカルまたはサードパーティのイベント・テンプレートのサブディレクトリをインストールするためのものです。このディレクトリは、シンボリック・リンクによってシステムのテンプレート・ディレクトリに結び付けられています。

`/var/evm/adm/channels`

このディレクトリには、ローカルおよびサードパーティのイベント・チャンネル・スクリプトがインストールされます。

`/var/evm/config`

このディレクトリとそのサブディレクトリには、各種の Event Manager 構成要素の 2 次構成ファイルがあります。本リリースで 2 次構成ファイルをサポートしているのは、ロガーだけです。詳細は、`evmlogger.conf(4)` を参照してください。

`/var/evm/adm/filters`

このディレクトリには、ローカルおよびサードパーティのイベント・フィルタ・ファイルがインストールされます。

`/var/run/evmd.pid`

このファイルには、デーモンのプロセス ID (PID) が記録されます。これは、`evmd` デーモンによって保存され、Event Manager を停止する場合などに利用されます。

`/var/run/evmlogger.info`

このファイルには、ロガーの PID と管理しているログ・ファイルの情報が記録されます。evmlog チャンネルの検索関数と日次クリーンアップ関数で、この情報を使用します。

システム提供の定義ファイル

システム提供のテンプレート、チャンネル、フィルタの定義ファイルが、`/usr/share/evm` ディレクトリの下の次のサブディレクトリにあります。

これらのファイルは変更しないでください。

`/usr/share/evm/channels`

このディレクトリには、binlog、syslog、および evmlog などのシステム提供のイベント・チャンネルのサブディレクトリがあります。各サブディレクトリには、そのチャンネルで利用できるサービスを定義するスクリプトが置かれています。

`/usr/share/evm/filters`

このディレクトリには、システムのフィルタ・ファイルがあります。

`/usr/share/evm/templates`

このディレクトリには、システムのイベント・テンプレート・ファイルおよびサブディレクトリがあります。

13.1.4 関連ユーティリティ

次のようなサブシステム、またはオプションのコンポーネントも、イベント処理機能を提供します。

システム・ロガー (syslogd)

システム・ロガー (syslogd)

システム・ロガーは、カーネル・レベルとユーザ・レベルのシステム・コンポーネントのためのテキスト・メッセージを記録します。自分自身のログ・ファイルにイベントを記録するだけでなく、syslogd デモンの省略時の構成では、選択されたイベントを Event Manager に転送します。そして、さらにこれを保管したり、配布したりすることができます。Event Manager では、syslog イベントを evmlog ファイルに格納することで、非常に大きいテキスト・ファイルからの取り

出しによるオーバーヘッドを軽減します。詳細は、syslogd(8) を参照してください。

バイナリ・エラー・ロガー (binlogd)

バイナリ・エラー・ロガーは、システム・エラーと構成情報をバイナリ形式で記録します。イベントは、システムのタイプによって、DECevent 変換機能 (dia) または Compaq Analyze (ca) により変換されます。binlogd デーモンは、自身のログ・ファイルにイベントを記録し、自身のクライアントにそれを配布するほか、イベントを Event Manager に転送します。Event Manager ではさらにこれを配布します。Event Manager では、バイナリ・エラー・ログ・イベントを binlog イベント・チャンネル関数を用いて記憶域から取り出します。詳細は、binlogd(8) を参照してください。

DECevent と Compaq Analyze

DECevent は、規則ベースの変換およびレポートを行うユーティリティであり、イベントをバイナリ・エラー・ログ・イベントに変換します。Event Manager は、DECevent の変換機能 dia を使って、バイナリ・エラー・ログ・イベントを表示できる形式に変換します。Compaq Analyze は、ほとんどの EV6 シリーズのプロセッサ上で同様の役割を果たします。詳細は、ca(8) と、Compaq Analyze のその他のドキュメントを参照してください。

13.2 Event Manager の管理

Event Manager が動作するシステムでの管理者の役割には、主に次のものがあります。

- Event Manager の起動と停止。 13.2.1 項で説明します。
- Event Manager の構成。 13.2.2 項で説明します。
- 誰がイベントをポストまたはアクセスできるかの制御。 13.2.3 項で説明します。
- ログ・ファイルの管理。 13.2.4 項で説明します。
- 他のシステム・ユーザのためのイベント・レポート機能。 13.2.5 項で説明します。

- Event Manager の機能を使用する新しい製品のインストール。 13.2.6 項で説明します。

Event Manager の使用についての詳細は、13.3 節を参照してください。

13.2.1 Event Manager の起動と停止

Event Manager はシステム起動時に自動的に起動され、システムのシャットダウンとともに停止します。

Essential Services Monitor (ESM) デーモン `esmd` は、Event Manager デーモンを含む重要なシステム・デーモンを自動的に再起動することにより、その可用性を維持しています。詳細は、`esmd(8)` を参照してください。

Event Manager を停止するには、ESM デーモンのプロセス識別子が必要です。Event Manager を停止するには、以下の手順を実行します。

1. ESM デーモンのプロセス識別子を調べる。

```
# ps -aef | grep esmd | grep -v grep
1. root    48  1  0.0  Apr 22  ??  0:00.09  /usr/sbin/esmd
```

この例では、PID は 48 です。

2. `kill` コマンドを使用して、ESM デーモンを停止します。

```
# kill -STOP PID
```

3. `evmstop` コマンドを使用して、Event Manager を停止します。

```
# /usr/sbin/evmstop
```

Event Manager と ESM を起動するには、以下の手順を実行します。

1. `evmstart` コマンドを使用して、Event Manager を起動します。

```
# /usr/sbin/evmstart
```

2. `kill` コマンドを使用して、ESM デーモンの実行を元に戻します。ESM デーモンを停止させたときに指定した `PID` を使用します。

```
# kill -CONT PID
```

Event Manager 構成を変更しても、Event Manager を停止して再起動する必要はありません。この場合、構成を変更した後で `evmreload` コマンドを実行します。詳細は、`evmreload(8)` を参照してください。

13.2.2 Event Manager の構成

Event Manager を構成することは、構成可能な常駐コンポーネントを確立し、保守することを意味します。この常駐コンポーネントには、次のものがあります。

- Event Manager デーモン `evmd`
- チャネル・マネージャ `evmchmgr`
- ロガー `evmlogger`

各コンポーネントは、動作を指示する構成ファイルを参照します。

オペレーティング・システムをインストールする際に、Event Manager は、たいていのシステムに適合する省略時の構成オプションで実行するように自動的に構成されます。ただし、たとえば以下の場合には構成ファイルを変更できます。

- イベント・チャネルを追加したり変更した場合
- ログ・ファイルのアーカイブや削除のオプションを変更する必要がある場合
- 別のログ・ディレクトリを作成した場合
- Event Manager 機能へのリモート・アクセスを有効にした場合

Event Manager は、バイナリ・ロガー (`binlogd`) イベントを変換するために DECEvent と Compaq Analyze を使用するように事前に構成されています。

構成を変更した場合は必ず、新しいファイルをロードするため、または変更を行うため、`evmreload` コマンドを実行して、構成を再度確立しなければなりません。このコマンドについての詳細は、`evmreload(8)` を参照してください。

構成ファイルについては、以下の項と該当するリファレンス・ページで説明します。

13.2.2.1 Event Manager のデーモンの構成

Event Manager デーモンは、`/etc/evmdaemon.conf` 構成ファイルを、システムのスタートアップ時と、`evmreload` コマンドを使用して再ロード要求が出されるたびに読み取ります。構成ファイルの内容と構文についての詳細は、

evmdaemon.conf(4) リファレンス・ページを参照してください。例 13-1では、Event Manager デーモン構成ファイルのエントリの例を示します。

例 13-1: Event Manager デーモン構成ファイルのエントリの例

```
# Event template directory:
sourcedir      "/usr/share/evm/templates" [1]

# Start the Event Manager Logger [2]
start_sync "/usr/sbin/evmlogger -o /var/run/evmlogger.info \
-l /var/evm/adm/logfiles/evmlogger.log"
# Start the Event Manager Channel Manager [2]
start_sync "/usr/sbin/evmchmgr -l \
/var/evm/adm/logfiles/evmchmgr.log"

# Event retrieval service definition: [3]
service
{
    name          event_get
    command       "/usr/sbin/evmget_srv"
}

# Set up an activity monitor. [4]
activity_monitor
{
    name          event_count
    period        10
    threshold     500
    holdoff       240
}

remote_connection false [5]
```

- [1] この文は、すべてのイベント・テンプレート・ファイルのディレクトリ階層の先頭を示します。
- [2] これらのコマンドでは、evmlogger コンポーネントと evmchmgr コンポーネントを2つのクライアントとして同期をとって開始しています。これにより、両方のクライアントがサブスクライブ要求を完了してから、デーモンがポスト側クライアントからのイベントを受け入れることを保証します。これらのコマンドのコマンド行オプションには、クライアントのログ・ファイルを指定します。ロガーの場合は、出力ファイルも指定します。この出力ファイルは、イベント・チャネル関数 evmlog から動作の詳細を利用できるようにするために使用されます。
- [3] これらの文は、イベント検索サービス event_get を定義しています。evmget コマンドはこれを使ってイベントを取り出します。

- ④ これらの文は、アクティビティ・モニタを定義します。この例では、10 分間に 500 以上のイベントを受け取った場合、デーモンはシステム管理者の注意を促すために高優先順位のイベントをポストします。その後、動作モニタリング(イベントのカウンティング)は、保留 (hold-off) 期間である 4 時間 (240 分) の間中断されます。
- ⑤ この行では、remote_connection を false にセットし、リモート Event Manager クライアントがこのシステムに接続できないようにしています。このセキュリティ関連の値を変更する方法についての詳細は、evmdaemon.conf(4) と 13.2.3 項を参照してください。

構成ファイルを変更した場合は、Event Manager デーモンにこの変更を認識させるために、evmreload コマンドを実行しなければなりません。詳細は、evmreload(8) を参照してください。

13.2.2.2 Event Manager のチャネルの構成

イベント・チャネルは、イベント情報のソースです。チャネル構成ファイル /etc/evmchannel.conf では、チャネル・マネージャ、evmshow コマンド、およびイベント検索プロセスが使用する、一連のイベント・チャネルとそこで動作する関数を定義しています。チャネル構成ファイルの内容と構文についての詳細は、evmchannel.conf(4) リファレンス・ページを参照してください。例 13-2 では、チャネル構成ファイル・エントリの例を示します。

例 13-2: Event Manager チャネル構成ファイルの例

```
# Global path for channel functions
path    /usr/share/evm/channels    ①

# Time-of-day at which daily cleanup function will run
cleanup_time 02:00:00    ②

# =====
# Event channel:  EVM log
# =====
channel
{
    ③
    name        evmlog    ④
    path        /usr/share/evm/channels/evmlog    ⑤
    events      *    ⑥
    fn_get      "evmlog_get"    ⑦
    fn_details  "evmlog_details"
    fn_explain  "evmlog_explain"
    fn_monitor  "evmlog_mon"
```

例 13-2: Event Manager チャンネル構成ファイルの例 (続き)

```
fn_cleanup "evmlog_cleanup 7 31" 8  
mon_period 15:00 # Monitor every 15 minutes 9  
}
```

- ❶ この行では、`/usr/share/evm/channels` ディレクトリを、すべての channel 関数の省略時設定のパスとして宣言しています。このファイルで定義されている channel 関数の名前でスラッシュ (/) で始まらないものには、そのチャンネル・グループに個別のパスが明記されていない場合、名前の前にこのパスがつけられます。
- ❷ この行では、すべてのチャンネルを毎日午前 2:00 にクリーンアップすることを指示しています。
- ❸ この行では、イベント・チャンネルを定義する構成グループを指定しています。
- ❹ この行では、チャンネルの名前として `evmlog` を指定しています。
- ❺ この行では、グローバル・レベルで定義された省略時のパス `/usr/share/evm/channels` を指定変更します。
- ❻ この行のアスタリスク (*) は、このチャンネルが省略時のイベント処理を行うことを示します。つまりこの関数は、どのチャンネルのイベント値にも一致しない名前を持つすべてのイベントのために、詳細と説明を提供するために呼び出されます。
- ❼ `fn_` で始まる行はすべて、各関数に対して実行されるスクリプトを定義します。
- ❽ この行の引数の値は、クリーンアップ・プログラムに渡され、その操作を制御します。この例では、7 日を超えて保持されたログ・ファイルを圧縮し、31 日を超えて保持されたこれらのログ・ファイルを削除するように指定しています。引数の意味は個々のチャンネル機能に固有であり、すべての場合に同じとは限りません。
- ❾ この行では、モニタリングの間隔を設定しています。この例では、`/usr/share/evm/channels/evmlog/evmlog_mon` 関数は、15 分ごとに呼び出されます。

13.2.2.3 Event Manager ロガーの構成

Event Manager ロガーは、`/etc/evmlogger.conf` 構成ファイルのエントリに従ってイベントの保存と転送を処理します。このファイルの内容と構文についての詳細は、`evmlogger.conf(4)` を参照してください。例 13-3 では、ロガーの構成ファイルのエントリの例を示します。ロガーに行うことが可能なカスタマイズの例としては、出力をログ・ファイルの他に、端末に行うことなどがあります。

例 13-3: Event Manager ロガー構成ファイルのエントリの例

```
# Main log file:
eventlog { 1
    name      evmlog 2
    logfile   /var/evm/evmlog/evmlog.dated 3
    type      binary 4
    maxsize   512 # Kbytes 5

    # Uncomment the following "alternate" line and set the
    # logfile path to specify an alternate logfile in case
    # of write failures.
    # The path must specify an existing directory.
    #alternate /your_alterate_fs/evmlog/evmlog.dated 6

    # Log all events with priority >= 200, except binlog events:
    filter     "[prio >= 200] & (! [name @SYS_VP@.binlog])" 7

    # Suppress logging of duplicate events:
    suppress 8
    {
        filter     "[name *]"
        period      30 # minutes
        threshold   3 # No. of duplicates before suppression
    }
}

# Forward details of high-priority events to root:
forward { 9
    name      priority_alert 10
    maxqueue  200 11

    # Don't forward mail events through mail
    filter     "[prio >= 600] & ![name @SYS_VP@.syslog.mail]" 12

    suppress 13
    {
        filter     "[name *]"
        period     120 # minutes
        threshold  1 # No. of duplicates before suppression
    }
}
```

例 13-3: Event Manager ロガー構成ファイルのエントリの例 (続き)

```
# This evmshow command writes a subject line as the first
# line of output, followed by a detailed display of the
# contents of the event.
# The resulting message is distributed by mail(1).
command "evmshow -d -t 'Subject: EVM ALERT [@priority]: @@' |
mail root" 14

# Limit the number of events that can be queued for this
# command:
maxqueue      100
}
# Secondary configuration files can be placed in the following
# directory. See the evmlogger.conf(5) reference page for
# information about secondary configuration files.
configdir      /var/evm/adm/config/logger
```

- ❶ この行から、イベント・ログ構成グループが始まります。
- ❷ この行でイベント・ログの name を指定しています。構成ファイルの他の部分は、この名前を参照します。
- ❸ この行では、ログ・ファイルを /var/evm/evmlog ディレクトリに格納するように指定しています。1日の最初のログが書き込まれるとき、dated 接尾語が yyyymmdd の形式の日付で置き換えられます。
- ❹ この行では、このログに書き込まれるイベントの type がバイナリ Event Manager イベントであり、フォーマットされた (ASCII テキスト) イベントではないことを指定します。
- ❺ この行では、ログ・ファイルの最大サイズを K バイト (KB) 単位で指定しています。この例の場合、現在のログ・ファイルのサイズが 512 KB を超えると、ロガーはこれをクローズし、続き番号の接尾語 (たとえば、_2) を用いて新しいログ・ファイルを開始します。
- ❻ この行のコメントアウト (#) を外し、サンプル・パスを実際の書き込み可能ディレクトリのパス名に置き換えると、主要なディレクトリでの書き込みができなくなった場合、このディレクトリに代替ログ・ファイルがオープンされます。
- ❼ この行では、イベントのフィルタリング条件を指定して、どのイベントがこのログ・ファイルに記録されるかを定義しています。Event

Manager フィルタの構文についての詳細は、`EvmFilter(5)` を参照してください。 `@SYS_VP@` エントリは、`sys.unix` が読み込まれたときに、そのファイルで置き換えられるマクロです。

- [8] この文では、イベント・ログの抑制パラメータを定義しています。この例の場合、イベントの抑制は、30 分以内に 3 つ以上の同じイベントを受け取ると開始されます。同じイベントの抑制により、ログ・ファイルのスペースが節約できます。イベントの抑制についての詳細は、`evmlogger.conf(4)` を参照してください。
- [9] この行では、`root` ユーザに転送するイベントの条件を定義しています。イベント転送プログラムは、選択されたイベントが発生すると指定されたコマンド文字列を実行します。これは、重大なエラーが発生したときにシステム管理者に知らせるのに役立ちます。
- [10] この行の `name` には、転送プログラムの名前を指定しています。
- [11] `maxqueue queue_limit` キーワードは、前のイベントが処理されている間に、転送プログラムがキューに入れることができるイベントの数を制限します。新たなイベントが到着したときに最大個のイベントが既にキューに入っている場合、この転送プログラムはこの新しいイベントを無視します。このキーワードが指定されていない場合、省略時の値は 100 イベントです。1000 イベントより大きい値を指定すると、ロガーは自動的に 1000 イベントに制限します。
- [12] この行では、イベントのフィルタ処理を定義しています。イベント・ログの定義と同様、フィルタ文字列は、この転送プログラムが処理する一連のイベントを指定します。メール・サブシステムでの問題を示す高優先順位のイベントをメールがポストした場合にイベント・ループが起きるのを避けるには、メール・イベントを、明示的にこの転送プログラムから除外します。
- [13] これらの行では、イベントの重複転送を抑制しています。転送プログラムの抑制メカニズムは、イベント・ログのものと似ています。ただし目的は、同じイベントが繰り返しポストされることにより、短い期間にそのコマンドが何度も送られるのを防ぐことです。この例の場合、イベントは最大で 2 時間に 1 度転送されます。
- [14] この行では、転送プログラムがイベントを処理したときに実行されるコマンドを定義しています。イベントは、このコマンドの `stdin` ス

トリームにパイプされます。このコマンドの結果は、コマンド行の上のコメントで説明しています。

ロガー構成ファイルを変更した場合は、ロガーにその変更を知らせるために `evmreload` コマンドを実行しなければなりません。詳細は、`evmreload(8)` を参照してください。

リモート・ロギング構成の詳細は、13.3.13.3 項を参照してください。

13.2.2.4 ロガーの 2 次構成ファイル

ロガーの 2 次構成ファイルを使用すると、1 次構成ファイル `/etc/evmlogger.conf` を変更せずにイベント・ログまたは転送プログラムを追加できます。この機能により、2 次ファイルの問題が 1 次構成に影響を及ぼすことはありません。したがって、異なるロガー構成を安全に試すことができます。ロガーが 2 次構成ファイル内の構文エラーを検出すると、エラー・メッセージが表示されてファイルが拒否されます。1 次構成ファイルと追加の (正しい) 2 次ファイルが処理され、Event Manager が正しく機能します。2 次構成ディレクトリ機能を使用すると、個々のシステム・コンポーネント、製品、およびアプリケーションで、ログ・ファイルと転送プログラムをインストールまたは変更することもできます。1 次構成ファイル内で行を挿入または修正するのではなく、ファイルをインストールまたは置き換えます。ファイルを削除することで、エントリをアンインストールすることができます。

2 次構成ファイルの省略時の推奨位置は、`/var/evm/adm/config/logger` ディレクトリまたはそのディレクトリのサブディレクトリです。それ以外の位置に構成ファイルを置き、省略時のディレクトリからシンボリック・リンクを作成することもできます。サポートされてはいますが、1 次構成ファイルに `configdir` 行を追加するのは避けてください。2 次構成ファイルのファイル名には、サフィックス `.conf` を付け、13.2.2.3 項で示す規則に従ったファイル構文でなければなりません。

ロガーの 2 次構成ファイルとディレクトリに適切なパーミッションを与えることが重要です。ロガーはスーパーユーザ特権で実行され、どの 2 次構成ファイルに指定されたコマンドでも実行できます。そのため、正しいパーミッションを持たない構成ファイルはすべてロガーで拒否され、警告イベントがポストされます。正しいファイル・パーミッションについては、`evmlogger.conf(4)` を参照してください。

クラスタ環境では、ロガー構成ファイルはすべてのクラスタ・メンバで共用されます。メンバ固有のイベント・ログまたはイベント転送プログラムが必要な場合は、2 次構成ファイルで指定することができます。2 次構成ディレクトリにコンテキスト依存シンボリック・リンク (CDSL) を作成し、ファイルを参照させます。CDSL の作成方法は、`mkcdsl(8)` を参照してください。

13.2.2.5 イベントの紛失を防ぐためのバッファ・サイズの変更

イベント紛失の問題が発生した場合は、システム・パラメータを変更して、受信バッファ・サイズを大きくすることができます。

受信バッファ・サイズには、システム・ソケット・バッファの省略時の最大値が設定されています。現在のこのパラメータの値を調べるために、次のコマンドを実行します。

```
# /sbin/sysconfig -q socket sb_max
```

このパラメータの実行時の値を変更するには、次のコマンドを実行します。

```
# /sbin/sysconfig -r socket sb_max=new-value
```

この変更は、次回のリブートまで有効で、新しい Event Manager 接続だけに効果があります。

この変更を恒久的なものにするには、`sysconfigdb` または `dxkerneltuner` ユーティリティを使用します。詳細は、`sysconfigdb(8)` または `dxkerneltuner(8)` を参照してください。

13.2.3 セキュリティについて

イベントを扱う場合、セキュリティは次のような理由により重要な問題となります。

- あるイベント情報へのアクセスが制御されないことにより、権限のないユーザにシステム運用の機密情報がもれてしまう場合があります。
- イベントをポストすることにより、アプリケーションのフェイルオーバーやシステムのシャットダウンなど、システムの運用にかかわる重大な動作が起こる場合があります。

従来、イベント情報に対するセキュリティは、ログ・ファイルへの読み取りアクセスを制限したり、特定のポスト動作を root ユーザだけに制限することで維持されています。Event Manager デーモンとイベント検索機能によって、すべてのイベントへのアクセスの代替の手段ができてしまうため、

デーモンはまた、イベントがポストされたときと記録された後の両方で、アクセスを制限する手段も提供します。これによって、権限のあるユーザだけがイベントを見ることができるようになります。このアクセスの制御は、権限機能 (authorization feature) と認証機構 (authentication technique) を通じて行います。

Event Manager 環境で使用する実行可能関数を作成する際にも、セキュリティへの配慮が必要です。チャンネル関数の保護については、『プログラミング・ガイド』を参照してください。

13.2.3.3 項で説明しているように、Event Manager は、指定されたユーザからはリモートでアクセスできます。

13.2.3.1 ユーザの認証

Event Manager デーモンは、接続要求を受け入れる前に、すべてのローカル・システム・ユーザの ID を認証します。クラスタの場合、同じクラスタの別のノードから接続を要求するユーザも認証します。リモート・ユーザの認証を含む、リモート接続については、13.2.3.3 項を参照してください。

13.2.3.2 ユーザの権限

イベントへのアクセスは、Event Manager の権限ファイル `/etc/evm.auth` によって制御されます。

root ユーザは、個々のユーザやユーザのグループに、次のことを行う権限を与えることができます。

- 選択したイベントのポスト
- 選択したイベントへのアクセス (サブスクライブまたは記憶域からの取り出し)
- 選択したサービスの実行

省略時の設定では、すべてのイベントがポストされます。イベント権限は、指定した権限を持っているユーザまたは明示的に権限が否定されているユーザのリストを各イベント・クラスごとに指定して、付与されます。ユーザのリストが後に続いていない正符号 (+) では、省略時の指定として、すべてのユーザに権限が与えられます。ユーザのリストが後に続いていない負符号 (-) では、省略時の指定として、すべてのユーザに権限が与えられません。root ユーザは、権限を明示的に否定されないかぎり、すべてのイベントをポスト

あるいはアクセスする権限を持ちます。例 13-4では、権限ファイルのエントリの例を示します。詳細は、`evm.auth(4)` を参照してください。

例 13-4: Event Manager 権限ファイルのエントリの例

```
# =====
#      EVENTS
# =====

event_rights {
    class      @SYS_VP@.evm.control    # EVM control events
    post       root
    access     +
}

event_rights {
    class      @SYS_VP@.evm.msg.admin  # EVM admin message
    post       root
    access     "root, group=adm"
}

event_rights {
    class      @SYS_VP@.evm.msg.user   # EVM user message
    post       +
    access     +
}

# =====
#      SERVICES
# =====

service_rights {
    service    event_get
    execute    +
}
```

- ❶ `sys.unix.evm.control` で始まる名前を持つクラスのイベントは、`root` ユーザだけがポストできます。このようなイベントは、すべてのユーザがアクセスできます。`@SYS_VP@` エントリは、ファイルが読み込まれるときに `sys.unix` に置き換えられるマクロです。
- ❷ `sys.unix.evm.msg.admin` で始まる名前を持つクラスのイベントは、`root` ユーザだけがポストできます。このようなイベントは、`root` と `admin` グループの全メンバがアクセスできます。

- ③ `sys.unix.evm.msg.user` で始まる名前を持つクラスのイベントは、すべてのユーザがポストやアクセスを行うことができます。
- ④ すべてのユーザが `event_get` サービスを実行できます。

権限ファイルを変更した場合は、Event Manager デーモンに変更を知らせるために `evmreload` コマンドを実行しなければなりません。

13.2.3.3 認証を含むリモート・アクセス

Event Manager では、リモート・システム上で実行されているクライアントにアクセスできます。このため、中央のシステムからイベントの監視および取り出しができます。このアクセスは、どのシステムからも可能にしたり、あるいは特定のシステムだけに制限することができます。また、リモート・システム上の個々のユーザごとに、イベントへのサブスクライブ、イベントのポスト、あるいはイベントの取り出しを行うためのパーミッションを設定できます。

`evmwatch`、`evmget`、および `evmpost` コマンド行ユーティリティに `-h` オプションを使用してホスト名または IP アドレスを指定すると、リモート・コネクションを確立できます。または、イベント・ビューアの「Get Events From...」ダイアログ・ボックスで、リモート・ホスト名を指定する方法でも確立できます。イベント・ビューアについては、13.3.11 項を参照してください。

リモート・アクセスを制御する 2 つのファイルがあります。
`/etc/evmdaemon.conf`

`remote_connection` の値が `true` の場合、リモート・アクセスは許可されます。この値が `false` (省略時の値) の場合、リモート・アクセスは拒否されます。

`/etc/evm.auth`

このファイルの `remote_host` の設定によって、どのリモート・システムにアクセスが許可されるか、そのリモート・システム上のどのユーザがアクセスを許可または拒否されるか、認証メソッドは、`callback` と `open` のどちらが使用されるかが制御されます。

リモート・ホスト，ユーザ，および認証の設定

/etc/evmdaemon.conf ファイルの remote_connection の値に true を設定した後で，リモート・ホスト・システムとユーザのアクセスを細かく設定します。この設定は，/etc/evm.auth ファイルの remote_host の設定によって行います。

remote_host の設定は，次のフォーマットで行います。

```
remote_host {
    host          "1 つ以上のホスト名のリスト"
    authentication 認証タイプ
    port          ポート番号
    users         "ユーザの指定"
}
```

これらのパラメータの順序は任意ですが，ホストの指定は，通常，最初に行います。これらのパラメータを，以下に説明します。

host

host パラメータには，ローカル・ホストの Event Manager にアクセスを許可する 1 つ以上のホストのリストを指定します。このリストのホストは，ホスト名でも IP アドレスでも，指定できますが，クラスタ別名は許されていません。ホストのリストを二重引用符で囲むと，スペース文字またはコンマで区切ることができます。二重引用符で囲まない場合は，ホスト名はコンマで区切ってください。次に示すのは，正しいホストのリスト例です。

```
host "hostA hostB hostC"
host "hostM, hostN"
host hostX,hostY,hostZ
```

authentication

この引数では，リストされたホストにアクセスする場合に使用される認証のタイプを指定します。2 つのタイプ evm_open と evm_callback があります。evm_open を指定すると，リモート・ホストへのオープン・アクセスが可能になります。evm_callback を指定すると，認証が行われ，その際にユーザを確認するためにリモート・ホストに連絡が行われます。

どちらの認証タイプも指定できます。ここで示す例の場合，リモート・ホストが evm_callback を使用できるかどうかをローカル・ホストが調べます。使用できる場合，evm_callback が使用されま

す。使用できない場合、`evm_open` が使用されます。例は、この項の末尾にある「例」にあります。

`port`

ポート番号では TCP/IP ポートを指定します。

`users`

この引数で、ローカル・ホストにアクセスできるリモート・システムのユーザと、そのローカル認証方法を指定します。ここでの指定方法にはいくつかの種類があります。最も単純な形式では、次のように、リモート・ユーザを個々に指定します。

```
users    adam
```

これは、リモート・ユーザ `adam` が、ローカル・ホスト上で、ローカル・ユーザ `adam` と同じ認証を受けることを意味します。

等号 (=) を使用すると、リモート・ユーザをローカル・ユーザにマップできます。リモート・ユーザ `remo` をローカル・ユーザ `lcal` にマップするには、次のように指定します。

```
users    remo=lcal
```

リモート・ユーザ名の前に、ハイフンすなわち負符号 (-) を指定すると、このユーザのアクセスは拒否することを意味します。

```
users    -eve
```

プラス記号 (+) は、すべてのユーザを意味します。次の例では、すべてのリモート・ユーザがローカル・ログイン `nobody` と同じ認証を受けることを意味します。

```
users    +=nobody
```

次の形式では、プラス記号によって、すべてのリモート・ユーザがローカル・システムで相当するユーザにマップされます。

```
users    +=+
```

これらのユーザ指定を組み合わせで使用できます。たとえば次の指定では、`root` は `root` 権限を持ち、`cain` を除くその他すべてのユーザは、ローカル・ユーザ `nobody` と同じ権限を持つことを意味します。リモート・ユーザ `cain` は、アクセスが拒否されます。

```
users    root
users    cain-
users    +=nobody
```

例

次の例では、システム systemA、systemB、および systemC は、コールバック認証によって、このコンピュータへのリモート・アクセスが許可されます。これらのリモート・システム上の各ユーザは、ローカル・システムの user1 にマップされます。

```
remote_host {
    host    "systemA, systemB,systemC"
    users   +=user1
    authentication  evm_callback
}
```

次の例では、システム systemA、systemB、および systemC は、コールバック認証によって、このコンピュータへのリモート・アクセスが許可されます。user2 と user3 を除く、これらのリモート・システム上の各ユーザは、ローカル・システムの user1 にマップされます。user2 と user3 は、明示的にアクセスを拒否されています。

```
remote_host {
    host    "systemA, systemB,systemC"
    users   +=user1
    users   -user2
    users   -user3
    authentication  evm_callback
}
```

次の例でも、システム systemA、systemB、および systemC は、コールバック認証によって、このコンピュータへのリモート・アクセスが許可されます。これらのリモート・システム上の各ユーザは、user2 を除いて、ホスト・システム上で相当するユーザにマップされます。user2 は、アクセスが拒否されます。

```
remote_host {
    host    "systemA, systemB,systemC"
    users   +=+
    users   -user2
    authentication  evm_callback
}
```

次の例では、すべてのシステムが、コールバック認証によって、このコンピュータへのリモート・アクセスが許可されます。root 以外のすべてのリモート・ユーザは、ローカル・システムの「special_user」にマップされますが、root は、アクセスが拒否されます。

```
remote_host {
    host +
    users +=special_user
    users -root
    authentication  evm_callback
}
```

次の例では、システム systemA と systemB は、オープン認証によって、リモート・アクセスが許可されます。すべてのユーザは、Event Manager にアクセスできます。

```
remote_host {
    host "systemA, systemB"
    authentication evm_open
}
```

次の例では、5 つのリモート・システムがローカル・ホスト上の Event Manager にアクセスできますが、アクセスできる範囲は異なります。リモート・システム julius, augustus, および caesar では、ユーザ root と adm だけが Event Manager にアクセスできます。これらのシステムのいずれかがこれらのユーザのための接続を試みる場合に、ローカル・システムの Event Manager は、これらのユーザが evm_callback 認証を使用できるかどうかを調べます。使用できれば、その認証を使用します。使用できなければ、evm_open を使用します。リモート・システム plato と socrates のすべてのユーザは、自分自身にマップされ、evm_callback 認証だけが使用されます。

```
remote_host {
    host "julius augustus caesar"
    users root
    users adm
    authentication evm_callback
    authentication evm_open
}

remote_host {
    host plato,socrates
    users +++
    authentication evm_callback
}
```

13.2.4 ログ・ファイルの管理

Event Manager チャンネル・マネージャ evmchmgr は、チャンネルの fn_cleanup 関数を使って、ログ管理機能を提供します。この機能は、チャンネル構成ファイル evmchannel.conf によってどのチャンネルにも定義することができます。このファイルについての詳細は、13.2.2.2 項を参照してください。

省略時の設定では、チャンネル・クリーンアップ関数は、Event Manager が起動するときに実行され、その後は毎日午前 2:00 に実行されます。チャンネル構成ファイルの cleanup_time 値を編集することで、この時間を変更することができます。クリーンアップがスケジュールされたら、チャンネル・マネー

ジャはイベント・チャンネルのリストをスキャンし、ファイルで指定された各チャンネルに対して、`fn_cleanup` コマンドを実行します。

`evmlog` のクリーンアップ関数 `evmlog_cleanup` には、次の 2 つの引数が指定できます。

- アーカイブの間隔。省略時の値は 7 日
- 削除の間隔。省略時の値は 31 日

この関数は、`find` ユーティリティを使ってアーカイブの間隔より古いログを見つけて圧縮 (`zip`) します。また、削除の間隔より古いアーカイブ・ファイルは削除します。これらの間隔の値は、チャンネル構成ファイルの関数定義を編集して、変更することができます。また、これらの値をゼロに設定すると、この関数は無効になります。

省略時のチャンネル構成ではまた、`misclog` イベント・チャンネルを通じて、SysMan Station のメッセージ・ログ・ファイル用の類似のクリーンアップ機能も提供します。`syslog` および `binary` のエラー・ログ・チャンネルは、`crontab` ファイル内のエントリを使用して管理することができます。パイナリ・エラー・ログ・ファイルは、通常日次ベースでは管理されず、チャンネルのクリーンアップ関数が、ログのサイズを報告する Event Manager イベントを毎日ポストします。ログが目立って大きくなった場合、ログ・エントリを調べてください。必要ならば `binlogd` のクリーンアップ・オプションを使って、クリーンアップを開始してください。詳細は、`binlogd(8)` を参照してください。

`evmget` コマンドは、アーカイブされた (`zip` された) ログに格納されている `evmlog` イベントは取り出しません。アーカイブされたログからイベントを取り出すには、まず `gunzip` コマンドでそれらを圧縮解除しなければなりません。アーカイブ・ファイルの圧縮解除については、`gunzip(1)` を参照してください。

13.2.5 イベント・テンプレート

イベント・テンプレートは、中央に保持されているイベントの記述です。イベント・テンプレートは、次の目的で使用します。

- Event Manager デーモンでイベントを登録することにより、イベントがポストされます。

- 情報を中央に集めて保管することにより、それぞれのアプリケーションに情報をハードコードする必要をなくす。

イベント・テンプレートの定義は、テンプレート・ファイルに保持されています。このファイルは、システムのテンプレート・ディレクトリ `/usr/share/evm/templates` の下の (またはこれにリンクされた) ディレクトリに格納されているテキスト・ファイルです。インストール固有または他社のイベント・テンプレートがある場合は、次の手順でロードします。

1. 適切な名前のサブディレクトリをローカル・テンプレート・ディレクトリ `/var/evm/adm/templates` に作成し、その中にイベント・テンプレートをコピーします。
2. `-d` オプションを指定して `evmreload` コマンドを実行し、Event Manager デーモンに内部テンプレート・データベースを再ロードするように指示します。

Event Manager が認識するには、テンプレート・ファイルに適切な所有権と許可が必要です。詳細は、`evmtemplate(4)` を参照してください。新しいイベント・テンプレート・ファイルのインストールについては、『プログラミング・ガイド』を参照してください。

イベントがポストされるたびに、Event Manager デーモンは、ポストされたイベントと名前の一致するテンプレート・イベントをその内部テンプレート・データベースで探します。次に、そのテンプレート・イベントに保持されている、中央に集められたデータ項目を取り出します。そしてこれらのデータ項目を、プログラムがイベントをポストしたときに渡した項目と結合します。これにより、イベントがマージされ、これがサブスクライブ側に配布されます。

13.2.6 新しい Event Manager クライアントのインストール

新しいアプリケーションをインストールし、その機能を使うために新しい管理スクリプトを開発する際に、新しいイベントをイベント・セットに追加することができます。イベントを追加する際には、Event Manager 構成ファイルと権限ファイルを変更した後、新しいテンプレートを追加する必要があります。構成ファイルについては、13.2.2 項を参照してください。新しいユーザの権限を変更する手順については、13.2.3.2 項を参照してください。

新しいイベント・テンプレートは、次の手順で追加します。

1. 13.2.5 項で説明したとおりに、新しいテンプレート・ファイルを作成します。
2. テンプレート・ファイルを `/var/evm/adm/templates` ディレクトリまたはサブディレクトリにコピーします。
3. `-d` オプションを指定して `evmreload` コマンドを実行し、Event Manager デーモンに内部テンプレート・データベースを再ロードするように指示します。

テンプレート・ファイルに必要な所有権と許可についての詳細は、`evmtemplate(4)` を参照してください。

Event Manager クライアント・アプリケーションの開発については、『プログラミング・ガイド』を参照してください。

13.2.7 binlog イベント変換ユーティリティの構成

イベント変換を行うユーティリティには、Compaq Analyze と DECevent の 2 つがありますが、新しいプロセッサでは DECevent がサポートされておらず、Compaq Analyze だけがサポートされています。

Compaq Analyze は、エラー・イベントの分析および変換を行う、規則ベースのハードウェア障害管理診断ツールです。Compaq Analyze の複数イベント相関分析機能により、システムのイベント・ログ・ファイルに格納されているイベントを分析したり、他のシステムからのイベントを分析することができます。ここでいう他のシステムには、他のオペレーティング・システム (OpenVMS や Windows NT など) も含まれます。

DECevent は、バイナリ・エラー・ログ・イベントのイベント変換を行う、規則ベースの変換/レポート・ユーティリティです。Event Manager は DECevent の変換機能 `dia` を使用して、バイナリ・エラー・ログ・イベントを可読形式に変換します。

Event Manager インフラストラクチャは Event Manager 形式のイベントのみを直接認識しますが、イベントは `binlogd` デーモンなどの他チャンネルを通してポストされます。これらのイベントは、下位レベルのイベントを可変データとして Event Manager イベントに挿入することで、ラッパー Event Manager イベントとして Event Manager に渡すことができます。その後、

パッケージ全体を Event Manager に渡すことができます。Event Manager は、可変部の内容やフォーマットは意識しません。

バイナリ・ロガー・デーモン binlogd は、この方法を使用して、自分のイベントを Event Manager を通して利用可能にします。binlogd デーモンはオペレーティング・システムからイベントを受信すると、そのイベントを最初に自分のログ・ファイルに格納し、クライアントに配布します。その後、binlogd は sys.unix.binlog という名前で始まる Event Manager イベントを作成し、binlogd イベント・データを含む binlog_event という名前の変数を追加します。最後に binlogd は、将来配布できるように、パッケージを Event Manager デーモンにポストします。Event Manager デーモンはパッケージを Event Manager イベントのように取り扱い、binlog_event 変数の内容については直接は認識しません。

コマンド行から evmshow -d コマンドを実行するか、イベント・ビューアのイベント要約ウィンドウの [詳細...] ボタンを選択してイベントの詳細表示を要求すると、Event Manager は、/etc/evmchannel.conf ファイル内でこのイベントに対して定義されている詳細表示プログラムを実行します。結果の表示は必ず、イベントの説明と、内容の詳細表示で始まります。イベントが binlogd イベントの場合、この表示の後に変換後の binlog_event 変数の内容が続きます。この変換情報は、システムの問題のトラブルシューティングに役立ちます。例 13-5 に、DECevent の変換を含む、binlogd イベントの詳細表示を示します。

例 13-5: DECevent 変換を表示する binlogd イベント

```
===== Binary Error Log event =====
Event Manager event name: sys.unix.binlog.op.shutdown
Binary error log events are posted through the binlogd
daemon, and stored in the binary error log file,
/var/adm/binary.errlog. This event is posted by the shutdown(8),
halt(8), and reboot(8) commands when the system is being shut
down. The message includes details of the user who initiated
the shutdown.
=====
Formatted Message:
    System shutdown msg:  System rebooted by root:
Event Data Items:
    Event Name      : sys.unix.binlog.op.shutdown
    Priority        : 200
    Timestamp       : 26-Jan-2000 20:54:36
    Host IP address  : 16.69.224.11
    Host Name       : kopper
```


例 13-5: DECEvent 変換を表示する binlogd イベント (続き)

```
Format          : System shutdown msg: $message
Reference       : cat:evmexp.cat:300
Variable Items:
  subid_class = 301
  message = "System rebooted by root:"
  binlog_event = [OPAQUE VALUE: 96 bytes]
===== Translation =====
DECEvent version: V3.2

Logging OS                2. operating system
System Architecture       2. Alpha
Event sequence number     752.
Timestamp of occurrence   26-JAN-2000 20:54:36
Host name                 kopper
System type register      x0000000F AlphaStation 600 or 500
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000
Event validity            1. O/S claims event is valid
Event severity            5. Low Priority
Entry type                301. Shutdown ASCII Message Type
SWI Minor class           9. ASCII Message
SWI Minor sub class       2. Shutdown
ASCII Message             System rebooted by root:
=====
```

Event Manager は、binlogd イベントを DECEvent または Compaq Analyze に渡して、そのイベントの変換情報を得ます。どちらのプログラムも利用できないか、変換に失敗した場合、変換情報のエリアには、失敗を示すメッセージが表示されます。

いくつかの要因により、個々のシステムで利用できる binlogd イベント変換情報のタイプが決まります。

- 旧世代の Alpha プロセッサ・プラットフォーム (初期の EV6 プラットフォームなど) では、DECEvent を利用できます。新しい EV6 プラットフォームでのイベントの変換には、Compaq Analyze を使用しなければなりません。
- 変換に DECEvent を使用する場合は、DECEvent イベント・フォーマッタ・ユーティリティ /usr/sbin/dia がローカル・システム上にインストールされていなければなりません。このユーティリティがシステムにインストールされていない場合は、「Associated Products」

CD-ROM からインストールしなければなりません。詳細については、インストールのドキュメントを参照してください。Compaq Analyze がサポートしているシステムには、DECevent をインストールする必要はありません。

- DECEvent とは異なり、Compaq Analyze はクライアント/サーバ・モデルを使用しているため、Compaq Analyze を使用するすべてのシステムにインストールする必要はありません。ライセンスされた Compaq Analyze が少数のシステム上でのみ実行されているサイトの場合でも、これらのシステムは他のシステムに対して変換サービスを提供することができます。変換にリモートの Compaq Analyze サーバを使用する必要がある場合は、ローカル・システムのチャンネル構成ファイルを、下記の説明のように編集しなければなりません。
- 新しいプロセッサでは、初期のプラットフォームで生成される形式とは異なる形式のヘッダを持つ、binlogd イベントが生成されます。これらの新しいフォーマットのイベントは、CEH (Common Event Header) イベントといいます。システムが CEH イベントを生成しない場合は、変換に Compaq Analyze を使用することはできません。また、DECevent フォーマッタ・ユーティリティ /usr/sbin/dia をインストールしなければなりません。

システムが binlogd イベントの変換に DECEvent を使用する場合は、ローカル・システム上で稼働している Compaq Analyze サーバを使用する場合には、標準の構成を変更する必要はありません。リモート・システム上で稼働している Compaq Analyze サーバを使用する場合は、/etc/evmchannel.conf ファイルを編集する必要があります。省略時のインストールでは、binlog イベント・チャンネルの fn_details 行は、次のように構成されています。

```
fn_details      "binlog_details -decevent -ca localhost"
```

この行は、DECevent が利用可能であればそれを変換に使用することを Event Manager に指示します。利用可能でなければ、Event Manager はローカル・ホスト上で稼働している Compaq Analyze サーバに接続しようとします。どちらのオプションも成功しなかった場合、Event Manager は Compaq Analyze のスタンドアロン・モードでの実行を試み、それでも失敗したら、変換は行われません。これらのオプションは、リスト内の最初の 2 項目として残しておいてください。ただし、Compaq Analyze サーバが稼働しているシステムが他にある場合は、さらに -ca 項目を追加することができます。

次の例では、Event Manager は DECevent、ローカル・システム上の Compaq Analyze、リモート・システム gandalf 上の Compaq Analyze、最後にリモート・システム tigger 上の Compaq Analyze を順に試します。

```
fn_details "binlog_details -decevent -ca localhost -ca gandalf -ca tigger"
```

構成ファイルを編集した後は、evmreload -c コマンドを実行して、ファイルがアップデートされたことを Event Manager チャンネル・マネージャに認識させなければなりません。

Event Manager は Compaq Analyze サーバを起動しないため、変換を成功させるためには、選択したシステム上で Compaq Analyze サーバがすでに稼働していなければなりません。このサーバは通常、システムの初期化時に自動的に起動されます。詳細は、Compaq Analyze のドキュメントを参照してください。

いずれかの変換ユーティリティがローカル・システム上で利用可能かどうかを調べる手順については、13.4 節を参照してください。

13.3 システム管理での Event Manager の使用

Event Manager の複数のイベント・ソースを監視し 1 つのイベント・ストリームとして結合する能力は、システムのアクティビティを監視する際に非常に役立ちます。省略時設定では、ロガーは 600 (警告) 以上の優先順位を持つイベントがポストされたときにメールを root ユーザに送ります。ただし、イベント・ビューアやコマンド行ユーティリティを用いて、すべてのイベント・ログを毎日確認する必要があります。選択した基準に従って別の手段 (ポケットベルにメッセージを送信するなど) をとるように、ロガーを構成することができます。また、端末で evmwatch コマンドを使ってイベントの発生を監視することができます。

以降の項では、イベント動作の監視や表示に使用できるコマンドについて、例を示します。Event Manager のコマンド・セットに慣れてくると、システムで何が起こっているかを簡単に追跡できるコマンド・セット、シェル・スクリプト、およびフィルタを簡単に作成できるようになります。

13.3.1 evmshow を使用した、イベントの表示

Event Manager イベントはバイナリ・データのパッケージであるため、端末上に表示するにはテキストへ変換しなければなりません。evmshow コマンドは stdin ストリームまたは指定されたファイルからバイナリの Event

Manager イベントを読み取り、同じイベントを `stdout` にテキスト形式で出力します。たとえば、Event Manager イベントが入っているファイルの内容を、次のコマンドで表示します。

```
# cat my_events | evmshow | more
```

このコマンドは、ログ・ファイルからイベントを取り出して省略時の方法で表示します。つまり、フォーマット・データ項目を各イベントから取り出し、参照している変数をその値に置き換え、展開して表示します。変数への参照は、ドル記号 (\$) で示されます。このため、`my_events` ファイルにフォーマット・データ項目 `AdvFS: AdvFS domain $domain is full` が含まれ、`root_domain` という値の `domain` という変数がイベントに含まれている場合、対応する出力行は次のとおりになります。

```
AdvFS: AdvFS domain root_domain is full
```

この情報では何が発生したかは分かりますが、いつ発生したかや、イベントの重要度は分かりません。 `-t` オプションを使用して表示テンプレートを指定することによって `evmshow` コマンドの出力を変更し、イベント内にタイムスタンプや優先順位などのデータ項目を含めることができます。表示テンプレートとは、イベントに表示するデータ項目と、それらをどのように表示するかを示すテキスト文字列です。

次の例では、表示テンプレートを使用して、タイムスタンプ、優先順位、フォーマットされたイベント・メッセージのあるイベントを表示します。表示テンプレートでは、表示される各項目の名前の前に、アットマーク (@) が付いています。アットマーク 2 個 (@@) は、イベントのフォーマット項目を展開して表示することを示します。2 番目の行は、ドメイン・フル・イベントの出力です。この出力は、イベントの優先順位が大カッコで囲まれ、メッセージ・テキストの前にスペースが 2 個あり、表示テンプレートで指定されたとおりになっています。

```
# cat my_events | evmshow -t "@timestamp [@priority] @" | more
22-Jun-2000 11:22:27 [600] AdvFS: AdvFS domain
root_domain is full
```

独自の表示テンプレートをセットアップして、任意のフォーマットで重要な項目を表示することができます。すべてのデータ項目のリストについては、`EvmEvent(5)` を参照してください。使用するスタイルが決まれば、省略時の表示テンプレートを `EVM_SHOW_TEMPLATE` 環境変数に設定することで、コマンド行での入力を少なくすることができます。次の Korn シェル (ksh) コマンドは、前の例と同等です。

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority]  @"
# cat my_events | evmshow | more
```

イベントの情報がさらに必要な場合は、evmshow コマンドに -d オプションを指定することによって、説明と内容のフル・ダンプを含む、詳細表示を要求できます。次の例は、AdvFS ドメイン・フル・イベントの詳細表示です。

```
# cat my_events | evmshow -d | more
===== EVM Log event =====
EVM event name: sys.unix.fs.advfs.fdmn.full

    This event is posted by the AdvFS filesystem to provide
    notification that the specified AdvFS domain is full.  No more
    space is available for writing.  [1]
=====

Formatted Message:
    AdvFS: AdvFS domain root_domain is full  [2]

Event Data Items:  [3]
    Event Name      : sys.unix.fs.advfs.fdmn.full
    Cluster Event   : True
    Priority         : 600
    PID             : 1177
    PPID            : 724
    Timestamp       : 22-Jun-2000 11:22:27
    Host IP address : 0.0.0.0
    Host Name       : x.x.example.com
    User Name       : root
    Format          : AdvFS: AdvFS domain $domain is full  [4]
    Reference       : cat:evmexp.cat:450

Variable Items:  [5]
    domain (STRING) = "root_domain"

=====
```

[1] そのイベントについて説明するテキスト文字列。 場合によっては、このデータ・フィールドに、問題を解決するために推奨されるアクションが示されます。

[2] 「Formatted Message」セクション。

[3] 「Event Data Items」セクション。 イベントに含まれる標準的なデータ項目がすべてリストされます。 各項目の説明については、EvmEvent(5)を参照してください。

ここに表示される項目は多くのイベントで一般的なものですが、一部の項目がない場合や、追加の項目が表示される場合があります。たとえば、大半のイベントはクラスタ内の全ノードには配布されないため、多くの場合「Cluster Event」項目は表示されません。

- ④ 「Format」データ項目。「Formatted Message」データ項目の内容とほぼ同じです。ただし、\$ 記号が前に付いた *domain* という変数への参照が含まれています。
- ⑤ 「Variable Items」セクション。変数 *domain* の値が入っています。

詳細表示のイベントを選択する方法は、13.3.12.2 項を参照してください。

`evmshow -x` コマンドを使用して、説明だけを表示することができます。また、`-x` オプションと `-t` オプションを一緒に使用すると、イベントの要約を表示し、その直後にその説明を表示することができます。例を次に示します。

```
#cat my_events | evmshow -x -t "@timestamp
[@priority]  @" | more
21-Jun-2002 11:22:27 [600] AdvFS: AdvFS domain root_domain is full
This event is posted by the AdvFS filesystem to provide
notification that the specified AdvFS domain is full.
No more space is available for writing.
```

この項の例では、1つのログ・ファイルに入っている Event Manager イベントの表示方法を示します。さまざまなシステム・ログ・ファイルに格納されているイベントを表示したり、イベントの発生状況を監視することもできます。このようにするには、`evmget` コマンドと `evmwatch` コマンドを使用します (13.3.3 項および 13.3.6 項を参照)。

多くのシステムは大量のイベントを生成しますが、ほとんどが通常動作の報告です。イベント・フィルタを使用して、表示を、関心のあるイベント・セットに限定してください。13.3.2 項では、Event Manager のフィルタ機能の概要について説明しています。

イベントの発生元に関係なく、`evmshow` コマンドを使用してイベントを表示用にフォーマットできます。表示テンプレートの詳細については、`evmshow(1)` を参照してください。

13.3.2 イベント・フィルタの概要

この項では、イベント・フィルタの概要について説明し、直前の項の `evmshow` コマンドの例に当てはめて説明します。フィルタリングは、イベントの取り出しおよび監視の技法を説明する、後半の項で広く使用されています。完全なフィルタ構文は、`EvmFilter(5)` のリファレンス・ページに定義されています。

Event Manager イベント・フィルタは、取り出すイベントを Event Manager に知らせるテキスト文字列です。たとえば、フィルタ文字列 `[priority >= 600]` は、優先順位が 600 以上のイベントを選択します。フィルタは簡単に指定できますが、フィルタ言語は強力です。多少練習すれば、監視したいイベント・セットを正確に定義するフィルタ表現を簡単に作成して格納できます。フィルタを使用するのは、一部の Event Manager コマンド行ユーティリティ、Event Manager ロガー、およびシステム・デーモンとクライアント・アプリケーションです。

`evmshow` コマンド、`evmget` コマンド、および `evmwatch` コマンドでは、`-f` オプションをサポートし、このオプションを使ってフィルタ文字列を指定します。次の例のようにして、表示するイベントを `my_events` ファイルから選択することができます。

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"  
# cat my_events | evmshow -f "[priority >= 600]" | more
```

(この例の概要は、13.3.1 項で説明されています。) この例では、`-f` オプションでフィルタを指定し、600 以上の優先順位を持つイベントを選択しています。このコマンドはすべてのイベントをファイルから読み取りますが、フィルタ文字列と一致したイベントだけを返します。

取り出すイベントの名前が分かっている場合は、次の例のように、フィルタにそれらの名前を指定することができます。

```
# cat my_events | evmshow -f "[name sys.unix.fs.advfs.fdmn.full]" | more
```

名前のコンポーネントの部分には、次のようにワイルドカード文字を使用できます。

- アスタリスク (*) 文字は 0 個以上の完全なコンポーネントと一致します。
- 疑問符 (?) は 1 個の完全なコンポーネントと一致します。

たとえば、直前のコマンド例を短くして、次の例のようにすることができます。

```
# cat my_events | evmshow -f '[name *.advfs.fdmn.full]' | more
```

このワイルドカードのアスタリスクが、コンポーネント `sys.unix.fs` と一致します。ワイルドカード文字をシェルがファイル名に展開しないようにするには、フィルタ文字列を、ダブルクォートではなく、シングル・クォートで囲みます。この注意は、シェル・コマンド内で特殊文字を使用する場合はいつでもいえることです。

名前でフィルタリングする場合、Event Manager はコマンドで指定されていない場合でも、名前文字列の末尾に ワイルドカード `.*` があるものとして扱います。このため、指定したものより名前コンポーネントの数が多いイベントを受け取ることがあります。次の 2 つのコマンドは同等ですが、1 番目のコマンドの最後のワイルドカード (`.*`) は不要です。

```
# cat my_events | evmshow -f '[name *.advfs.*]'\n# cat my_events | evmshow -f '[name *.advfs]'
```

`evmshow` コマンドの実行時に表示テンプレート内の項目の 1 つとして `@name` を指定すると、イベントの名前を表示させることができます。

フィルタの構文では、`AND` キーワード、`OR` キーワード、および `NOT` キーワードを使用して、複数の条件を 1 つのフィルタに結合することができます。また、カッコを使用して、条件をグループ化することができます。次のコマンド例は、`advfs` というコンポーネントを名前に含み、優先順位が 600 以上のすべてのイベントを選択します。

```
# cat my_events | evmshow -f '[name *.advfs] and [priority >= 600]'
```

次のコマンドは、優先順位に関係なく、名前コンポーネントが `binlog` のイベントも選択します。この例では、`priority` キーワードが `pri`、`name` が `na` に短縮されています。大半のフィルタ・キーワードは、`EvmFilter(5)` のリファレンス・ページに説明されているように短縮形で記述できます。

```
# cat my_events | evmshow -f '([na *.advfs] and [pri >= 600]) or [na *.binlog]'
```

この項の例では、最もよく使用されるフィルタ・キーワードについて説明します。`evmshow` コマンドでのフィルタの使い方と、以下の項で説明されている Event Manager コマンドに慣れたら、より高度なフィルタ機能を使用して、役に立つフィルタを作成して保存したり、最も必要なイベントを選択する能力を高めることができます。高度なフィルタリング技法については 13.3.12 項を参照し、完全な構文については `EvmFilter(5)` のリファレンス・ページを参照してください。

13.3.3 `evmget` を使用した、格納されているイベントの取り出し

システム・ログ・ファイルには、さまざまなフォーマットと詳細度レベルでイベントが格納されているため、従来のシステム・ユーティリティを使用してすべてのイベントに対する整った表示を行うことは困難です。`evmget` コマンドを使用すると、各種のログ・ファイルからイベントを取り出し、まだ Event Manager の形式でなければイベントを Event Manager イベントに変換し、Event Manager イベントの単一ストリームを返すことによって、

整った表示を行うことができます。evmshow コマンドを使用すると、Event Manager イベント・ストリームを表示形式に変換できます。

次のコマンド・パイプラインは、evmget コマンドを使用してすべてのシステム・イベントを取り出し、表示するために evmshow コマンドに渡します。

```
# evmget | evmshow -t "@timestamp" [@priority] @@ | more
```

evmget コマンドは、Event Manager デーモンへのサービス・コネクションを確立します。このコネクションは、get サーバ・プログラム /usr/sbin/evm_getsrv の新しいコピーを起動します。get サーバ・プログラムはチャンネル構成ファイルを読み取り、get 関数 (通常、チャンネル構成ファイル /etc/evmchannel.conf に構成されている各チャンネルに対するシェル・スクリプト) を実行します。この構成ファイルについては、13.2.2.2 項を参照してください。

get 関数は、次の処理を実行します。

- チャンネルのログ・ファイルを読み取ります。
- イベントを EVM フォーマットに変換します。
- stdout ストリームにイベントを書き込む evmget コマンドに、イベントを戻します。

すべてのチャンネル get 関数が実行され、すべてのイベントが返却されると、get サーバ・デーモンと evmget コマンドはどちらも終了します。

注意

イベントはテキスト行または特殊なバイナリ・フォーマットとしてログ・ファイルに格納されていることがありますが、evmget コマンドはすべてのイベントを、evmshow に渡して表示することができる、バイナリ形式の Event Manager イベントとして返します。evmget の出力を直接端末に送信すると、このコマンドはエラー・メッセージを表示します。これは、バイナリ出力は適切に表示できないので、端末の設定に影響を与えることがあるためです。出力を more コマンドや pg コマンドなどの他のコマンドにパイプすると、evmget コマンドはこのエラーを検出できないため、ランダムな文字が表示されます。

evmshow コマンドと同様に, evmget コマンドは返却するイベントを制限するためのフィルタ・オプションをサポートしています。たとえば, 次のコマンドは, 優先順位が高いイベントだけを表示します。

```
# evmget -f '[pri >= 600]' | evmshow | more
```

evmshow コマンドでフィルタを指定するより evmget コマンドでフィルタを指定する方がより効果的です。これは, evmget コマンドがフィルタ文字列をイベント・チャンネルの get 関数 (フィルタに一致するイベントだけを返す) に渡すためです。get サーバ・デーモンから evmget コマンドに返されるイベントが少なくなり, コマンドが転送し処理するイベントが少なくなるため, コマンドの動作が速くなります。

取り出されたイベントを後で分析するために保存したい場合や, 別のシステムにコピーしたい場合は, evmget コマンドの出力をファイルにリダイレクトできます。たとえば, 次のコマンドを入力します。

```
# evmget -f '[pri >= 600]' > my_events
```

evmget コマンドのバイナリ形式の出力を保存する方が, evmshow コマンドのテキスト形式の出力を保存するよりも, 柔軟性があります。後で, バイナリ・ファイルをソートしフィルタリングして evmshow コマンドに渡し, 好きなフォーマットで表示することができます。

evmget コマンドを試してみると, 複数のイベントがまとめて出力されていることが分かります。多くの場合, すべてのバイナリ・エラー・ロガー・イベントが最初に出力されています。それぞれのまとまりの中では, イベントは一般的に, 日時順に並んでいます。これは, 省略時のチャンネル構成ファイルでは binlog イベント・チャンネルが最初に指定されており, その get 関数が最初に行われるためです。各 get 関数はイベントを順に evmget コマンドに渡し, evmget コマンドはそれらのイベントを受け取った順で出力します。イベントを何らかの順序 (多くの場合, 日時順) で表示したくなるので, イベントを evmsort コマンド (13.3.4 項を参照) にパイプする必要があります。13.3.5 項では, パイプラインを使用しないで, イベントを取り出し, ソートし, 表示することができる, -A オプションを指定した evmget コマンドの使用方法について紹介します。

システムの規模やタイプ, 記録されるイベントの数によっては, イベントの取り出しに長時間かかることがあります。これは, 取り出し操作のたびに, 各チャンネルの get 関数とそのログ・ファイルを読み取り, イベントを Event Manager イベントに変換してから, イベントを evmget コマンドに渡すかど

うかを判断するためにフィルタ文字列を適用する(フィルタがある場合) 必要があるためです。ログ・ファイルが大きくなるほど、処理に時間がかかります。ログ・ファイルを注意深く管理すると、処理を速くするのに役立ちます。特定のイベント・チャンネルに属しているイベントだけを表示したい場合は、`evmget -C` コマンドを使用して指定したチャンネルだけを表示するようにすると、処理時間を短くできます。たとえば、次のコマンドを入力します。

```
# evmget -f '[pri >= 600]' -C binlog | evmshow | more
```

この例では、`get` 関数は `binlog` チャンネルについてのみ処理するため、コマンドの作業が早く完了します。フィルタ文字列が指定されており、優先順位が 600 以上のイベントが返却されます。どのようなチャンネルが構成されているかは、`evminfo -lc` コマンドを使用するか、チャンネル構成ファイルを調べると分かります。詳細については、`evminfo(1)` を参照してください。

13.3.4 `evmsort` を使用した、イベントのソート

`evmsort` コマンドは Event Manager イベントのストリームを入力とし、要求された順序でそのイベントをソートし、`stdout` ストリームにイベントを書き込みます。このコマンドは `evmget` コマンドの出力をソートするのに最も適していますが、どのソースからの Event Manager イベントのソートにも使用できます。詳細は、`evmsort(1)` を参照してください。

13.3.3 項では、`evmget` コマンドで取り出したイベントは、イベント・チャンネル構成に応じて、まとめて出力されると説明しました。`evmsort` コマンドを使用すると、`evmshow` コマンドに送って表示する前に、イベントを希望の順序にソートすることができます。次の例は、一般的なコマンド・シーケンスです。

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

特に指定しなければ、`evmsort` コマンドはイベントを日時順でソートします。このため、このコマンドは多くの場合、適切です。別の方法でイベントをソートしたい場合は、`-s` オプションを使用してソート指定を宣言できます。ソート指定とは、イベントをソートするデータ項目であるソート・キーを 1 つ以上定義するテキスト文字列です。この指定は、コロン(:) で区切られたデータ項目名のリストです。例を次に示します。

```
priority:timestamp
```

上記の指定は、優先順位の同じイベントをタイムスタンプ順にソートします。このため、返却されるイベントの最初のグループは、発生順にソートされた、優先順位が最も低いイベントです。この指定は、次のように使用します。

```
# evmget -f '[pri >= 600]' | evmsort -s "priority:timestamp" | evmshow | more
```

省略時のソート順は昇順ですが、マイナス記号 (-) を付加すると、項目指定子ごとに降順に変更できます。プラス記号 (+) を指定すると、明示的に昇順を要求できます。たとえば、次のコマンドは優先順位が最も高いイベントを最初に表示します (降順) が、各優先順位内ではイベントは古い順にソートされます (昇順)。

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+" | evmshow | more
```

表示テンプレート構文との一貫性を保つために、evmsort コマンドでは、各項目指定子の前に @ (アット) 文字を付けることができます (13.3.1 項を参照)。このようにする必要はありませんし、付けても動作には影響しません。

ソート方法が決まったら、EVM_SORT_SPEC 環境変数を設定して、省略時のソート・シーケンスを変更できます。次の Korn シェル (ksh) コマンドは、上記の例と同等です。

```
# export EVM_SORT_SPEC="priority-:timestamp+"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

-s オプションで別のソート指定を渡すと、いつでも EVM_SORT_SPEC 変数の値より優先させることができます。

13.3.5 -A オプションを使用した、コマンド文字列の簡略化

Event Manager のコマンドは、ブロック構造で設計されており、各コマンドは特定の動作を 1 つ行います。このことにより、イベント情報を操作するシェル・スクリプトの開発を、非常に柔軟に行うことができます。コマンドをコマンド行から入力する場合は、簡単な形式のコマンドを使用することもできます。

イベント取り出しの最も一般的なコマンド・シーケンスでは、evmget コマンドを evmsort コマンドにパイプした後、evmshow コマンドにパイプします。この後、テキスト出力を more コマンドにパイプし、出力を表示することができます。例を次に示します。

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+" | evmshow | more
```

evmget -A コマンド・オプションを使用すると、上記のコマンドを簡単にすることができます。このオプションは、コマンドの出力を他の Event

Manager コマンドへ自動的にパイプします。たとえば、`-A` オプションを使用すると、上記のコマンド例を次のように簡単にできます。

```
# evmget -A -f '[pri >= 600]' -s "priority-:timestamp+" | more
```

`evmget -A` コマンドを起動すると、`evmsort -A` コマンドが自動的に起動され、出力がそのコマンドにパイプされます。`evmsort` コマンドを起動すると、`-A` オプションの使用により `evmshow` コマンドが起動され、イベントがパイプで渡されて表示されます。`-s` オプションによるソート指定や、`-t` オプションによる表示テンプレートを使用することができます。これらのオプションは、それぞれ `evmsort` コマンドと `evmget` コマンドに渡されます。

`evmwatch` コマンドは、`-A` オプションをサポートしています (13.3.6 項を参照)。

13.3.6 `evmwatch` を使用した、イベントの監視

`evmwatch` コマンドを使用すると、ターミナル・ウィンドウを通してイベントをアクティブに監視できます。このコマンドは、サブスクライブを行う Event Manager のクライアントです。このコマンドは Event Manager デモンへのコネクションを確立し、サブスクライブ要求を送信し、イベントの受信を待機します。イベントが到着すると、`evmwatch` コマンドはイベントを自分の標準出力ストリーム (`stdout`) に、バイナリ Event Manager イベントとして書き込みます。

`evmwatch` コマンドの出力はバイナリ・イベントのストリームであるため、これを表示することはできません。`evmshow` コマンドを使用して、イベントをフォーマットしなければなりません。次の例はすべてのイベントを監視し、発生するたびにターミナルに表示します。

```
evmwatch | evmshow -t "@timestamp [@priority] @@"
```

システムのタイプとイベント動作のレベルによっては、イベントを表示するまでに、このコマンドがしばらく動作することがあります。このコマンドは、ユーザが (普通は `Ctrl/c` を押して) コマンドを終了させてターミナルの制御を取り戻すまで、動作し続けます。

システムの正常動作時にポストされるイベントの多くは、優先順位の低い情報イベントです。システムでのイベント動作が高レベルの場合は特に、これらのイベントをフィルタリングして外したいこともあります。`evmwatch` コマンドにフィルタを指定すると、情報メッセージをフィルタリングできます。

```
# evmwatch -f "[priority >= 400]" | evmshow -t "@timestamp [@priority] @@"
```

この例では、優先順位がエラー以上のイベントを監視します。このフィルタ文字列を変更すると、常時発生する重要でないイベントのセットを除外することができます。代わりに、特定のイベントのセットを監視しなければならないこともあります。

上記の例では、`evmshow` の出力が、表示用の `more` にパイプされていません。これは、`evmwatch` コマンドがリアルタイムのモニタだからです。`evmwatch` コマンドは、ファイルからイベントを表示するのではなく、発生するたびにイベントを出力します。`pg` や `more` のようなコマンドでは、1 画面目のデータを表示した後、入力パイプからさらにデータを読み込む前にオペレータの入力を待機するため、長時間ほっておくとパイプラインが詰まってしまうことがあります。Event Manager デーモンでは、クライアント (`evmwatch` コマンド) がバックログをクリアするのを待つことができないため、これにより `evmwatch` コマンドがイベントを紛失してしまいます。イベントを `pg` や `more` にパイプする代わりに、`evmwatch` コマンドの出力を直接ターミナル・ウィンドウ上に表示し、スクロールバーを使用してイベント・リストを参照するようにします。

`evmwatch` コマンドの出力は、`evmsort` にパイプしないでください。`evmsort` コマンドは入力の最後まで読み取らないと、イベントをソートできないためです。通常 `evmwatch` コマンドは、明示的に終了させるまで、監視プログラムとして入力を待ちます。このため、`evmwatch` コマンドの出力を直接 `evmsort` コマンドにパイプすると、`evmsort` コマンドからは何も出力されません。

`-A` オプションを使用すると、`evmsort` コマンドと `evmshow` コマンドが自動的に実行されるため、コマンド文字列が簡単になります。`evmwatch` コマンドも `-A` オプションをサポートしており、このオプションを指定すると `evmshow` コマンドが自動的に実行されます。`evmwatch` コマンドのオプションとして、表示テンプレートを次のように指定できます。

```
# evmwatch -A -f "[priority >= 400]" -t "@timestamp \
[@priority] @@"
```

`evmget` コマンドを使用すると、関心のあるイベントのセットをファイルに捕捉し、後で参照することができます。バイナリ形式でイベントを格納する方が、テキスト形式で格納するよりも便利です。このため、`evmwatch` コマンドの出力を `evmshow` コマンドにパイプするのではなく、次のように直接ファイルにリダイレクトしてください。

```
# evmwatch -f "[priority >= 400]" > my_events
```

evmwatch コマンドは、シェル・スクリプト内でイベントを監視するのに役立つ、その他のオプションもサポートしています。詳細については、evmwatch(1) を参照してください。

13.3.7 evmpost を使用した、クイック・メッセージ・イベントのポスト

大半のイベントはシステムおよびアプリケーション・ソフトウェアからポストされますが、コマンド行やシェル・スクリプトからイベントをポストしたい場合もあります。たとえば、タスクが完了したことや、重要な事項を示すメッセージ・イベントをシステム・ログにポストすることができます。システム・ログ内にエントリを作成すると、作成したエントリとの位置関係で、他のイベントがいつ発生したかが分かりやすくなります。

evmpost コマンドを使用すると、イベントをポストできます。このコマンドの最も簡単な形式は、`-a` (管理者) オプションまたは `-u` (ユーザ) オプションを使用して指定できる、クイック・メッセージ形式です。メッセージをポストするには、クオートされた文字列として、メッセージをコマンド行に指定します。

```
# evmpost -a "Fire drill started - evacuating computer room"
```

管理用クイック・メッセージは `sys.unix.evm.msg.admin` という名前でポストされるため、名前フィルタを使用して検索できます。

```
# evmget -f '[name *.msg.admin]' |
evmshow -t 'timestamp [@priority] @@'
27-Jun-2000 15:40:49 [200] EVM admin msg: Fire drill
started - evacuating computer room
```

特に指定しなければ、このメッセージは優先順位 200 の通知イベントとしてポストされます。優先順位は、`-p` オプションで変更できます。たとえば、優先順位として 400 を指定すると、メッセージがエラー・イベントとして分類されます。

```
# evmpost -p 400 -a \
"Users reporting possible network problems"
```

省略時の設定では、root ユーザか adm グループのメンバだけが、`-a` オプションでイベントをポストできます。ただし、認証ファイル `/etc/evm.auth` を 13.2.3.2 項の説明のように編集すると、他の特権ユーザも利用できるようになります。どのユーザも、`-u` オプションを使用して同じ方法でメッセージをポストできます。必要であれば、認証ファイルを編集して、信頼できるユーザだけにこの特権を制限することができます。

13.3.8 登録済みのイベントのリスト表示

テンプレート・ファイル・エントリを追加し (13.2.5 項を参照), `-d` オプションを指定して `evmreload` コマンドを実行し Event Manager デーモンにエントリを認識させるか, システムを再起動して, イベントを登録します。

登録したイベントのリストを取り出すには, `evmwatch -i` コマンドを使用できます。 `evmwatch -i` コマンドの出力を `evmshow` コマンドにパイプし, イベント・テンプレートを好みのフォーマットで表示します。たとえば, 次のコマンドを入力します。

```
# evmwatch -i | evmshow -t "@name [@priority] @format" -x
```

テンプレートは, ファイルにリダイレクトすることも, `evmshow` コマンドにパイプして表示することもできる, バイナリ形式の Event Manager イベントとして返されます。上記の例では, 表示テンプレート (`-t` オプション) の内容に従って, イベントの名前, 優先順位, およびメッセージ・フォーマットが表示されます。 `-x` オプションにより, 各要約行の後にイベントの説明が表示されます。

表示しているのはテンプレートであり実際のシステム・イベントではないため, 展開されたメッセージではなく, イベントのメッセージ・フォーマットを要求するコマンド・シーケンスを指定します。この出力では, 要約行には変数の値ではなく変数の名前で, メッセージが表示されます。たとえば, 次のような要約行と説明テキストが表示されます。

```
sys.unix.fs.advfs.fdmn.bal.error [400] AdvFS: Balance error on AdvFS domain $domain
  This event is posted by the balance(8) command to indicate that an
  error has occurred while balancing the domain.

  Action: Please see balance(8) for further information.
```

この例では, `$domain` 変数は, ポストされたイベントのインスタンスを `evmget` コマンドを使用して取り出す際にドメイン名に置き換えられています。

登録済みイベントを全部は表示したくない場合, フィルタを使用して `evmwatch` コマンドの出力を, 表示したいイベントだけに限定することができます。

```
# evmwatch -i -f '[name *.evm]' | evmshow -t "@name \
[@priority] @format" -x
```


13.3.9 シェル・スクリプトからのイベントのポスト

新しく登録したイベントを、イベント情報をソース(テキスト)形式でコマンドに渡してポストするには、`evmpost` コマンドを使用します。イベントの構文についての詳細は、`evmpost(1)` のリファレンス・ページを参照してください。ソース・レベルでのポスト機能は、イベントで操作の成功または失敗を示すことがあるルーチン操作を実行するシェル・スクリプトで、最も役に立ちます。この項では、バックアップの終了を通知する新しいイベントを作成し、ポストする手順について説明します。基本的なステップは、次のとおりです。

1. テンプレート・ファイルを作成し、その構文を確認します。
2. テンプレート・ファイルをインストールし、Event Manager デーモンに知らせます。
3. 認証ファイルをアップデートし、イベントをポストできるようにします。
4. イベントをポストするシェル・スクリプト・コマンドを作成します。

イベントの設計ガイドラインについては、『プログラミング・ガイド』を参照してください。この本に説明されている概念を理解してから、新しいイベントの設計を開始してください。この例では、バックアップ・スクリプトは2つのイベント(優先順位 200 (通知) の `local.admin.backup.ok` と、優先順位 400 (エラー) の `local.admin.backup.failed`) のうちの1つをポストします。失敗したイベントには、バックアップ・プログラムから返された終了コードを保持する `result_code` という変数が入っています。この変数は8ビットの符号なし整数で、テンプレート内ではダミー値 0 が入っています。このダミー値は、イベントのポスト時に実際の値に置き換えられます。テンプレート・ファイルの構文については、`evmtemplate(4)` のリファレンス・ページを参照してください。

次の手順は、新しいイベントを作成してポストする方法を示します。

1. `/var/evm/adm/templates/local` ディレクトリが存在しない場合は、作成します。
2. `vi` などのテキスト・エディタを使用して、次のテキスト・ファイルを作成します。

```
# This file contains EVM event templates for local
# backup notification events.
event {
```

```

    name local.admin.backup.ok
    format "BACKUP: Backup completed OK"
    priority 200
}

event {
    name local.admin.backup.failed
    format "BACKUP: Backup failed - code $result_code"
    var {name result_code type UINT8 value 0}
    priority 400
}

```

3. このファイルを backup.evt という名前で、/var/evm/adm/templates/local ディレクトリに保存します。

/var/evm/adm/templates ディレクトリ下のディレクトリであれば、どこにでも新しいテンプレート・ファイルをインストールできます。ただし、サブディレクトリおよびテンプレート・ファイルの名前は識別しやすいように、イベントの名前に由来する名前にしてください。密接に関連する少数のイベント・テンプレートを1つのテンプレート・ファイルに保持すると、管理が簡単になります。

4. テンプレートの構文を確認します。テンプレート・ファイルの構文はイベントをポストするときに使用する構文と同じなので、evmpost -r コマンドを使用して構文を確認できます。-r オプションは、イベントのポストではなく、構文の確認であることを evmpost コマンドに指示し、入力をバイナリの Event Manager イベントに変換してから、Event Manager イベントを自分の標準出力 (stdout) ストリームに書き込みます。テンプレート項目のイベントへのマージを回避するか、タイムスタンプやホスト名などの環境項目を追加するには、evmpost -M コマンド・オプションを使用します。

どのバイナリ Event Manager イベントのストリームの場合も、evmshow コマンドを使用して、evmpost コマンドの出力を確認できます。この処理を行うには、次のコマンドを入力します。

```
# cat /var/evm/adm/templates/local/backup.evt |
  evmpost -r -M | evmshow -t "@priority @"
```

ファイルが正しく作成された場合、次の出力が表示されます。

```
200 BACKUP: Backup completed OK
400 BACKUP: Backup failed - code 0
```

5. ファイルの所有者が root または bin で、パーミッションが 0400、0600、0440、または 0640 であることを確認します。必要であれば、

chown コマンドおよび chmod コマンドを使用して、パーミッションを修正します。

6. 次のコマンドを実行して、Event Manager デーモンに構成を再ロードさせます。

```
# evmreload -d
```

コマンドがエラー・メッセージを表示した場合は、問題を解決して、コマンドを再入力します。最も考えられる原因は、ファイルの所有者やパーミッションの誤りです。

7. Event Manager デーモンのデータベースからテンプレートを取り出す `evmwatch -i` コマンド・オプションを使用して、テンプレートの登録を確認します。 `evmwatch` コマンドはテンプレートをバイナリ Event Manager イベントの形式で出力します。テンプレートを表示するには `evmshow` コマンドを使用します。正しく登録されていることを確認するのに必要なのは、次の例のように、イベントの名前の表示だけです。

```
# evmwatch -i -f "[name local.admin.backup]" |  
  evmshow -t "@name"  
local.admin.backup.ok  
local.admin.backup.failed
```

8. 認証ファイル `/etc/evm.auth` をアップデートし、イベントがポストされるようにします。次の行を追加して、root ユーザだけがこのイベントをポストし、すべてのユーザがこのイベントを参照できるようにします。

```
# Local backup events:  
event_rights {  
    class      local.admin.backup  
    post       root  
    access     +  
}
```

名前の最初の3コンポーネントだけが指定されています。これらのコンポーネントは新しい2つのイベントに共通なので、どちらのイベントがポストされても、その名前とこのエントリが一致します。

9. `evmreload -d` コマンド・オプションを実行して、デーモンがこの新しい認証エントリを認識するようにします。

10. 次のコマンドを使用して、イベントのログが正しく取得されていることを確認します。

```
# echo 'event {name local.admin.backup.ok}' | evmpost  
# echo 'event {name local.admin.backup.failed}' | evmpost  
# evmget -f '[name local.admin.backup]' |  
  evmshow -t '@timestamp [@priority] @@'
```

```
28-Jun-2002 15:21:39 [200] BACKUP: Backup completed OK
28-Jun-2002 15:21:40 [400] BACKUP: Backup failed - code 0
```

上記の例では、`evmpost` コマンドが標準入力 (`stdin`) ストリームからソース入力を読み取り、入力を Event Manager イベントに変換して、そのイベントをポストしています。最後のコマンドの出力は、ポストされたイベントを表示します。この出力には、テンプレート・ファイルに指定された優先順位が含まれています。これは、各イベントがポストされる際に、Event Manager デーモンがテンプレート情報を各イベントとマージするためです。2 番目のイベントのコードの値はゼロです。これはテンプレート・ファイルに設定されたダミー値であり、ポストされたイベント内でその値が置き換えられなかったためです。バックアップ・スクリプトでは、この値にゼロ以外の値を設定します。

11. 次の例のように、ポスト・コマンドをバックアップ・スクリプトに追加します。

```
#!/bin/sh
# This shell script runs the backup operation
# and posts an event to indicate success
#or failure.

do_backups # Performs the backup operation
if [ $? -eq 0 ]
then
# success
echo 'event {name local.admin.backup.ok}' | evmpost
else
# failure
RES=$?
evmpost << END
event {
name local.admin.backup.failed
var { name result_code type UINT8 value $RES }
}
END
fi
```

上記の例で、成功したイベントの `evmpost` コマンドへの入力は簡単なので、`echo` コマンドを使用して同じ行で指定しています。失敗したイベントの場合、`result_code` 変数の値も渡さなければなりません。この値を渡すには、シェルの `<<` 構文を使用して、より構造化された複数行フォームの入力を使用します。どちらの入力フォームも、標準入力 (`stdin`) ストリームを通して、`evmget` コマンドにソース・コード入力を渡します。

コマンド行またはシェル・スクリプトからのイベントのポストについての詳細は、`evmpost(1)` を参照してください。

13.3.10 Event Manager のマーク・イベントについて

イベント動作の表示や監視を行うと、次のイベントが 15 分ごとに発生していることが分かります。

```
26-Jun-2000 08:57:45 [200] EVM: Mark event
```

evmlog イベント・チャンネルでは、このイベントをポストして、定期的にイベント動作が発生するようにします。システムに問題が発生し、最後に動作していたのがいつかを調べる必要がある場合、次のコマンドでシステム・ログ内のマーク・イベントを探すことで確認できます。

```
# evmget -f "[name *.evm.mark]" | evmshow -t "@timestamp @last_timestamp @@"
26-Jun-2000 00:57:35 26-Jun-2000 04:42:40 [16 times] EVM: Mark event
26-Jun-2000 04:57:41 - EVM: Mark event
26-Jun-2000 05:12:41 - EVM: Mark event
26-Jun-2000 05:27:41 - EVM: Mark event
26-Jun-2000 05:42:41 26-Jun-2000 09:12:45 [15 times] EVM: Mark event
```

省略時のログ構成ファイルを使用している場合は通常、個別のマーク・イベントが 3 個と、その後に [n times] (n は数で、最大 16) が前に付いたイベント 1 個が表示されます。これは、最大 4 時間の間の複数のイベントを結合して無駄なスペースを最小限にする、ログの抑制機能の結果です。通常のタイムスタンプ値は、結合されたイベントのうち最初のイベントの発生を示します。last_timestamp データ項目は、最後のイベントの発生時刻を示します。この例では、09:12:45 にポストされた最後のマーク・イベントを表示する表示テンプレート内に、last_timestamp データ項目があります。このマーク・イベントは、この時刻にシステムが動作していたことを示しています。

マーク・イベントのポストを無効にするには、チャンネル構成ファイルを編集して、次のいずれかの変更を行います。

- evmlog チャンネルの fn_monitor エントリをコメント・アウトして、完全に無効にする。
- チャンネルの mon_period 値を変更して、イベントがポストされる頻度を変更する。

チャンネル構成ファイルについての詳細は、13.2.2.2 項と、evmchannel(4) を参照してください。イベントの抑制についての詳細は、13.2.2.3 項と、evmlogger.conf(4) を参照してください。

13.3.11 SysMan イベント・ビューアを使用した，イベントの表示

SysMan のグラフィカル・イベント・ビューアには，システム・イベント・ログへの簡単で便利なインタフェースが用意されています。イベント・ビューアは SysMan システム管理群の不可欠な部分であり，X ウィンドウ・ディスプレイや，PC アプリケーションなどの文字セル端末，Web ブラウザなど，さまざまなグラフィカル環境で使用できます。ビューアを SysMan Station から起動することもできます。SysMan の使用については，第 1 章を参照してください。

イベント・ビューアをコマンド行から起動するには，`sysman` コマンドを入力してから，[モニタリング/チューニング] メニュー項目をオープンします。[イベントの参照] オプションを選択して，イベント・ビューアを起動します。CDE からイベント・ビューアを直接起動するには，CDE のフロント・パネルのツール・ドロワをオープンし，「システム管理」，「日常管理」，「イベント・ビューア」の順で選択します。

イベント・ビューアを初めて実行すると，イベントがフィルタリングされて優先順位が高いイベントだけが表示されるという警告メッセージが表示されることがあります。システムが正常に動作している場合，イベント要約ウィンドウには何もイベントが表示されないのが普通です。表示するイベントを選択するには，ウィンドウの下部にある [フィルタ...] を選択し，「フィルタ」ウィンドウ内のフィルタ基準を変更します。格納されているすべてのイベントを表示する場合は，ウィンドウの左側のチェックボックスがすべてチェックされていないことを確認してから，[了解] を選択します。システムのイベント動作が高レベルの場合，「プライオリティ」ボックスをチェックして優先順位の範囲を調整すると，表示されるイベントの数を減らし，イベントの表示にかかる時間を短くすることができます。この範囲に 400 ~ 700 を設定すると，優先順位が `error` 以上のすべてのイベントが表示されます。範囲の下限に 300 を設定すると，警告イベントも表示されます。

「フィルタ」ウィンドウの左側のボタンをクリックすると，表示フィルタの基準を追加することができます。変更を行うたびに，[適用] を選択してイベント・リストに変更を適用するか，[了解] を選択して変更を適用してからメインのビューア・ウィンドウに戻らなければなりません。

「フィルタ」ダイアログ・ウィンドウでは，イベント・フィルタ文字列を入力することなく，直感的で便利な方法でフィルタを作成できます。フィルタ文字列の構文に習熟していて，フィルタの高度な機能を利用したい場合は，

メインのイベント・ウィンドウの下部にある [追加オプション...] を選択してアクセスする、「アドバンスド・フィルタ」ダイアログ・ボックスでフィルタ文字列を入力することができます。フィルタ文字列を保存して、後で再使用することもできます。フィルタ構文についての詳細は、EvmFilter(5) のリファレンス・ページを参照してください。

ビューアの最も重要な機能の 1 つは、任意のイベントの詳細表示を簡単に行えることです。要約ウィンドウ内にあるイベントを選択し、[詳細...] を選択するだけで、説明テキスト、DECEvent や Compaq Analyze からの変換 (binlog イベントの場合) など、すべての情報が表示されます。メイン・ウィンドウに戻らなくても、「イベントの詳細」ウィンドウでイベント・リストをブラウズできます。

[カスタマイズ...] および [追加オプション...] を選択すると、イベントのソースなどの、ビューアの表示を変更できます。イベントが表示される順序を変更するには、[ソート...] を選択します。ビューアとその機能についての詳細を表示するには、任意のウィンドウで [ヘルプ...] を選択します。

注意

イベント・ビューアは、イベントの発生をリアル・タイムに監視するわけではありません。アップデートされたイベント・リストを表示するには、メイン・ウィンドウの [Refresh] を選択します。

これらのアプリケーションの使用方法的詳細は、sysman(8) および evmviewer(8) を参照してください。ビューアのオプションの使用方法的については、イベント・ビューアのオンライン・ヘルプを参照してください。

13.3.12 高度な選択とフィルタリング技法

以降の項では、追加のフィルタリング技法をいくつか示しています。この技法を使ってイベント選択方法を改善すると、必要なイベントだけを受け取ることができます。

- ポスト時刻でイベントをフィルタリングする方法 (13.3.12.1 項)
- event-id 識別子を使用してイベントをフィルタリングする方法 (13.3.12.2 項)

- 予約済みのコンポーネント名を使ってイベントをフィルタリングする方法 (13.3.12.3 項)
- フィルタ・ファイルの使用方法 (13.3.12.4 項)

13.3.12.1 時刻によるフィルタリング

`timestamp` キーワード, `before` キーワード, `since` キーワード, および `age` キーワードを使用すると, イベントがポストされた時刻でイベントをフィルタリングすることができます。これらのキーワードのうち, `age` キーワードが一番簡単に使用でき, 毎日の操作で一番便利です。

`timestamp` キーワードを使用する場合は, 次の形式で, 時刻の範囲を定義する文字列を指定しなければなりません。

`year:month-of-year:day-of-month:day-of-week:hours:minutes:seconds`

任意のコンポーネントに対し, アスタリスク (*) をワイルドカード文字として使用できます。このため, 2002 年 7 月 6 日に発生したイベントを選択するには, 次のコマンドを使用します。

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# evmget -A -f '[timestamp 2002:7:6:*:*:*:*)' | more
```

最後の 4 つのコンポーネント内のアスタリスク (*) は, 時刻に関係なく, この日に発生したすべてのイベントが必要であることを示します。また, 次のコマンドのように, 任意の位置に 1 つ以上の範囲を指定することもできます。

```
# evmget -A -f '[timestamp 2002:*:*:1-3,5:*:*:*)' | more
```

4 番目のコンポーネントは, 曜日を指定します。ポスト時刻の範囲が 1 ~ 3 または 5 のイベントを検索すると, 2002 年の月曜日, 火曜日, 水曜日, または金曜日にポストされたすべてのイベントが取り出されます。

`before` キーワードと `since` キーワードでは同様の指定子文字列を使用しますが, ワイルドカード文字は使用できず, 曜日を示すコンポーネントはありません。たとえば, 次のコマンドは 2002 年 7 月 6 日の午後 3 時以降にポストされたイベントを見つけます。

```
# evmget -A -f '[since 2002:7:6:15:0:0]' | more
```

`age` キーワードを使用すると, タイムスタンプに従い, 便利で直感的な方法でイベントを選択できます。システム管理者としては, システムの問題を示す最近のイベントが最も重要です。このようなイベントを見つけるには, イベント・フィルタの `priority` キーワードと `age` キーワードを組み合わせます。たとえば, 次のコマンド・シーケンスは, 優先順位がエラー

(400) 以上で、昨日か今日発生した (イベントの経過時間が 2 日未満) すべてのイベントを表示します。

```
# evmget -A -f '[pri >= 400] and [age < 2d]' | more
```

上記の例では、2d で経過時間 (age) が 2 日未満のイベントを指定しています。age は、秒 (s)、分 (m)、時 (h)、日 (d)、または週 (w) で指定できます。各指定子がどのように使用されてイベントの経過時間 (age) が算出されるかについては、EvmFilter(5) を参照してください。

より複雑なフィルタを使用すると、より限定された期間に発生したイベントを返すことができます。次の例は、3 日以上、6 日未満前に発生したエラー・イベントを見つけます。

```
# evmget -A -f '[pri >= 400] and ([age < 6d] and [age > 3d])' | more
```

タイムスタンプによるイベントの選択と、全フィルタ構文については、EvmFilter(5) を参照してください。

13.3.12.2 詳細表示のイベントを選択する event-id の使用

evmshow -d コマンド・オプションを使用してイベントを表示すると、出力が大量になることがあります。このような場合は、表示されるイベントの数を制限できます。Event Manager を使ってポストされるイベントには、event-id という一連の識別子が含まれています。event-id を使い、詳細を表示する特定のイベントまたはイベントの範囲を選択することができます。

event-id は、デーモンが再起動されるたびにカウンターにゼロが設定されるため、特定のイベントのセット内で一意であるとは限りません。イベントを確実に一意にするには、次の例に示すようにイベントの選択時にタイムスタンプも使用する必要があります。

```
# evmget -A -f '[age < 1d]' -t "@timestamp @event_id @" | more
```

```
15-Apr-1999 14:19:06 0 EVM daemon: Configuration completed
15-Apr-1999 14:19:06 1 EVM daemon: Initialization completed
15-Apr-1999 14:19:06 2 EVM logger: Logger started
15-Apr-1999 14:19:06 3 EVM: Mark event - initial
15-Apr-1999 14:19:06 5 EVM logger: Started eventlog /var/evm/evmlog/evmlog.19990415
```

1

2

.
.
.

- ❶ この `age` フィルタ・キーワードは、今日発生したイベントすべてを選択します。今日発生したかどうかは、データの 1 番目の欄にあるタイムスタンプで示されています。
- ❷ 表示テンプレート内の `@event_id` 指定子は、取り出された各イベントの `event-id` を表示するように、`evmshow` コマンドに指示します。この `id` は、データの 2 番目の欄に示されています。

`event-ids` が表示されると、必要なイベントを選択することができます。たとえば次のコマンドを使い、開始マークのイベントの詳細を表示します。前述の例の出力では、3 という `event-id` があります。

```
# evmget -f '[age < 1d] and [event_id = 3]' | evmshow -d | more
```

次の例で示すように、さらに複雑なフィルタを使ってイベントの範囲を選択することができます。

```
# evmget -f '[age < 1d] and [event_id >= 1] and [event_id <= 3]' |  
  evmshow -d | more
```

正しいイベントのセットを選択するために、時間範囲を注意深く選択する必要があります。システムをリブートしたばかりの場合、最近 2 時間以内に発生したイベントだけを選択するには、`[age < 2h]` を指定します。

詳細表示するイベントを選択する一番簡単な方法は、13.3.11 項で説明しているイベント・ビューアを使うことです。

13.3.12.3 予約済みのコンポーネント名の検索

一部のイベント名は、名前拡張子として予約済みのコンポーネント名を含みます。このようなコンポーネントは下線文字 (`_`) で始まり、通常、イベントがポストされる原因となった項目を識別するコンポーネントが続きます。たとえば、多くのハードウェア関連のイベントの名前には、`_hwid` というコンポーネントが含まれ、その後にはその項目の数値ハードウェア識別子が続きます。予約済みのコンポーネント名は、イベント名に対する拡張として自動的に追加されます。この名前が追加されると、その後ろには変数が追加されます。これは各予約済みのコンポーネント名ごとに行われます。たとえば、`@SYS_VP@.temperature_high` という名前のイベントで変数 `_degrees` が 212 の場合、`@SYS_VP@.temperature_high._degrees.212` という名前のイベントとして出力されます。

次のコマンドを使用して、このようなイベントをすべて検索することができます。

```
# evmget -A -f '[name *._hwid]' | more
```

特定のデバイスのハードウェア識別子が分かっている場合、次のようなコマンドを使用して、検索の範囲をそのデバイスに関連するイベントに絞ることができます。

```
# evmget -A -f '[name *._hwid.4]' | more
```

13.3.12.4 フィルタ・ファイルの使用

役に立つフィルタをファイルに保存して、Event Manager の間接フィルタ機能を使用して呼び出すことができます。フィルタ・ファイルは、名前に `.evf` というサフィックスが付いていて、任意の数の名前付きフィルタを含めることができます。たとえば、次のフィルタ・ファイル・エントリは、SCSI デバイスを参照するすべての binlog イベントを選択します。

```
filter {
    name "scsi"
    value "[name @SYS_VP@.binlog.hw.scsi]"
    title "Binlog SCSI events"
}
```

この例の `@SYS_VP@` は、フィルタの使用時に `sys.unix` に置き換えられる、標準の Event Manager マクロです。

間接フィルタを使用するには、次の例のように、アットマーク (@) の後に、フィルタが格納されたファイルの名前をフィルタ文字列の代わりに指定します。

```
# evmget -A -f @binlog
```

このようなコマンドでフィルタ・ファイル名を指定する場合は、`.evf` サフィックスを省略できます。

上記の例では、ファイル内の最初のフィルタが使用されますが、次のように名前を指定して別のフィルタを選択することもできます。

```
# evmget -A -f @binlog:scsi
```

1 つのファイルに必要な数だけフィルタを入れることも、各フィルタを別々のファイルに入れることもできます。上記の例は、Event Manager に含まれている binlog フィルタを指定しています。他のフィルタは、`/usr/share/evm/filters` ディレクトリ内にあります。これらのファイルを例として使用し、独自のフィルタ・ライブラリを作成してください。

evmshow -F コマンド・オプションを使用すると、格納されているフィルタの内容を簡単に参照できます。-F オプションを使用すると、evmshow コマンドはフィルタ文字列を表示し、イベントを読み込まずに終了します。次の例では、evmshow コマンドはbinlog.evf ファイルに格納されているscsi という名前のフィルタの内容を表示します。

```
# evmshow -f @binlog:scsi -F
( [name sys.unix.binlog.hw.scsi] )
```

フィルタ・ファイルの構文の完全な情報と、ファイルを置く位置については、evmfilterfile(4) を参照してください。

注意

/usr/share/evm/filters ディレクトリ内のフィルタ・ファイルは編集しないでください。変更を行っても、将来のアップデート・インストール時に、警告なしに上書きされてしまいます。

13.3.13 イベントのロギングと転送

イベントへの応答は、サイト固有の要件や環境によって決まる任意のアクションです。この応答は、アラームを起動したり責任者を呼び出すものから、ログ・エントリを作成するものや、通常のアクティビティで予想できる現象を無視するものまであります。

あるタスクのイベント出力を次のプロセスの起動のトリガとして使用して、一連の依存関係のあるタスクを実行するようなイベント処理シーケンスを構成することもできます。Event Manager では、ロギング機能を用いて、応答アクティビティとインタフェースをとることができます。利用可能なオプションには、イベントの保存とイベント転送があります。

Event Manager ロガー evmlogger は、Event Manager デーモンによって自動的に起動され、次のような役割をします。

- システム・コンソールや他のデバイスに、選択したイベントを表示する。
構成ファイル内で端末デバイスが logfile として指定されている場合は、eventlog 文のフィルタ指定に適合するイベントすべてが端末表示用にフォーマットされます (構成ファイルの説明は、13.2.2.3 項を参照してください)。

- 選択したイベントを、1 つ以上のログ・ファイルに格納する。
- 選択したイベントを、興味を示したグループに他の形式で転送する。

ロガーは、省略時は、ローカル・デーモンからポストされたイベントを処理しますが、リモート・システムからポストされたイベントを処理するように構成することも可能です。詳細は、13.2.3.3 項を参照してください。

ロガーは、構成ファイルによって制御される、通常の Event Manager クライアントです。省略時の構成ファイルは `/etc/evmlogger.conf` ファイルであり、13.2.2.3 項で説明しています。このファイルの詳細は `evmlogger.conf(4)` を、このコマンドについての詳細は `evmlogger(8)` を参照してください。

13.3.13.1 イベントのロギング

構成ファイル内の `eventlog` グループの仕様に一致するすべてのイベントは、イベント・ログに記録されます。このファイルの省略時の位置と命名規則については、13.1.3.3 項を参照してください。

例 13-3 で示したように、構成ファイルの `eventlog` 文に `suppress` グループの仕様を含めることができます。この文があると、抑制の基準に一致するイベントはログに記録されなくなります。イベントの最初のインスタンスだけが記録され、後はそのイベントの数と最初と最後の出現時刻のデータだけが記録されます。この基準についての説明は、`evmlogger.conf(4)` を参照してください。

13.3.13.2 転送を使用したイベントの自動処理

選択したイベントを自動的に処理したい場合、Event Manager ロガーを構成して、コマンドの実行によってイベントを転送することができます。たとえば、イベント情報をポケットベル・サービスにメールしたり、またはイベント処理アプリケーション・プログラムを呼び出すことができます。

省略時の設定では、ロガーは高優先順位のイベントを `root` ユーザにメールで送るように構成されています。この省略時の転送コマンドを例として使って、独自のアクションを記述することができます。詳細は、13.2.2.3 項と `evmlogger.conf(4)` を参照してください。

構成ファイル内の `forward` 文の指定に一致するイベントはすべて、この文で指定されたコマンドの標準入力 (`stdin`) に書き込まれます。コマンドは、

シェル・スクリプトの名前，1つのUNIXコマンド，一連のUNIXコマンド(パイプライン)，またはその他の実行可能文です。一般的な転送アクションとしては，次の操作があります。

- `mail` コマンド，`mailx` コマンド，または別のコマンド行メール処理を指定し，担当者またはページング・サービスにメール・メッセージを送ります。
- 緊急シャットダウン手続きを開始する追加ソフトウェアを起動します。
- イベントの発生を待機している依存プロセスを起動します。

イベントを転送するようにロガーを構成する場合，次の点に注意してください。

- 転送するように選択したイベントは，構成済の転送コマンドにパイプされます。コマンドでテキスト情報を扱う必要がある場合，イベントをテキスト形式に変換するために `evmshow` コマンドをパイプラインの最初のコマンドにする必要があります。
- ロガーは非同期に転送コマンドを実行します。つまり，コマンドを開始してから，コマンドの終了を待たずに通常操作を続けます。次の動作は正常です。
 - ロガーの構成ファイル内に複数の転送プログラムが指定され，同じイベントが2つ以上の転送プログラムで処理される場合は，ロガーで各転送コマンドが開始され，他方のプログラムの終了を待つことはありません。そのため，コマンドが同時に実行される可能性があります。
 - 転送コマンドに処理させる新たなイベントをロガーが受け取り，転送コマンドが前のイベントをまだ処理している場合，ロガーは新しいイベントをキューに入れます。コマンドが終了すると，ロガーはコマンドを再起動して新しいイベントを渡します。省略時の設定では，各転送コマンドに対してロガーは100個までのイベントをキューに入れます。この限界値を大きくするには，転送プログラムの構成で `MAXQUEUE` キーワードを指定します。

詳細は，`evmlogger.conf(4)` を参照してください。

- イベントのテキストには，引用符などの文字が入っている可能性があります，これは，シェルにとって特別の意味があります。イベントのテ

キスト・バージョンをポストする場合は、コマンドが実際のさまざまな条件下で正しく動作することを検証する必要があります。

- 転送コマンド自身でイベントがポストされないように注意する必要があります。これはイベント・ループの原因となります。たとえば、メールを使ってイベントを転送する場合、転送プログラムのフィルタでメール・イベントを除外する必要があります。

転送プログラムやその他の構成項目を追加するには、13.2.2.4 項で示すように、ロガーの 2 次構成ファイル機能を使用します。

13.3.13.3 リモート・システムからのイベントのロギング

省略時の設定では、ロガーはローカル・デーモンからポストされるイベントだけをサブスクライブします。構成ファイルに 1 つ以上の `remote_hosts` セクションを追加することで、他のシステム上でポストされたイベントをロガーがログをとるか転送するように構成できます。最善の方法は、2 次構成ファイルにリモート接続を指定することです。2 次構成ファイルの詳細は、13.2.2.4 項を参照してください。

ロガーは、接続が確立、切断、あるいは再確立するたびにイベントをポストすることで、リモート接続のステータスをレポートします。

例 13-6 に、`remote_hosts` セクションの例を示します。完全な構文は、`evmlogger.conf(4)` を参照してください。

例 13-6: リモート・ロギングを行うためのロガー構成ファイルのエントリの例

```
remote_hosts {  
    name                appsys_hosts                1  
    hostnames           appsys1,appsys2  
    hostnames           appsys3                    2  
    targets             appsys_log                  3  
    filter              "[priority >= 400]"        4  
    retry               10                          5  
}  
  
eventlog {  
    name                appsys_log                  6  
    logfile             /eventlogs/applog.dated  
    explicit_target     yes                        7  
    filter              all                        8  
}
```

1 name キーワードでこのグループを識別します。

例 13-6: リモート・ロギングを行うためのロガー構成ファイルのエントリの例 (続き)

- 2 hostnames 行にはロガーがサブスクライブするリモート・ホストのリストを指定します。このリストは複数行にわたって指定することができ、また各行に複数のホストを含めることもできます。
- 3 targets 行は、このグループにリストされたリモート・ノードから受信するイベントが、eventlog グループ appsys_log で定義されるイベント・ログに記録されることを示しています。ターゲットは、イベント・ログまたはフォワードのいずれかです。
- 4 filter 行は、任意のリモート・システムでポストされた、優先度が 400 以上のすべてのイベントのログがとられることを示しています。
- 5 retry 行は、任意のリモート・ホストへの接続が確立できないか切断された場合に、ロガーが 10 秒ごとに再接続を試みることを示しています。
- 6 これが、サンプルの remote_hosts セクションのターゲットになる eventlog グループです。
- 7 explicit_target キーワードに yes (または true) を設定すると、この eventlog グループが、target 行で名前を指定した remote_hosts グループを通じて受信したイベントのログだけを記録するようになります。ローカル・デーモンから受信したイベントは、このグループでは扱われません。
- 8 ログのフィルタに all を設定すると、リモート・ホストから受信したすべてのイベントのログが記録されます。remote_hosts グループで指定したフィルタ文字列によって、受信するイベントの集合が決まります。

構成を変更した場合、ロガーにその変更を認識させるために、`evmreload -l` コマンドを実行してください。

TruCluster 環境で `remote_hosts` セクションを指定した場合には、各種のクラスタ・ノードで実行されている個々のロガーが、同じセットのリモート接続を確立し、クラスタの各ノードで独立してイベントのログをとります。この動作を変更するには、`mkcdsl` コマンドを使用して、2 次構成ファイルのメンバ固有のバージョンを作成してください。CDSL (コンテキスト依存シンボリック・リンク) の詳細は、`mkcdsl(8)` を参照してください。

13.4 Event Manager のトラブルシューティング

Event Manager が正常に動作していない疑いがある場合、まず最初に `/var/evm/adm/logfiles` ディレクトリ内のメッセージ・ファイルを調べます。このファイルのメッセージはまた、Event Manager ビューアと `evmget` でも、`misclog` イベント・チャンネルの一部として表示されます。

次のリストでは、典型的な問題と、その解決のために最初に行うべきステップを示します。

- カーネル・イベントがポストされない

次のコマンドを使用して、Event Manager デーモン・ログ・ファイルにエラーが記録されていないかチェックします。

```
# more /var/evm/adm/logfiles/evmdaemon.log
```

次のコマンドを使用して、カーネル・インタフェースの擬似デバイスがあるかチェックします。

```
# ls -l /dev/kevm
```

この擬似デバイスがない場合、次のコマンドを使用して作成します。

```
# dsfmgr -vF
```

- サブスクライブする側のアプリケーションが、期待するイベントを受け取ることができない

次のコマンドを使って権限ファイルを調べ、ポスト側にこれらのイベントをポストする権限があるか確認します。

```
# more /etc/evm.auth
```

次のコマンドを使用して、イベントが登録されているか調べます。

```
# evmwatch -i -f '[name event_name]' |  
  evmshow -t "@name"
```

これによって、イベントが表示されない場合、`evmreload` を実行してもう一度検査します。 イベントがまだ表示されない場合は、テンプレート・ファイルが適切にインストールされているか調べます。

次のコマンドを使用して、サブスクライブ側がこれらのイベントにアクセスする権限があるか調べます。

```
# more /etc/evm.auth
```

次のコマンドを使用して、期待するイベントが実際にポストされているか調べます。

```
# evmwatch | evmshow -t "@name @"
```

イベントをポストするプログラムを実行してから、前述の `evmwatch` コマンドでそれらが適切に表示されるか確認します。

- ポストする側のプログラムがイベントをポストできない

次のコマンドを使用して、Event Manager デーモンが動作しているか調べます。

```
# ps -aef | grep evmd
```

次のコマンドを使って権限ファイルを調べ、ポスト側にこれらのイベントをポストする権限があるか確認します。

```
# more /etc/evm.auth
```

次のコマンドを使用して、イベントが登録されているか調べます。

```
# evmwatch -i -f '[name event_name]' |  
  evmshow -t "@name"
```

これによって、イベントが表示されない場合、evmreload コマンドを実行してもう一度検査します。イベントがまだ表示されない場合は、テンプレート・ファイルが適切にインストールされているか調べます。

- syslog または binlog イベントが Event Manager を通じて期待どおりに表示されない

syslog および binlog イベントにアクセスするためには、root としてログインするか、adm グループに所属していなければなりません。

省略時の設定では、Event Manager は過去 8 日以内にポストされた binlog イベントだけを取り出します。古い binlog イベントを参照したい場合は、チャンネル構成ファイル /etc/evmchannel.conf を編集します。binlog チャンネル・グループでは、省略時の fn_get 行に -r 8d オプションが含まれ、過去 8 日以内のイベントだけが取り出されます。このオプションを削除してすべての binlog イベントを参照することも、8 を別の値に変更することもできます。

ps コマンドを使用して、binlogd および syslogd デーモンが動作しているか調べます。

/etc/syslog_evm.conf ファイルが、表示させたいイベントを転送するように構成されているか調べます。

次のコマンドを使用して、syslog および binlog との通信をテストします。

```
# evmwatch | evmshow &#amp; logger "test syslog message"  
# logger -b "test binlog message"
```

- evmget またはイベント・ビューアを使ったイベントの取り出しが遅い
すべてのログ・ファイルのサイズ、特に evmlog ファイル (/var/evm/evmlog)、バイナリ・エラー・ログ (/var/adm/binary.errlog) および SysMan Station デーモン・ログ・ファイル (/var/adm/sysman/sysman_station/logs) のサイズを調べます。

ファイルのサイズは、`ls -L` コマンドを使ってリストします。これによって、シンボリック・リンクやコンテキスト依存シンボリック・リンク (CDSL) ではなく、このファイル自身を確認することができます。

バイナリ・ログのサイズの管理についての詳細は、`binlogd(8)` を参照してください。ただし、Event Manager がアーカイブ・ログ・ファイルからイベントを取り出すため、新しくログを開始しても Event Manager が利用できるイベントの数はすぐには減らないので注意してください。cron ユーティリティを使って、定期的にアーカイブ・タスクを実行することができます。evmlog ファイルのサイズは、`/etc/evmlogger.conf` ファイルと `/etc/evmchannel.conf` ファイル内の構成値を変更することで、小さくすることができます。

- 予期したイベントが記録されない

イベントの優先順位を検査します。優先順位が 200 以上のイベントだけが、Event Manager ロガーによって記録されます。

- リモート・デーモンを使ってイベントをポストまたはサブスクライブできない

リモート・アクセスが構成されているか確認し、構成されていない場合、次のことを行います。

1. リモート・デーモンの構成ファイルの `remote_connection` に `True` を設定します。
2. `/etc/evm.auth` ファイルにリモート・ホストが指定されていることを確認します。そのために、次のコマンドを実行します。

```
# grep remote-host /etc/evm.auth
```

3. 指定した認証タイプが `evm_callback` の場合は、一時的に `evm_open` に変更してください。これでうまく行けば、問題はリモート・ホストまたはユーザ・マッピングの指定にあります (ローカル・ユーザがないとか、必要なパーミッションをもっていないなど)。詳細は、13.2.3.3 項を参照してください。認証タイプを `evm_callback` に戻すのを忘れないでください。

4. 次のコマンドを実行します。

```
# evmreload -d
```

リモート接続を有効にする際は、セキュリティへの影響を考慮する必要があります。セキュリティについての詳細は、13.2.3 項を参照してください。

- フィルタが不正であるというメッセージがリモート・コネクションから返される

リモート・システムへ接続してイベントの取り出しまたは監視を行おうとしたときに、ローカル・システムで使用した場合には正しく動作するフィルタで、フィルタが不正であるというメッセージが返されることがあります。

フィルタ構文はオペレーティング・システムの新しいリリースでバージョン・アップされることがあり、新しいキーワードや省略形が古いバージョンでは認識されないことがあります。リモート・システムにログインし、`EvmFilter(5)` を見て、フィルタで使用した構文がそのオペレーティング・システムのバージョンでサポートされているか調べてください。

- `binlog` イベントが変換されない

変換ユーティリティが存在しない場合のトラブルシューティングには、次の手順を使用します。

1. 次のコマンドを実行します。

```
# usr/sbin/dia
```

DECevent がインストールされている場合は、このコマンドにより、現在のバイナリ・エラー・ログ・ファイル `/var/adm/binary.errlog` の変換後の内容が表示されます。

2. `dia` コマンドが見つからない場合は、次のコマンドを使用して、DECevent ソフトウェア・サブセット (配布キット) のステータスを調べます。

```
# setld -i | grep OSFDIA
```

このコマンドは、DECevent Base Kit (変換/分析) がインストールされている場合、文字列 `OSFDIABASE***` を返します。インストールされていない場合は、インストレーション・メディアをマウントし、`setld` コマンドを使ってこのサブセットをインストールします。詳細は、`setld(8)` のリファレンス・ページを参照してください。

Compaq Analyze があるかどうかは、次の手順で確認します。

1. 次のコマンドを実行して、Compaq Analyze director サービスがローカル・ホストで実行されているかどうかを調べます。

```
# ps agx | grep desta
```

2. desta デーモンが実行されていない場合は、Compaq Analyze ユーティリティがインストールされていて、実行されていないか、正しく構成されていない可能性があります。Compaq Analyze がインストールされているか確認するには、次のコマンドを実行してバイナリを探します。

```
# ls /usr/opt/compaq/svctools/bin/desta*
```

3. バイナリがない場合は、setld コマンドを使用して CD-ROM から Compaq Analyze をインストールします。Compaq Analyze の情報は、お近くの営業担当者にお問い合わせください。

- evmlogger: *number* イベントの受信ミス

このエラーは、イベントが受信バッファからオーバーフローした場合に発生します。受信バッファのサイズは、システム・ソケット・バッファの省略時の最大値に設定されています。この値を、以下のように変更します。

1. sysconfig コマンドを使用して、sb_max システム・パラメータの現在の値を調べます。

```
# sysconfig -q socket sb_max
```

2. 新しいバッファ・サイズを見積もります。
3. 次のコマンドを使用して、このパラメータの実行時の値を変更します。

```
# sysconfig -r socket sb_max=newvalue
```

この変更は、次にリブートされるまで有効です。

4. この変更を恒久的なものにするためには、sysconfigdb または dxkerneltuner を使用して、sb_max の値を変更します。



クラッシュ・ダンプの管理

この章では、システム・クラッシュ・ダンプの構成方法と作成方法、クラッシュ・ダンプとそれに関連するデータを保存する方法（グラフィカル・ユーザ・インタフェースを使用する方法と手動で行う方法）について説明します。クラッシュ・ダンプは、システムの予期しないシャットダウン時に自動的に取得される、実行中のカーネルのスナップショットです。クラッシュ・ダンプは、テクニカル・サポート担当者に連絡した際に、システム・クラッシュの原因となった問題の分析および修正のためによく参照されます。ただし、経験豊富なシステム管理者や開発者の場合は、クラッシュ・ダンプの解析方法に慣れていれば、自分でクラッシュ・ダンプの取得と解析を行うこともできます。

この章では、次のトピックについて説明します。

- クラッシュ・ダンプの概要（14.1 節）
- クラッシュ・ダンプの構成用と「スナップショット」の作成用の新しい2つのグラフィカル・ユーザ・インタフェース（14.2 節）
- クラッシュ・ダンプの作成方法（14.3 節）
- クラッシュ・ダンプの内容および方式の選択方法（14.4 節）
- 手動でクラッシュ・ダンプを取得する方法（14.5 節）
- クラッシュ・ダンプの保存およびアーカイブ方法（14.6 節）

14.1 クラッシュ・ダンプの概要

システムは予期しないシャットダウン時に、物理メモリ上のデータのすべてまたは一部を、(a) ディスクのスワップ領域（仮想メモリ領域）または (b) メモリのいずれかに書き込みます。このようなシャットダウンは、一般にシステム・クラッシュまたはパニックと呼ばれます。格納されたデータおよびステータス情報は、クラッシュ・ダンプと呼ばれます。クラッシュ・ダンプは、アプリケーションで生成された、コア・ダンプとは異なります。コア・ダンプの場合は、通常、出力後もシステムは動作を続けます。クラッ

シュ・ダンプ後は、システムはシャットダウンされてコンソール・プロンプト (>>>) が表示されます。 `auto_action` Boot Halt Restart オプションの指定に応じて、リブートが必要だったり、必要でなかったりします。

リブート処理の際に、システムはクラッシュ・ダンプをファイルに移動し、カーネルの実行可能イメージを他のファイルにコピーします。これらのファイルをまとめてクラッシュ・ダンプ・ファイルと呼びます。クラッシュ・ダンプ・ファイルは、システムがクラッシュしたときの分析のために、またはカスタム・カーネルの開発 (デバッグ) の際に必要です。システムの問題を分析するために、テクニカル・サポート部門へのクラッシュ・ダンプ・ファイルの提供が必要になる場合もあります。

クラッシュ・ダンプを管理するには、クラッシュ・ダンプ・ファイルがどのように作成されるかを理解する必要があります。クラッシュ・ダンプとクラッシュ・ダンプ・ファイル用に、ディスク上に空きスペースを確保しなければなりません。確保するスペースの量は、システム構成と選択したクラッシュ・ダンプのタイプによって異なります。

14.1.1 関連ドキュメントとユーティリティ

クラッシュ・ダンプでは、ディスク上に確保されている仮想メモリ・スワップ領域が使用されます。スワップ領域の管理については、第 3 章を参照してください。システム・イベント管理については、`binlogd` および `syslogd` イベント管理チャンネルの説明がある第 12 章を参照してください。

クラッシュ・ダンプの詳細と関連トピックは、マニュアルとリファレンス・ページを参照してください。

14.1.1.1 マニュアル

クラッシュ・ダンプや関連するトピックが記載されているマニュアルを、以下に示します。

- 『*Kernel Debugging*』では、クラッシュ・ダンプの分析方法について説明しています。この機能を使用するためには、ソフトウェア開発サブセットおよび適切なライセンスをインストールする必要があります。
- 『インストレーション・ガイド』では、インストレーション時に構成される初期スワップ領域とダンプ設定について説明しています。

14-2 クラッシュ・ダンプの管理

14.1.1.2 リファレンス・ページ

以下に示すリファレンス・ページには、関連ユーティリティに関する詳細情報があります。

| | |
|--|--|
| <code>savecore(8)</code> | コア・ダンプをスワップ・パーティションまたはメモリからファイルにコピーするプログラムです。 |
| <code>expand_dump(8)</code> | カーネル・クラッシュ・ダンプ・ファイルの圧縮を解除します。 |
| <code>dumpsys(8)</code> | システムを停止せずに、メモリのスナップショットをダンプ・ファイルにコピーします。これは継続ダンプとも呼ばれ、ダンプ構成を計画する際の、クラッシュ・ダンプ・サイズの見積りに役立ちます。 |
| <code>sysconfig(8)</code> および <code>sysconfigdb(8)</code> | カーネル・サブシステムの構成を保守し、カーネルにおけるクラッシュ・ダンプの属性を設定してクラッシュ動作を制御するために使用されるプログラムです。カーネル・チューナ・グラフィカル・ユーザ・インタフェース (<code>/usr/bin/X11/dxkerneltuner</code>) を使用してカーネル属性を変更することも可能です。詳細については、 <code>dxkerneltuner(8)</code> を参照してください。このインタフェースでは、オンライン・ヘルプも利用できます。カーネル・チューナは、CDE デスクトップから起動することもできます。これは、「アプリケーション・マネージャ：システム管理」フォルダ内にあります。 |
| <code>swapon(8)</code> | ページングおよびスワッピング用にファイルを追加するプログラムです。フル・ダンプを作成するために、一時または永続スワップ領域を追加する必要がある場合に、このコマンドを使用します。 |
| <code>dbx(1)</code> | ソース・レベル・デバッガです。 |

14.1.1.3 SysMan Menu アプリケーション

クラッシュ・ダンプの構成や作成を行うアプリケーションが、SysMan Menu から使用できます。

| | |
|---------------------|---|
| システム・ダンプ の構成 | このアプリケーションを使用して、 <code>savecore</code> コマンドに関連付けられた汎用システム構成変数を構成します。 |
| ダンプ・スナップ ショットの作成 | このアプリケーションを使用して、メモリのスナップショットを手動でダンプするための <code>dumpsys</code> コマンドを構成します。 |

詳細は、14.2 節を参照してください。

14.1.2 クラッシュ・ダンプ時に使用されるファイル

特に指定しなければ、`savecore` コマンドでは、クラッシュ・ダンプ・ファイルが `/var/adm/crash` ディレクトリにコピーされますが、指定した任意のファイル・システムにクラッシュ・ダンプをリダイレクトすることもできます。また、リモート・ホストにリダイレクトすることもできます。他の多くのシステム・ディレクトリと同じように、`/var/adm/crash` ディレクトリはコンテキスト依存シンボリック・リンク (CDSL) であり、システムとクラスタを結合する機能を果たします。このディレクトリの CDSL は `/var/cluster/members/member0/adm/crash` です。このディレクトリでは、次のファイルが作成または使用されます。

| | |
|---------------------------------------|--|
| <code>/var/adm/crash/bounds</code> | 次のダンプの増分番号を指定するテキスト・ファイル (<code>vmzcore.n</code> の <code>n</code>) |
| <code>/var/adm/crash/minfree</code> | クラッシュ・ダンプ・ファイルが書き込まれた後、最低何 KB 残すかを指定するファイル |
| <code>/var/adm/crash/vmzcore.n</code> | クラッシュ・ダンプ・ファイル。ファイルが圧縮されていない場合は、 <code>vmcore.n</code> という名前 (<code>z</code> がない) |

| | |
|--|---|
| <code>/var/adm/crash/vmunix.n</code> | クラッシュ時に実行していたカーネルの コピーで、通常は <code>/vmunix</code> |
| <code>/etc/syslog.conf</code> , <code>/etc/binlog.conf</code> , および <code>/etc/evmdaemon.conf</code> | ロギング構成ファイル |

14.2 クラッシュ・ダンプのアプリケーション

クラッシュ・ダンプの構成を簡単にするアプリケーションと、クラッシュ・ダンプ・ファイルを手動で作成するアプリケーションがあります。これらのアプリケーションは、SysMan Menu の「サポートとサービス」ブランチにあります。

最初のアプリケーションは、「システム・ダンプの構成」です。目的は、システム・ダンプのパラメータを構成して、将来クラッシュ・ダンプが発生したときに、適切な情報を得ることです。

2 番目のアプリケーションは、「ダンプ・スナップショットの作成」です。これを使用すると、各種のオプションを設定でき、クラッシュ・ダンプを作成するためにシステムを停止できない場合に、メモリのスナップショットをファイルに格納できます。

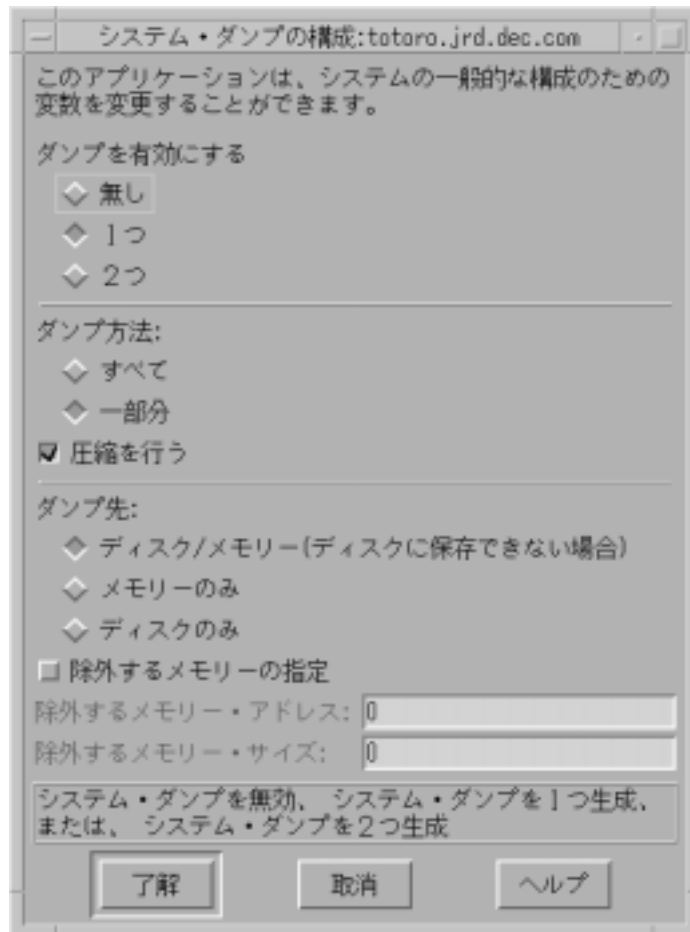
14.2.1 「システム・ダンプの構成」アプリケーションの使用法

「システム・ダンプの構成」アプリケーションを使用すると、クラッシュ・ダンプ・データを必要に応じてカスタマイズできます。このアプリケーションを使用すると、将来クラッシュ・ダンプが発生したときにクラッシュ・ダンプ・ファイルに影響する、各種のオプションを設定できます。

SysMan Menu で「サポートとサービス」を選択して、「ダンプの設定」を選択すると、このアプリケーションにアクセスできます。

図 14-1 に、このアプリケーションのメイン・ウィンドウを示します。

図 14-1: 「システム・ダンプの構成」アプリケーション



SysMan Menu でこのアプリケーションを起動すると、以下の情報を設定できます。

1. 1 番目の項目「ダンプを有効にする」では、次のいずれか 1 つを選択します。

「無し」

クラッシュ・ダンプを作成するメカニズムを無効にします。

「1 つ」

メカニズムを有効にして、クラッシュ・ダンプが発生したとき、クラッシュ・ダンプ・ファイ

ルが 1 セット (クラッシュ・ダンプ・ファイルとカーネルのコピー) 書き込まれるようにします。

「2 つ」

これも、メカニズムを有効にして、クラッシュ・ダンプが発生したとき、クラッシュ・ダンプ・ファイルが 1 セット書き込まれるようにします。このオプションでは、クラッシュ・ダンプ・ファイルの書き込み中にさらに追加のシステム障害が発生した場合に、追加のクラッシュ・ダンプ・ファイル・セットも書き込まれます。

2. 2 番目の項目では、フル・ダンプまたは部分ダンプの選択を行います。

フル・ダンプは、クラッシュ・ダンプ・ヘッダ情報とすべての物理メモリを保存します。

部分ダンプは、クラッシュ・ダンプ・ヘッダと物理メモリの一部のコピーを保存します。一部の物理メモリとは、システム・クラッシュの発生時点の重要な情報を含むと考えられる部分です。

3. 「圧縮を行う」チェック・ボックスをチェックすると、クラッシュ・ダンプ・ファイルを圧縮できます。特に理由がない限り、圧縮は必ず有効にしておきます。
4. 次の項目「ダンプ先」では、クラッシュ・ダンプ・データの保存方法を指定します。

「ディスク/メモリ
(ディスクに保存できない場合)」

クラッシュ・ダンプ・ファイルをディスクに保存します。これが失敗すると、圧縮された部分的なメモリ・ダンプが試みられます。

「メモリのみ」

クラッシュ・ダンプ・ファイルをメモリ領域に保存します。

「ディスクのみ」

クラッシュ・ダンプ・ファイルをディスクに保存します。これが失敗しても、圧縮された部分的なメモリ・ダンプは試みられません。

5. 最後の項目では、クラッシュ・ダンプ・ファイルを除外メモリにダンプするか、しないかを指定します。除外メモリにダンプする場合は、「除

外するメモリの指定」チェック・ボックスをチェックします。これにより、次の 2 つのフィールドに値が指定できるようになります。

「除外するメモリの アドレス」 ダンプの保存先の開始アドレスを指定します。

「除外するメモリ・ サイズ」 メモリ領域のサイズを指定します。

注意

これらのフィールドは、10 進数または 16 進数で指定できません。16 進数の場合には、先頭に 0x を付けてください。

「システム・ダンプの構成」アプリケーションには、詳細情報が記載されたオンライン・ヘルプがあります。

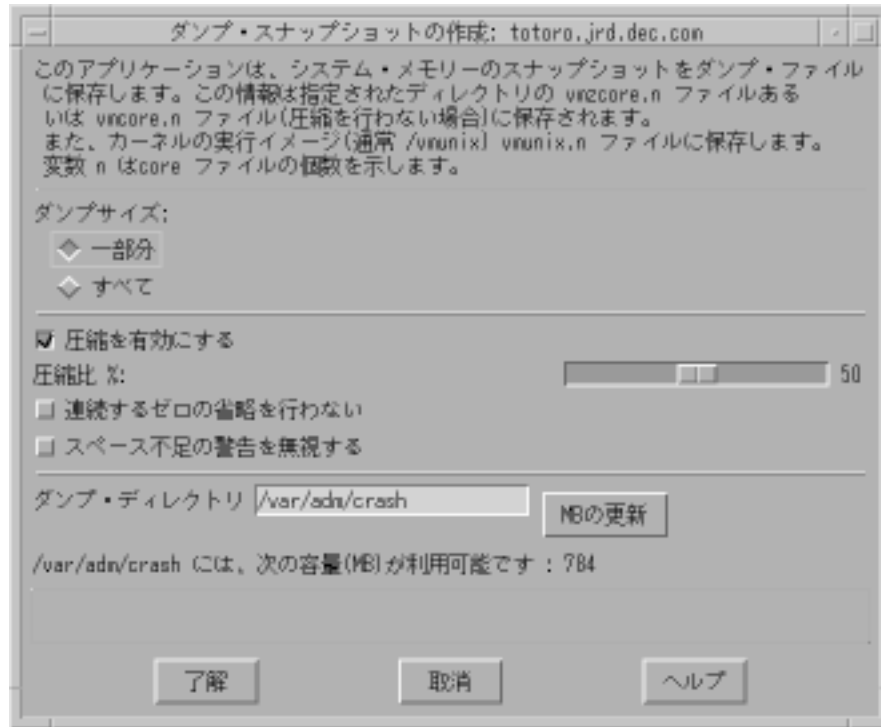
14.2.2 「ダンプ・スナップショットの作成」アプリケーションの使用 方法

図 14-2 で示した「ダンプ・スナップショットの作成」アプリケーションを使用すると、システム・メモリのスナップショットをダンプ・ファイルに保存できます。

SysMan Menu で「サポートとサービス」を選択して、「ダンプ・スナップショットの作成」を選択すると、このアプリケーションにアクセスできます。

図 14-2 に、このアプリケーションのメイン・ウィンドウを示します。

図 14-2: 「ダンプ・スナップショットの作成」アプリケーション



SysMan Menu でこのアプリケーションを起動すると、以下の情報を設定できます。

1. フル・ダンプと部分ダンプのどちらかを選択します。
2. データの圧縮を行うかどうかを指定します。圧縮する場合、「圧縮比 %」スライド・バーを使用して、圧縮率を指定します。小さな値を指定するほど、(可能な限り) 圧縮率が高くなります。
3. 隣接するゼロの圧縮を行うかどうかを、「連続するゼロの省略を行わない」チェック・ボックスで指示します。この圧縮はお勧めしません。
4. クラッシュ・ダンプ・データを保存するのに十分な領域がない場合でもアプリケーションによる警告が必要でない場合は、「スペース不足の警告を無視する」チェック・ボックスをチェックします。
5. 「ダンプ・ディレクトリ」フィールドに、クラッシュ・ダンプ・ファイルが書き込まれるディレクトリの絶対パス名を入力します。そのディレクトリで使用可能な領域のサイズが、「次の容量が利用可能です」

フィールドに M バイト単位で表示されます。その表示フィールドを更新するには、[MBの更新] を選択します。

「ダンプ・スナップショットの作成」アプリケーションには、詳細情報が記載されたオンライン・ヘルプがあります。

14.3 クラッシュ・ダンプの作成

システム・クラッシュの後、通常、コンソール・プロンプトから `boot` コマンドを入力してシステムをリブートします。システムのリブート中に、`savecore` コマンドは、クラッシュ・ダンプ情報をスワップ・パーティションまたはメモリからファイルに移動し、クラッシュ時に実行中だったカーネルを別のファイルにコピーします。これらのファイルを分析すれば、クラッシュの原因を究明するのに役立ちます。`savecore` コマンドはまた、クラッシュをシステム・ログ・ファイルに記録します。

`savecore` コマンドは、コマンド行から起動することもできます。詳細は、`savecore(8)` を参照してください。

14.3.1 generic サブシステムのダンプ・カーネル属性の設定

次のように `generic` サブシステムで定義されたカーネル属性を設定することで、クラッシュ・ダンプを行う方法を制御することができます。

| | |
|--------------------------------|--|
| <code>dump_savecnt</code> | クラッシュとリブートのシーケンス 1 回で生成される正常なクラッシュ・ダンプの数を制限するか、ダンプを使用不可にします。 14.3.2 項を参照してください。 |
| <code>dump_to_memory</code> | 1 次システム・コア・ダンプをメモリに書き込むかディスクに書き込むかを指定します。 14.3.2 項を参照してください。 |
| <code>dump_sp_threshold</code> | クラッシュ・ダンプが書き込まれるパーティションを制御します。省略時の値を使用すると、1 次スワップ・パーティションに十分に収まるサイズのクラッシュ・ダンプに対して、1 次スワップ・パー |

ティションが排他的に使用されます。
14.3.4 項を参照してください。

| | |
|--|---|
| <code>dump_user_pte_pages</code> | 部分クラッシュ・ダンプにユーザ・ページ・テーブルを含めるかどうかを指定します。この属性は、省略時はオフです。14.4.2 項を参照してください。 |
| <code>expected_dump_compression</code> | システムに通常達成させたい圧縮のレベルを指定します。省略時の設定は 500 ですが、0 ~ 1000 の整数を設定することができます。14.4.4 項を参照してください。 |
| <code>partial_dump</code> | 部分クラッシュ・ダンプとフル・クラッシュ・ダンプのどちらを取得するかを指定します。この属性は、省略時はオンです。14.4.3 項を参照してください。 |
| <code>compressed_dump</code> | スペースを節約するためにダンプを圧縮するかどうかを指定します。この属性は、省略時はオンです。オフを設定していても、別のダンプ属性の値により自動的にオンが設定されることがあります。14.4.5 項と 14.4.6 項を参照してください。 |
| <code>dump_kernel_text</code> | ダンプにカーネルのテキスト・ページを含めて大きなダンプ・ファイルを作成するかどうかを指定します。この属性は、部分ダンプが使用可能なときのみ適用されます。14.4.3 項を参照してください。 |
| <code>live_dump_dir_name</code> | 継続ダンプを書き込むディレクトリの完全パスを指定します。14.5.1 項を参照してください。 |

| | |
|--------------------------------------|--|
| <code>live_dump_zero_suppress</code> | 継続ダンプのゼロ圧縮を有効にするか無効にするかを指定します。ダンプ・ファイルの作成に時間がかかりますが、使用スペースは小さくなります。 14.5.1 項を参照してください。 |
|--------------------------------------|--|

除外メモリが使用可能な場合、除外メモリへのダンプが次の属性で制御されます。

| | |
|---------------------------------|--|
| <code>dump_exmem_addr</code> | 1 次ダンプの書き込みに使用する除外メモリ領域の開始アドレス (仮想または物理) を指定します。 |
| <code>dump_exmem_size</code> | ダンプを書き込む除外メモリ領域の大きさ (バイト) を指定します。 |
| <code>dump_exmem_include</code> | ダンプに除外メモリ・ページを含めるかどうかを指定します。 |

この機能の詳細は、14.4.6 項を参照してください。

次のコマンドは、一般的なダンプ属性の設定内容を表示します。

```
# sysconfig -q generic | grep dump
compressed_dump = 1
dump_exmem_addr = 0
dump_exmem_size = 0
dump_exmem_include = 0
dump_kernel_text = 0
dump_savecnt = 1
dump_sp_threshold = 4096
dump_to_memory = 0
dump_user_pte_pages = 0
expected_dump_compression = 500
live_dump_zero_suppress = 1
live_dump_dir_name = /var/adm/crash
partial_dump = 1
```

ダンプ属性とダンプ設定の詳細は、`sys_attrs_generic(5)` を参照してください。属性の値の設定の詳細は、`sysconfig(8)` と `sysconfigdb(8)` を参照してください。

14.3.2 クラッシュ・ダンプ・ファイルの作成

`savecore` コマンドは、リブート処理時に実行を始めると、クラッシュ・ダンプが発生したかどうか、およびファイル・システムにそれを保存するのに十分なスペースがあるかどうかを判断します (システムをシャットダウンしてリブートした場合、クラッシュ・ダンプは保存されません。すなわち、システムがクラッシュした場合のみ、クラッシュ・ダンプが保存されます)。

`dump_savecnt` 属性の値は、ダンプの数を制御します。使用できる値は次のとおりです。

(ゼロ)
(zero) クラッシュ・ダンプを生成することはありません。

1 1 次クラッシュ・ダンプを生成します (省略時の設定)。

2 2 次クラッシュ・ダンプを生成します。

`dump_to_memory` 属性の値は、ダンプの位置を制御し、次のように `dump_savecnt` 属性の値と連携して機能します。

-1 メモリへのダンプの書き込みを不可にします。この値は、`dump_savecnt` 属性の値が 2 のときに 2 次ダンプの書き込みも無効にします。

0 (zero) ディスク障害のイベント以外の場合は、ディスクにダンプが書き込まれます。ディスク障害の場合は、メモリに書き込まれます。これは、省略時の動作です。

1 十分なメモリがあるときは、ダンプはメモリだけに書き込まれます。例外は、2 次ダンプが有効になっている場合です (`dump_savecnt=2`)。詳細は、`sys_attrs_generic(5)` を参照してください。

場合によっては、メモリ内のダンプは重ね書きされる可能性があります。重ね書きを防ぐために、除外メモリと呼ばれるメモリの保護領域にダンプを書き込むこともできます。詳細は、14.4.6 項を参照してください。

クラッシュ・ダンプが存在し、ファイル・システムにクラッシュ・ダンプ・ファイルを保存するのに十分なスペースがある場合、`savecore` コマンドは、クラッシュ・ダンプとカーネルのコピーを省略時のクラッシュ・ディレクトリ (`/var/adm/crash`) 内のファイルに移動します (クラッシュ・ディレクトリの位置は変更できます)。

次の操作を選択できます。

- 14.4.7 項で説明しているように、ネットワーク接続を使って、すべてのクラッシュ・ファイルをリモート・ホストに書き込みます。
- 14.5.1 項で説明しているように、継続ダンプ・ファイルを代替ディレクトリに書き込みます。

`savecore` コマンドは、カーネル・イメージを `vmunix.n` ファイルに格納します。また、省略時の設定では、物理メモリの内容 (圧縮済み) を `vmzcore.n` ファイルに格納します。

変数 `n` は、クラッシュの番号を示します。このクラッシュの番号は、クラッシュ・ディレクトリの `bounds` ファイルに記録されます。最初のクラッシュが起こると、`savecore` コマンドは `bounds` ファイルを作成し、その中に数字の 1 を格納します。このコマンドは、その後、クラッシュが起こるたびに、その値をカウントアップします。

`savecore` コマンドは、コマンドの実行前にシステム・スワッピングがほとんどまたはまったく発生しないように、リブート処理の早い時期に実行されます。これにより、スワップによってクラッシュ・ダンプが壊されないことが保証されます。

14.3.3 クラッシュ・ダンプのロギング

`savecore` コマンドは、クラッシュ・ダンプ・ファイルを書き込むと、次の手順を行ってクラッシュをシステム・ログ・ファイルにロギングします。

1. リブート・メッセージを `/var/adm/syslog/auth.log` ファイルに書き込みます。

パニック状態のためにシステムがクラッシュした場合は、パニック文字列がログ・エントリに含まれます。

`syslog.conf` ファイルの `auth` ファシリティ・エントリを変更することにより、`savecore` コマンドが、リブート・メッセージを別のファイル

に書き込むように設定することができます。syslog.conf ファイルから auth エントリを削除すると、savecore コマンドはリブート・メッセージを保存しなくなります。

2. クラッシュ・ダンプのカーネル・メッセージ・バッファの保存を試みます。

カーネル・メッセージ・バッファには、クラッシュしたカーネルが作成したメッセージが入っています。これらのメッセージは、クラッシュの原因を究明する際に役立つことがあります。

省略時の設定では、savecore コマンドは、カーネル・メッセージ・バッファを /var/adm/crash/msgbuf.savecore ファイルに保存します。savecore がカーネル・メッセージ・バッファを書き込む位置は、/etc/syslog.conf ファイルの msgbuf.err エントリを変更することにより変更できます。/etc/syslog.conf ファイルの msgbuf.err エントリを削除すると、savecore は、カーネル・メッセージ・バッファを保存しなくなります。

その後、リブート処理で syslogd デーモンが起動し、msgbuf.err ファイルの内容を読み取り、/etc/syslog.conf ファイルに指定されているように、それらの内容を /var/adm/syslog/kern.log ファイルに移動します。次に syslogd デーモンが msgbuf.err ファイルを削除します。システム・ロギングの実行方法についての詳細は、syslogd(8) を参照してください。

3. クラッシュ・ダンプのバイナリ・イベント・バッファの保存を試みます。

バイナリ・イベント・バッファには、特にクラッシュがハードウェア・エラーに起因する場合、クラッシュの原因となった問題を把握するのに役立つメッセージが入っています。

省略時の設定では、savecore コマンドは、バイナリ・イベント・バッファを /usr/adm/crash/binlogdumpfile ファイルに保存します。savecore がバイナリ・イベント・バッファを書き込む位置は、/etc/binlog.conf ファイルの dumpfile エントリを変更することにより変更できます。/etc/binlog.conf から dumpfile エントリを削除すると、savecore はバイナリ・イベント・バッファを保存しなくなります。

その後、リブート処理で binlogd デーモンが起動し、/usr/adm/crash/binlogdumpfile ファイルの内容を読み取り、

/etc/binlog.conf ファイルに指定されているように、それらの内容を /usr/adm/binary.errlog ファイルに移動します。次に binlogd デーモンが binlogdumpfile ファイルを削除します。バイナリ・エラー・ロギングの実行方法についての詳細は、binlogd(8) を参照してください。

4. すべてのカーネル・イベントが処理されてポストされる前に、システムがクラッシュする可能性があります。そのような場合、savecore プログラムでそのようなイベントを回復させ、後で処理するために格納します。この回復は、そのようなイベントが利用可能で、savecore プログラムがイベントを正常に抽出して保存することができる場合のみ発生します。省略時の設定では、イベントは /var/adm/crash/evm.buf ファイルに格納されます。詳細は、savecore(8) と EVM(5) を参照してください。

14.3.4 スワップ領域

システムは、クラッシュ・ダンプをディスクに作成する際に、そのダンプをスワップ・パーティションに書き込みます。システムがスワップ・パーティションを使用するのは、スワップ・パーティションに格納されている情報が、実行中のシステムにとってのみ意味を持つからです。システムがクラッシュすると、この情報は無意味になるため、上書しても問題ありません。

システムはクラッシュ・ダンプを書き込む前に、/etc/sysconfigtab ファイルで定義されているスワップ・パーティションにダンプがどのように収まるかを判断します。たとえば、次の /etc/sysconfigtab ファイルのエントリでは、3 つのスワップ・パーティションが利用できることがわかります。

```
vm:
    swapdevice=/dev/disk/dsk0b, /dev/disk/dsk3h, /dev/disk/dsk13g
    vm-swap-eager=1
```

次に、システムがクラッシュ・ダンプを書き込む位置をどのように判断するかを示します。

- クラッシュ・ダンプが 1 次スワップ・パーティションに収まる場合は、/etc/sysconfigtab ファイルの swapdevice の下に列挙された最初のパーティションにダンプされます。システムは、可能な限りパーティションの後方からダンプを書き込み、パーティションの先頭をブート時スワッピングに利用できるよう残しておきます。

- クラッシュ・ダンプが、1 次スワップ・パーティションよりは大きい、2 次スワップ領域または 3 次スワップ領域に収まる場合、システムはクラッシュ・ダンプをそのスワップ・パーティション (/dev/disk/dsk3h および /dev/disk/dsk13g) に書き込みます。
- クラッシュ・ダンプが、利用可能なスワップ・パーティションのどれよりも大きい場合、システムは、2 次および 3 次のスワップ・パーティションをまたいで、それらのスワップ・パーティションが満杯になるまで、クラッシュ・ダンプを書き込みます。更にスペースが必要な場合は、残りのクラッシュ・ダンプ情報を 1 次スワップ・パーティションの最後から書き込みます (1 次スワップ・パーティションも満杯になる可能性があります)。
- スワップ・パーティションをすべて集めても、容量が小さすぎてクラッシュ・ダンプが収まらない場合、システムはクラッシュ・ダンプを作成しません。

クラッシュ・ダンプにはそれぞれヘッダがあり、システムはそれを常に 1 次スワップ・パーティションの最後書き込みます。ヘッダには、ダンプのサイズや格納位置についての情報が入っています。この情報によって、savecore は、システムのリブート時にダンプを見つけて保存することができます。

ほとんどの場合、圧縮したダンプが 1 次スワップ・パーティションに収まります。次の項では、dump_sp_threshold について説明します。この属性は、クラッシュ・ダンプの作成方法の理解に役立ちます。その他のカーネル属性は、ダンプの内容を制御します。これらの属性については、14.4 節で説明します。

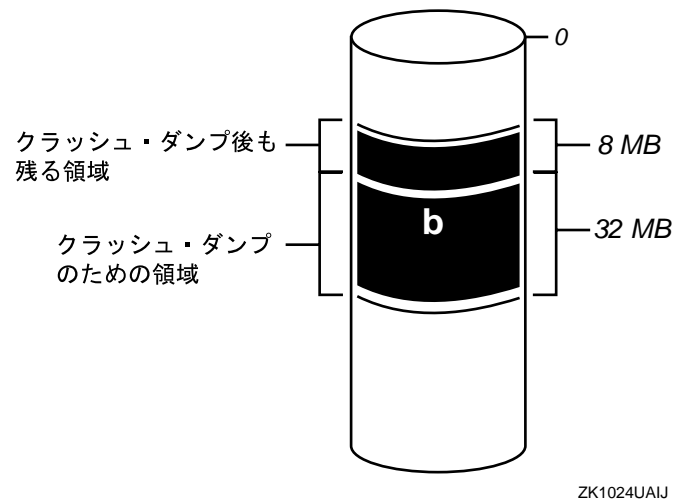
スワップ・パーティションの使用の制御

システムの構成により、1 次スワップ・パーティションに情報 (ダンプ・ヘッダを除く) を書き込む前に、2 次スワップ・パーティションにダンプ情報を書き込むようにすることができます。クラッシュ・ダンプを最初に書き込む位置を構成するために使用する属性は、dump_sp_threshold 属性です。

dump_sp_threshold 属性の値は、システムのリブート時のスワッピングのために通常用意しておく容量を示します。省略時の設定では、この属性は 16,384 ブロックです。つまり、システムはダンプの書き込み後に、1 次スワップ・パーティションに 8 MB のディスク容量が残るようにします。

図 14-3 には、40 MB のスワップ・パーティションに対する `dump_sp_threshold` の省略時設定を示します (40 MB という値は、ほとんどのシステムでは実際のスワップ・パーティションとして一般的な値ではありません。分かりやすくするために、例では小さい数を使用しています)。

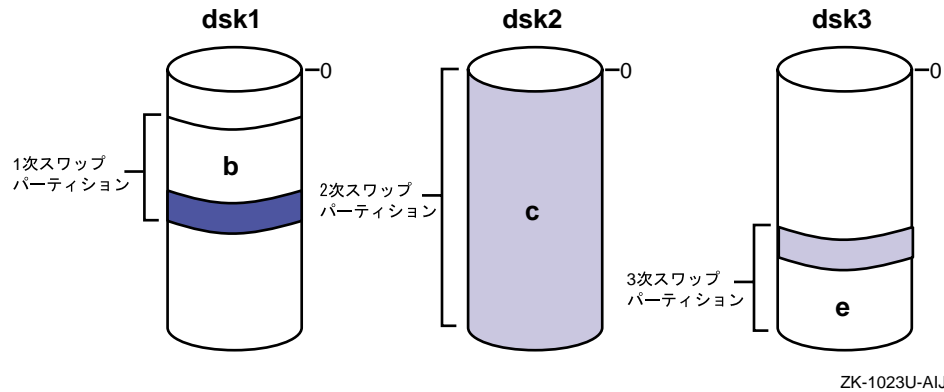
図 14-3: 省略時の `dump_sp_threshold` 属性の設定



システムは、図 14-3 に示す 1 次スワップ・パーティションに 32 MB のダンプ情報を書き込むことができます。したがって、30 MB のダンプは 1 次スワップ・パーティションに収まり、そのパーティションに書き込まれます。しかし、40 MB のダンプは大きすぎます。システムはクラッシュ・ダンプ・ヘッダを 1 次スワップ・パーティションに書き込み、クラッシュ・ダンプの残りは 2 次スワップ・パーティションが利用できればそこに書き込みます。

`dump_sp_threshold` の値を高く設定すると、システムは 1 次スワップ・パーティションにダンプ情報を書き込む前に、2 次スワップ・パーティションに書き込むようになります。たとえば、`dump_sp_threshold` 属性の値を 1 次スワップ・パーティションのサイズと同じ値にすると、システムは 2 次スワップ・パーティションに先に書き込みます (`dump_sp_threshold` 属性の設定については、14.4.1 項で説明しています)。図 14-4 には、複数デバイスの 2 次スワップ・パーティションにクラッシュ・ダンプがどのように書き込まれるかを示しています。

図 14-4: 複数のデバイスに書き込まれるクラッシュ・ダンプ



圧縮されていないクラッシュ・ダンプが図 14-4 のパーティション e を埋めた場合、システムは残りのクラッシュ・ダンプ情報を 1 次スワップ・パーティションの最後には書き込みます。システムは、ダンプ全体を格納するために必要なだけ 1 次スワップ・パーティションを使用します。システム・スワッピングから保護するため、ダンプは 1 次スワップ・パーティションの最後には書き込まれます。ただし、ダンプが 1 次スワップ・パーティション全体を埋めると、システム・リブート時のスワッピングによって壊れることがあります。

dumpsys を使用したクラッシュ・ダンプ・サイズの見積り

クラッシュ・ダンプのサイズを見積るには、`dumpsys` コマンドを使用して、実行時ダンプつまり続行可能なダンプを作成します。 `dumpsys` コマンドの使用方法についての詳細は、14.5.1 項を参照してください。一時的にファイル・システム領域を作成して、実験用のダンプを保持しなければならないこともあります。この方法では、フル・ダンプと部分ダンプの両方が作成できます。 `dumpsys -u` コマンド・オプションを指定しない限り、クラッシュ・ダンプは圧縮されます。 `dumpsys` コマンドの圧縮された出力から圧縮されていないダンプを作成するには、`expand_dump` コマンドを使います。

スワップに書き込まれるクラッシュ・ダンプは、その結果保存されるクラッシュ・ダンプ・ファイルとほぼ同じサイズなので、クラッシュ・ダンプ・ファイルのサイズを調べれば、クラッシュ・ダンプの大きさは簡単に判断できます。たとえば、システムが作成した最初のクラッシュ・ダンプ・ファイルの大きさを調べるには、次のコマンドを入力します。

```
# ls -ls /var/adm/crash/vmzcore.0
20480 vmzcore.0
```

このコマンドは、クラッシュ・ダンプ・ファイルに占有されている 512 バイトのブロックの数を表示します。この場合、ファイルは 20,480 個のブロックを占有しているので、スワップ・パーティションに書き込まれるクラッシュ・ダンプも約 20,480 ブロックだということがわかります。

場合によっては、システムに含まれるアクティブ・メモリが多いため、クラッシュ・ダンプを 1 つのディスクに格納できないことがあります。たとえば、システムに 2 GB のメモリが実装されているのに 4 GB のディスクが数台しかなく、そのほとんどがデータの格納に占有されているとします。このシステムのクラッシュ・ダンプは、大きすぎて 1 つのデバイスの 1 つのスワップ・パーティションには収まらないことがあります。クラッシュ・ダンプを複数のディスクに分散させるには、複数のディスクに 2 次 (そして、多分 3 次) スワップ・パーティションを作成します。システムは、大きすぎて 1 次スワップ・パーティションの指定部分に収まらないダンプを、自動的に別の使用可能なスワップ・パーティションに書き込みます。

14.3.5 クラッシュ・ダンプ領域の計画

クラッシュ・ダンプはシステムのスワップ・パーティションに書き込まれるため、スワップ・パーティションのサイズを調整して一時または永続スワップ領域を作成して、クラッシュ・ダンプの領域を確保します。スワップ・パーティションのサイズの変更についての詳細は、`swapon(8)` を参照してください。

注意

すべての永続スワップ・パーティションを、`/etc/sysconfigtab` ファイルにリストしてください。クラッシュ・ダンプをスワップ・パーティションからファイルにコピーする `savecore` コマンドは、`/etc/sysconfigtab` ファイルの情報を使用してスワップ・パーティションを見つけます。`/etc/sysconfigtab` ファイルに指定されていないスワップ・パーティションがある場合、`savecore` コマンドはそのパーティションを見つけられない可能性があります。

スペースの要件は、システムによって異なります。インストール処理では次のアルゴリズムを使い、`/var` ファイル・システムに必要なスペースを計算します。

$3 * \text{memsize} / 24\text{MB} + 3 * 15\text{MB}$

このとき、`memsize` は M バイト単位の物理メモリの容量で、15 MB はカスタム・カーネルのおおよその大きさです。このアルゴリズムでは、3 つのダンプを保存することが可能です。以降の項では、システム上の部分クラッシュ・ダンプおよびフル・クラッシュ・ダンプに必要なスペースを見積る際のガイドラインを示します。また、`dump_sp_threshold` 属性の設定についても説明します。

14.3.6 クラッシュ・ダンプ・ファイルに対するファイル・システム領域の割り当て

ご使用のシステムの標準的なクラッシュ・ダンプ・サイズを使用して、`/var/adm/crash` ディレクトリに必要なファイル・システム領域の計画と割り当てを行うことができます。

たとえば、部分クラッシュ・ダンプを保存するとします。また、システムには 96 MB のメモリがあり、クラッシュ・ダンプとスワッピングに 85 MB のディスク・スペースを予約しているとします。この場合、クラッシュ・ダンプ・ファイルの格納には 20 MB の領域を予約してください。複数のクラッシュ・ダンプをファイルに保存するには、さらに多くの領域が必要です。複数のクラッシュ・ダンプをファイルに保存する場合は、古いクラッシュ・ダンプをアーカイブすることを検討してください。クラッシュ・ダンプ・ファイルのアーカイブについての詳細は、14.6 節を参照してください。

特に指定しなければ、`savecore` コマンドは `/var/adm/crash` ディレクトリにクラッシュ・ダンプを書き込みます。省略時のディレクトリにクラッシュ・ダンプ・ファイルの領域を予約するには、ディスク・スペースが十分にあるファイル・システムを `/var/adm/crash` ディレクトリにマウントしなければなりません (ファイル・システムのマウントについての詳細は、第 6 章と `mount(8)` を参照してください)。クラッシュ・ダンプ・ファイルが大きくなることが予測される場合、LSM (Logical Storage Manager) ボリュームを使用してクラッシュ・ダンプ・ファイルを格納しなければならないこともあります。LSM ボリュームの作成についての詳細は、『*Logical Storage Manager*』を参照してください。

ディスク・スペース不足でシステムがクラッシュ・ダンプ・ファイルを保存できない場合、システムはシングルユーザ・モードに戻ります。シングルユーザ・モードに戻ることで、システム・スワッピングによるクラッシュ・ダンプ破壊が防止できます。いったんシングルユーザ・モードになる

と、クラッシュ・ディレクトリに必要な領域を確保したり、クラッシュ・ディレクトリを変更したりすることができます。この状況で考えられる方法の1つとして、シングルユーザ・モードのプロンプトでの `savecore` コマンドの実行があります。コマンド行で、十分な領域を持つディレクトリの名前を指定して、クラッシュ・ダンプ・ファイルを保存します。たとえば、次の `savecore` コマンドは、クラッシュ・ダンプ・ファイルを `/usr/adm/crash2` ディレクトリに書き込みます。

```
# savecore /usr/adm/crash2
```

`savecore` コマンドがいったんクラッシュ・ダンプ・ファイルを保存すると、システムはマルチユーザ・モードで起動できるようになります。または、ネットワークを起動して、`ftp` コマンドを使用し、別のホストにリモート・ダンプが行えるようになります。

`savecore` コマンド行でディレクトリを指定すると、クラッシュ・ディレクトリは、そのコマンド実行時に限り変更されます。その後システムがクラッシュし、システムのスタートアップ・スクリプトが `savecore` スクリプトを起動すると、`savecore` コマンドはクラッシュ・ダンプを省略時の `/var/adm/crash` ディレクトリにコピーします。

クラッシュ・ディレクトリの省略時の位置は、`rcmgr` コマンドで `SAVECORE_DIR` 変数を設定して制御できます。たとえば、省略時の動作で (システムのスタートアップ時に) クラッシュ・ダンプ・ファイルを `/usr/adm/crash2` ディレクトリに保存するには、次のコマンドを実行します。

```
# /usr/sbin/rcmgr set SAVECORE_DIR /usr/adm/crash2
```

クラッシュ・ダンプを保存したかどうかにかかわらず、システムをマルチユーザ・モードに戻したいときには、次のコマンドを実行します。

```
# /usr/sbin/rcmgr set SAVECORE_FLAGS M
```

14.4 クラッシュ・ダンプの内容と方式の選択

省略時の設定では、クラッシュ・ダンプは圧縮された部分ダンプですが、必要に応じてフル・ダンプや圧縮されていないダンプ、またはフル・ダンプで圧縮されていないダンプも可能です。通常、部分クラッシュ・ダンプは、クラッシュの原因を判断するために必要な情報を提供します。しかし、クラッシュが繰り返し発生し、クラッシュの原因を究明するのに部分クラッシュ・

ダンプでは役に立たない場合には、システムにフル・クラッシュ・ダンプを生成させることをお勧めします。

部分クラッシュ・ダンプには、次の情報が入っています。

- クラッシュ・ダンプ・ヘッダ
- 物理メモリの一部のコピー

システムは、物理メモリの一部で、システム・クラッシュの時点で重要な情報を含むと思われる部分、すなわち基本的には、カーネル・ノードのコードおよびデータを書き込みます。省略時の設定では、システムはユーザ・ページ・テーブルのエントリを省略します。

フル・クラッシュ・ダンプには、次の情報が入っています。

- クラッシュ・ダンプ・ヘッダ
- クラッシュの時点における物理メモリの全内容のコピー

クラッシュ・ダンプの取得方法は、次のように変更できます。

- クラッシュ・ダンプのしきい値を調整する。
- 省略時の指定を変更し、ユーザ・ページ・テーブルのエントリが部分クラッシュ・ダンプに書き込まれるようにする。
- 部分クラッシュ・ダンプかフル・クラッシュ・ダンプかを選択する。
- ダンプ圧縮の期待値を変更する。
- 圧縮されたクラッシュ・ダンプか非圧縮のクラッシュ・ダンプかを選択する。

これらのオプションについては、以降の項で説明します。

14.4.1 1 次スワップ・パーティションにおけるクラッシュ・ダンプのしきい値の調整

小さなクラッシュ・ダンプでも、1 次スワップ・パーティションに書き込む前に 2 次スワップ・パーティションに書き込むようにシステムを構成するには、`dump_sp_threshold` 属性に大きな値を設定します。14.3 節で説明するように、この属性に割り当てる値は、通常、システム・クラッシュの後、システム・スワップのために利用できる領域の大きさを示します。

`dump_sp_threshold` 属性を調整するには、`sysconfig` コマンドを実行します。たとえば、1 次スワップ・パーティションが 40 MB だとします。シ

システムがクラッシュ・ダンプを 2 次スワップ・パーティションに書き込むように値を大きくするには、次のコマンドを実行します。

```
# sysconfig -r generic dump_sp_threshold=81920
```

この例では、generic サブシステムの dump_sp_threshold 属性に 81,920 ブロック (512 バイト・ブロック) (40 MB) が設定されます。この例では、システムは 1 次スワップ・パーティション全体をシステム・スワップ用に残そうとします。システムは、自動的に、クラッシュ・ダンプを 2 次スワップ・パーティションに書き込み、クラッシュ・ダンプ・ヘッダを 1 次スワップ・パーティションの最後に書き込みます。

sysconfig コマンドは、現在実行中のカーネルのシステム属性値を変更します。dump_sp_threshold 属性の新しい値を sysconfigtab データベースに格納するには、sysconfigdb コマンドを使用して、データベースを変更します。sysconfigtab データベースと sysconfigdb コマンドについての詳細は、sysconfigdb(8) を参照してください。

注意

savecore プログラムがクラッシュ・ダンプをファイルにコピーするとすぐに、すべてのスワップ・デバイスがマウントやスワップに使用できるようになります。スワップ領域の共用は、ブート時の短時間だけ、通常は物理メモリの少ないシステムで発生します。

14.4.2 部分クラッシュ・ダンプにユーザ・ページ・テーブルを含める

省略時の設定では、部分クラッシュ・ダンプにはユーザ・ページ・テーブルは出力されません。これらのテーブルは、通常はクラッシュ・ダンプの原因究明に役立たないため、省略してクラッシュ・ダンプおよびクラッシュ・ダンプ・ファイルのサイズを削減します。ただし、テクニカル・サポート担当者から、クラッシュ・ダンプ解析用にユーザ・ページ・テーブルを含めるように依頼されることがあります。

部分クラッシュ・ダンプにユーザ・ページ・テーブルを含めるには、dump_user_pte_pages 属性の値に 1 を設定します。dump_user_pte_pages 属性は、generic サブシステムにあります。次の例は、この属性を設定するために実行するコマンドを示します。

```
# sysconfig -r generic dump_user_pte_pages = 1
```

`sysconfig` コマンドは、現在実行中のカーネルのシステム属性の値を変更します。`sysconfigtab` データベースに `dump_user_pte_pages` 属性の新しい値を格納するには、`sysconfigdb` コマンドを使用してデータベースを変更するか、カーネル・チューナ GUI (`dxkerneltuner`) を使用します。

ユーザ・ページ・テーブルを部分クラッシュ・ダンプに書き込まないという省略時の設定に戻すには、`dump_user_pte_pages` 属性に 0 (ゼロ) を設定します。

14.4.3 部分クラッシュ・ダンプかフル・クラッシュ・ダンプかの選択

省略時、システムは部分クラッシュ・ダンプを生成します。フル・クラッシュ・ダンプを生成するようにしたい場合は、次のようにカーネルの `partial_dump` 変数に 0 (ゼロ) を設定すると、省略時の動作を変更できます。

```
# sysconfig -r generic partial_dump=0
partial_dump: reconfigured
# sysconfig -q generic partial_dump
generic:
partial_dump = 0
```

カーネル・チューナ GUI または `sysconfigdb` コマンドを使用すると、カーネル・エントリを変更し、リブート後も変更を保持することができます。部分クラッシュ・ダンプに戻すには、`partial_dump` 変数に 1 を再設定します。

部分ダンプが有効になっているときは、`dump_kernel_text` 属性も有効にしてカーネル・テキスト・ページを含めるようにすることができます。

14.4.4 ダンプ圧縮の期待値

`expected_dump_compression` 変数は、ダンプにどの程度の圧縮を期待するかを示すために使用します。省略時は `expected_dump_compression` の値は 500 で、指定可能な最小値 0 (ゼロ) と最大値 1000 の中間に設定されています。次の手順では、システムに適した `expected_dump_compression` 変数を算出する方法を説明します。

1. 14.5.1 項で説明するように、`dumpsys` コマンドを使用して圧縮ダンプを作成します。`ls -s` コマンドを使用して、このダンプのサイズを書き留めます (値 `a` とします)。

2. `expand_dump` コマンドを使用して、圧縮されていないバージョンのダンプを作成します。 `ls -s` コマンドを使用して、このダンプのサイズを書き留めます (値 `b` とします)。
3. `a` を `b` で割り、圧縮率の近似値を算出します。
4. 以上の手順を何度か繰り返して、圧縮率の最大値を選択します。圧縮率に 1000 を掛けて、期待するダンプの値を算出します。
5. 期待するダンプの値の 10 パーセントを足して `expected_dump_compression` 変数の値にします。

次のように、`sysconfig` コマンドを使用して、カーネルの `expected_dump_compression` 変数に、必要な値を設定します。

```
# sysconfig -r generic expected_dump_compression=750
expected_dump_compression: reconfigured
# sysconfig -q generic partial_dump
generic:
expected_dump_compression=750
```

カーネル・チューナ GUI または `sysconfigdb` コマンドを使用すると、カーネル・エントリを変更し、リブート後も変更を保持することができます。

14.4.5 圧縮されていないクラッシュ・ダンプの選択と使用

省略時の設定では、クラッシュ・ダンプはディスク・スペースを節約するために圧縮されます。これにより、小さなパーティションに大きなクラッシュ・ダンプ・ファイルを書き込むことができます。これは、特にリアルタイム操作でのスワップを抑制するようにシステムをチューニングするために、大容量の物理メモリを持たせたシステムで大きな利点となります。クラッシュの後のリブート時に、`savecore` コマンドは自動的に起動され、スワップ・パーティション内のクラッシュ・ダンプ・ヘッダの情報を使用して、ダンプが圧縮されていることを検出します。次に、クラッシュ・ダンプ・ファイルがスワップ・パーティションから `/var/adm/crash` ディレクトリにコピーされます。圧縮されたクラッシュ・ダンプ・ファイルでは、ファイル名中に文字 `z` が付加され、圧縮されていないクラッシュ・ダンプ・ファイルと区別されます。たとえば、`vmzcore.1` のようになります。

このタイプの圧縮クラッシュ・ダンプ・ファイルは、`dbx` などのデバッグ・ツールで扱うことができます (`compress` や `gzip` などのツールで作成された圧縮ファイルでは使用できません)。圧縮されたクラッシュ・ダンプ・ファイルをサポートしていないツールを使用する場合は、`expand_dump` ユーティ

リティによって、従来の圧縮されていないフォーマットに変換できます。次の例では、`expand_dump` ユーティリティの使用方法を示します。

```
# expand_dump vmzcore.2 vmcore.2
```

圧縮されたフォーマットでは正しく動作しないツールやスクリプトを常使用する場合や、`expand_dump` コマンドが使用しづらい場合は、ダンプの圧縮を行わないようにできます。ダンプの圧縮を行わないようにするには、次のように `sysconfig` コマンドを使用します。

```
# sysconfig -r generic compressed_dump=0
```

このコマンドは、ダンプのモードを非圧縮へ一時的に変更します。次にリブートすると、モードは圧縮ダンプに戻ります。変更を持続的にするには、`sysconfigdb` コマンドを使用して `/etc/sysconfigtab` ファイル内の `compressed_dump` 属性の値を変更するか、カーネル・チューナ GUI を使用して `generic` サブシステムの値を変更します。

注意

メモリ・ダンプは圧縮されます。`compressed_dump` システム属性を設定していない場合、メモリ・ダンプを書き込む前にシステムは自動的に圧縮を有効にします。

クラッシュ・ダンプの圧縮と非圧縮クラッシュ・ダンプ・ファイルの作成方法については、`savecore(8)`、`expand_dump(8)`、および `sysconfig(8)` を参照してください。

14.4.6 除外メモリへのダンプ

除外メモリは、特別な目的で確保されている、物理メモリの領域です。メモリの除外領域は、`/etc/sysconfigtab` ファイルに指定して作成することができます。指定すると、システムのブート時に除外領域が作成されます。たとえば次のように行います。

```
cma_dd:
  CMA_Option = Size-0x3000000, Alignment - 0, /
  Addrlimit - 0x4000000, Type - 0x96, Flag-0
```

この `/etc/sysconfigtab` ファイルのエントリは、48 MB の大きさの除外メモリの領域を予約します。その `Type` は、値 `0x96` により `M_EXEMPT` として指定されます。`Addrlimit` の値は、除外領域の開始位置を設定します。この場合は `0x4000000` で、物理メモリの 64 MB の位置になります。マシ

ンをブートするたびに、この同じ物理メモリの領域が予約され、その他の目的では使用できなくなります。

メモリの除外領域を作成する方法には、擬似デバイス・ドライバ内で、タイプ `M_EXEMPT` を指定した `contig_malloc()` 関数呼び出しを使うという別の方法もあります。 `M_EXEMPT` タイプについての詳細は、`malloc.h` ファイルを参照してください。関数呼び出しの使用の詳細は、`contig_malloc(9r)` を参照してください。

`vmstat` コマンドに `-M` オプションを付けて実行し、除外メモリ領域を調べることができます。

除外メモリにダンプするには、14.3.2 項で示すように `dump_to_memory` 属性を有効にする必要があります。必要に応じて、次の属性も構成します。

| | |
|---------------------------------|--|
| <code>dump_exmem_size</code> | ダンプを書き込む除外メモリ領域の大きさ (バイト単位) を指定します。省略すると、値は 0 (ゼロ) となり、除外メモリ領域にダンプを書き込むことができません。 |
| <code>dump_exmem_addr</code> | 1 次ダンプの書き込みに使用する除外メモリの領域の開始アドレス (仮想または物理) を指定します。 |
| <code>dump_exmem_include</code> | ダンプに除外メモリ・ページを含めるかどうかを指定します。省略すると、値は 0 (ゼロ) となり、除外メモリ・ページは除外されます。 |

`dump_exmem_addr` 属性を設定しても、`dump_exmem_size` 属性も構成しない限り、有効になりません。属性の実行時設定を記録し、システム障害からの回復後にクラッシュ・ダンプを見つけることができるようにしてください。

次の例では、これらの属性の再構成方法を示しています。

```
# sysconfig -q generic dump_to_memory
generic:
dump_to_memory = 0
# sysconfig -r generic dump_to_memory=1
dump_to_memory: reconfigured
# sysconfig -q generic dump_to_memory
generic:
```

```
dump_to_memory = 1
```

省略時の設定では、メモリ・ダンプは圧縮されます。 `compressed_dump` システム属性にオンを設定していない場合は、その属性は自動的に有効になります。 `savecore` コマンドは `vmzcore` 文字型特殊デバイス・ファイルを使い、圧縮されたダンプを復元します。詳細は、`savecore(8)` と `vmzcore(7)` を参照してください。

14.4.7 リモート・ホストへのダンプ

`savecore` コマンドに `-r` オプションを付けて使用し、`ftp` 接続を使用してクライアント・ホストからリモート・ホストへクラッシュ・ダンプ・ファイルを書き込みます。リモートの宛先用に次の定義のどちらかを指定することができます。

- リモート・ホストの名前と有効なアカウントとパスワード
- `ftp` 接続とログイン情報の入っている構成ファイルへのパス

たとえば、次のコマンドは、`ftp` 接続をデバッグすることができる冗長モードで、リモート・ホストへのログインを指定します。

```
# savecore -v -r soserv:jeffdump:Cr$hDeBuG
```

ターゲット・ホストへ接続するとき、`savecore` ユーティリティはリモート `ftpd` サーバ・デーモンに、クライアントのホスト名が前に付いた名前でディレクトリを作成するように指示します。クラッシュ・ダンプ・ファイル (`bounds`, `msgbuf.savecore`, `evm.buf`, `vmunix.N`, および `vmcore.N` または `vmzcore.N`) がこのディレクトリに書き込まれます。リモート・デバイス上に、クラッシュ・ダンプ用のスペースを十分確保しなければなりません。

この機能の詳細と、この機能を使用するときの制限については、`savecore(8)` と `ftpd(8)` を参照してください。

14.5 手動によるクラッシュ・ダンプの生成

以降の項では、次の2つの状況でクラッシュ・ダンプを手動で作成する方法について説明します。

継続ダンプ

`dumpsys` コマンドを使用して、システムを停止せずに、実行中のメモリのスナップショットをコピーします (システムは動作し続けます)。

強制ダンプ コンソール・コマンド `crash` を使用して、応答しない (ハングした) システムのクラッシュ・ダンプ・ファイルを作成します。

クラッシュ・ダンプ・ファイル用の領域が適切に計画されており、カーネル・パラメータがこれまでの項で説明したように設定されているものとします。

14.5.1 実行中のシステムでの継続ダンプ

システムを停止させて通常のクラッシュ・ダンプを取得することができない場合は、`dumpsys` コマンドを使用してメモリのスナップショットをダンプします。システムは `dumpsys` コマンドがスナップショットを取得している間も動作するため、メモリ内容をコピーしている最中にそのメモリが変更されることもあります。結果として生成されたダンプを分析すると、不完全にリンクされたりリストや、部分的にゼロクリアされたページなどが存在することがあります。これらは障害ではなく、メモリの一時的な状態を示しています。このため、`dumpsys` コマンドを使用してもシステムの問題を検出できず、14.5.2 項で説明するように、システムを停止させて強制的にクラッシュ・ダンプを生成しなければならないことがあります。省略時の設定では、`dumpsys` コマンドはクラッシュ・ダンプを `/var/adm/crash` ディレクトリに書き込みます。

`/var/adm/crash/minfree` テキスト・ファイルは、`dumpsys` コマンドがダンプをコピーした後、ファイル・システムに残す最小限の容量を KB 単位で指定します。省略時の設定ではこのファイルは存在せず、最小値は設定されません。最小値を指定するには、ファイルを作成し、予約する容量を KB 単位の数値で指定します。`minfree` ファイルでの設定は、`-i` オプションを使用して置き換えることができます。`-s` オプションを指定すると、フル・ダンプおよび部分ダンプに必要なディスク・ブロックの概算値が表示されます。正確な値は、次の理由で事前に把握することはできません。

- 省略時の設定では、実際のダンプは、連続するゼロの書き込みを抑制してディスク・スペースを最適化するため (非圧縮ダンプの場合のみ)。
- システムによるカーネル動的メモリの使用状況 (`malloc/free`) は、システムの動作によって変化するため。
- 格納に必要な間接ディスク・ブロックの数が未知であるため。

次の例は、512 KB の物理メモリを持つシステムでのダンプを示します。この例では、圧縮されないクラッシュ・ダンプを示しています。特に指定しなければ、ダンプは圧縮されます。

```
# dumphsys -s
Approximate full dump size = 1048544 disk blocks,
  if compressed, expect about 524272 disk blocks.
Approximate partial dump size = 94592 disk blocks,
  if compressed, expect about 47296 disk blocks.

# dumphsys -i /userfiles
Saving 536797184 bytes of image in /userfiles/vmzcore.0
# ls /userfiles
bounds vmzcore.0 vmunix.0
```

generic カーネル・サブシステム内の 2 つの属性を使用すると、継続ダンプを制御できます。

| | |
|-------------------------|--|
| live_dump_dir_name | ディレクトリへのパスを指定します。このディレクトリには、継続ダンプ・ファイルが書き込まれます。省略時の値は、 <code>/var/adm/crash</code> ディレクトリです。 |
| live_dump_zero_suppress | 継続ダンプのゼロ圧縮の有効/無効を指定します。このオプションを有効にすると、ファイルの作成に時間がかかりますが、スペースは小さくなります。 |

詳細は、`dumphsys(8)` と `sys_attrs_generic(5)` を参照してください。継続クラッシュ・ダンプの分析についての詳細は、『*Kernel Debugging*』を参照してください。

14.5.2 ハングしたシステムでの強制クラッシュ・ダンプ

システムがハングしたときに、強制的にクラッシュ・ダンプを行うようにすることができます。ほとんどのハードウェア・プラットフォームでは、次の手順でクラッシュ・ダンプを強制的に行います。

1. システムに停止 (Halt) ボタンの使用可能/使用不能を設定するスイッチがある場合、そのスイッチを使用可能な位置に設定します。
2. 停止ボタンを押します。

3. コンソール・プロンプトで、`crash` コマンドを入力します。

システムによっては、停止ボタンのないものもあります。その場合は、次の手順を実行して、ハングしたシステム上でクラッシュ・ダンプを強制的に行います。

1. コンソール・プロンプトで `Ctrl/p` を入力します。
2. コンソール・プロンプトで、`crash` コマンドを入力します。

システムがハングしたときにクラッシュ・ダンプを強制的に行うと、クラッシュ・ダンプに記録されるパニック文字列は次のようになります。

```
hardware restart
```

停止ボタンを押すか `Ctrl/p` を入力することによってシステムの動作に割り込みをかけると、このパニック文字列が必ず記録されます。

14.6 クラッシュ・ダンプ・ファイルの保存とアーカイブ

圧縮された (`vmzcore.n`) クラッシュ・ダンプ・ファイルのみで作業している場合は、効率良く保存するためにすでに十分圧縮されています。以降の項では、特殊な場合について説明します。

14.6.1 項では、保存や転送のために次の状況でファイルを圧縮する方法について説明します。

- 圧縮されていない (`vmcore.n`) クラッシュ・ダンプ・ファイルで作業している場合
- 可能な限り最大の圧縮が必要な場合 – たとえば、低速の伝送回線でクラッシュ・ダンプ・ファイルを伝送する必要がある場合

14.6.2 項では、`vmcore.n` ファイルを圧縮して作成された、部分クラッシュ・ダンプ・ファイルの圧縮を解除する方法について説明します。

14.6.1 クラッシュ・ダンプ・ファイルの圧縮

`vmcore.n` クラッシュ・ダンプ・ファイルを圧縮するには、`gzip`、`compress`、または `dxarchiver` などのユーティリティを使用します。たとえば、次のコマンドでは、`vmcore.3.gz` という名前の圧縮ファイルが作成されます。

```
# gzip vmcore.3
```

クラッシュ・ダンプ・ファイル `vmzcore.n` は特別な圧縮方式を使用しているため、圧縮解除しなくても、デバッガやクラッシュ分析ツールで読み取ることができます。 `vmzcore.n` ファイルは、同じ内容の `vmcore.n` ファイルと比べると相当圧縮されていますが、`vmcore.n` ファイルを `gzip` のような標準の UNIX 圧縮ユーティリティを使用して圧縮した場合ほどには圧縮されません。 `vmzcore.n` ファイルに対して標準の圧縮を適用すると、ファイルの大きさは同じ内容の `vmzcore.n` ファイルよりも 40% 程度小さくなります。

`vmzcore.n` ファイルに、可能な限り最大の圧縮を適用する必要がある場合は、次の手順を実行します。

1. `expand_dump` コマンドを使用して、`vmzcore.n` ファイルを圧縮解除します (`expand_dump(8)` を参照)。次の例では、`vmzcore.3` ファイルから `vmcore.3` という名前の圧縮されていないファイルが作成されます。

```
# expand_dump vmzcore.3
```

2. 作成された `vmcore.n` ファイルを、標準の UNIX ユーティリティを使用して圧縮します。次の例では、`gzip` コマンドを使用して、`vmcore.3.gz` という名前の圧縮ファイルが作成されます。

```
# gzip vmcore.3
```

`vmzcore.n` ファイルは `expand_dump` コマンドでのみ圧縮解除できます (`gunzip`, `uncompress`, その他のユーティリティは使用できません)。
`expand_dump` コマンドを使用して `vmzcore.n` ファイルを `vmcore.n` ファイルに圧縮解除すると、`vmzcore.n` ファイルに圧縮し直すことはできません。

14.6.2 部分クラッシュ・ダンプ・ファイルの圧縮解除

この項は、`vmcore.n` ファイルから圧縮された部分クラッシュ・ダンプ・ファイルの圧縮解除の場合にのみ適用されます。

`vmcore.n` ファイルから圧縮した部分クラッシュ・ダンプ・ファイルを圧縮解除する際は注意が必要です。オプションを指定しないで `gunzip` または `uncompress` コマンドを使用すると、メモリ・サイズと同じ格納スペースを必要とする `vmcore.n` ファイルが作成されます。つまり、圧縮解除されたファイルには、フル・クラッシュ・ダンプから得られる `vmcore.n` ファイルと同じ大きさのディスク・スペースが必要になります。

このような状況になるのは、元の `vmcore.n` ファイルに UNIX File System (UFS) ファイルのホールがあるためです (UFS ファイルには、データ・ブ

ロックに関連づけられていない、ホールと呼ばれる領域が含まれていることがあります)。 `gunzip` または `uncompress` コマンドのようなプロセスがファイルのホールから読み取りを行うと、ファイル・システムは、値 0 のデータを返します。このため、部分ダンプから省略されていたメモリが、0 だけを含むディスク・ブロックとして、圧縮されていない `vmcore.n` ファイルに追加されます。

圧縮されていないコア・ファイルを部分ダンプのサイズのままにするには、`-c` オプションを指定した `gunzip` または `uncompress` コマンドから、`conv=sparse` オプションを指定した `dd` コマンドに出力をパイプしなければなりません。たとえば、`vmcore.0.z` という名前のファイルを圧縮解除するには、次のコマンドを実行します。

```
# uncompress -c vmcore.0.z | dd of=vmcore.0 conv=sparse
262144+0 records in
262144+0 records out
```

管理ユーティリティ

この付録では、管理ユーティリティおよびコマンドについての説明と定義を行います。

A.1 X11 グラフィカル・ユーザ・インタフェース (CDE アプリケーション・マネージャ)

X11 ベースのグラフィカル・ユーザ・ユーティリティ (GUI) は、CDE アプリケーション・マネージャまたはコマンド行から使用できます。GUI に SysMan ユーティリティと類似のユーティリティが用意されていたり、GUI が SysMan Menu タスクに置き換えられていることもあります。CDE アプリケーションは、第 1 章で説明した方法で起動します。

すべての管理タスクが、SysMan Menu のオプションとして提供されているわけではありません。GUI、SysMan Menu タスク、およびコマンドを組み合わせる必要があります。各リリースでは、SysMan Menu のオプションが追加され、古い管理方法は廃止されます。管理に関するオプションを理解するために、表 A-1 ~ 表 A-6 にユーティリティをリストします。各表では、次の情報を示します。

- 1 番目の欄は、タスクを識別します。このタスクは、構成するサブシステム、またはシステム・コンポーネント (ディスクやファイル・システムなど) に対して実行する管理アプリケーションです。この欄に出てくるアプリケーションの形式には次の 3 つがあります。
 - カーネル・チューナ (dxkerneltuner) などの、X11 準拠グラフィカル・ユーザ・インタフェース (GUI)
 - 多様なユーザ環境で実行可能な SysMan Menu ユーティリティ
 - コマンド行スクリプト
- 『インストレーション・ガイド』の説明にあるように、スクリプトによってはアクセスするために、OSFRETIREDDXXX サブセットをインストールしなければなりません。

- 2 番目の欄は、SysMan Menu のタスクを示します。[dns] のように大カッコで囲まれているテキストはコマンド・オプションで、コマンド・プロンプトからユーティリティを直接起動する場合に、sysman コマンドに指定します。例を次に示します。

```
# sysman dns
# sysman dns_client
```

1 番目のコマンドは、すべての DNS タスクのサブメニューを呼び出します。2 番目のコマンドは、ローカル・システムを DNS クライアントとして構成するために使用できるユーティリティを呼び出します。

- 3 番目の欄は、同じタスクを実行するコマンド(コマンド行オプション)を示します。

多くの SysMan Menu タスクには、同等の CDE GUI はありません。コマンド行では大抵の機能が提供されていますが、制限があります。たとえば、usermod コマンドを使って複数のユーザ・アカウントを選択して変更を適用させることはできませんが、アカウント・マネージャの GUI (dxaccounts) を使うとこのような操作を行うことができます。

以降の表は、CDE アプリケーションの分類と同じになっています。SysMan Menu、SysMan Station、および System Setup を起動するアイコンの他に、アプリケーション・マネージャにはシステム管理ユーティリティが収められた次のフォルダがあります。

| | |
|-----------------|--|
| 「システム設定」 | 表 A-1 に、構成ユーティリティをリストします。これらのユーティリティは、はじめてシステムを構成するときにも、システムの定期保守にも使用されます。 |
| 「日常管理」 | 表 A-2 に、日常管理ユーティリティをリストします。これらのユーティリティは、システムの日常的な管理作業に使用されます。 |
| 「モニタリング/チューニング」 | 表 A-3 に、システム動作のモニタリングおよび性能のチューニング用のユーティリティをリストします。 |

「ソフトウェア管理」

表 A-4 に、インストール・ソフトウェア管理用のユーティリティをリストします。

「ストレージ管理」

表 A-5 に、ファイル・システムとストレージの管理用ユーティリティをリストします。

「ツール」

表 A-6 に、システムの統計情報を提供するユーティリティをリストします。

表 A-1: システム管理のシステム設定アプリケーション

| 構成するサブシステム | SysMan Menu オプション | コマンド行インタフェース |
|------------------------------|--|--|
| ATM (atmsetup(8)) | ATM (Asynchronous Transfer Mode) の設定 [atm] | atmconfig(8) |
| 監査構成 | 監査構成 [auditconfig] | なし |
| CDE セットアップ (dtsetup(8)) | なし | なし |
| DHCP サーバ (xjoin(8), DHCP(7)) | ネットワーク・セットアップ・ウィザード [net_wizard] システムを DHCP サーバとしてセットアップ [join] | なし |
| ディスク (diskconfig(8)) | diskconfig で実行されるファイル・システム・タスクの一部が、[ストレージ] オプション下に存在する。 | disklabel(8), newfs(8) |
| DNS (BIND) (bindconfig(8)) | Domain Name Service (DNS (BIND)) [dns , dns_client , dns_server , dns_deconfigure] | Version 5.1 で廃止 |
| DOP (dop(8)) | DOP (Division of Privilege) の設定 [dopconfig] | dop(8) |
| ダンプ | ダンプ構成 [dumpconfig] | なし |
| IPsec | インターネット・プロトコル・セキュリティの構成 [IPsec] | なし |
| latsetup | LAT (ローカル・エリア・トランスポート) の構成 [lat] | latsetup(8) , lat_man- ual_setup(7) |
| LDAP | LDAP 構成のセットアップ [ldap] | ldapadd(8) , ldapdelete(8) , ldapmodify(8) |
| メール (mailconfig(8)) | メールの設定 [mailsetup] | mailsetup(8) |
| NFS (nfsconfig(8)) | NFS (Network File System) [nfs] | Version 5.1 で廃止 |

表 A-1: システム管理のシステム設定アプリケーション (続き)

| 構成するサブシステム | SysMan Menu オプション | コマンド行インタフェース |
|-----------------------|---|-----------------|
| NIS (nissetup(8)) | NIS (Network Information Service) の設定 (NIS) [nis] | なし |
| NTP | NTP (Network Time Protocol) [ntp , ntp_config , ntp_status , ntp_start , ntpstop] | Version 5.1 で廃止 |
| ネットワーク | ネットワーク・セットアップ・ウィザード [net_wizard] | なし |
| PPP (pppd(8)) | Point-to-Point Protocol (PPP) [ppp] | なし |
| プリント (printconfig(8)) | ライン・プリンタの設定 [lprsetup] | lprsetup(8) |
| セキュリティ | セキュリティの設定 [secconfig] | なし |
| SLIP | シリアル・ライン・ネットワーク [serial_line] | startslip(8) |

表 A-2: システム管理の日常管理アプリケーション

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|--|---|---|
| アカウント・マネージャ (dxaccounts(8))。ASU (Advanced Server for UNIX) の機能も参照。 | アカウント [accounts , users , groups , nis_users , nis_groups , ldap_users , ldap_groups] | useradd(8) , usermod(8) , userdel(8) , groupadd(8) , groupmod(8) , groupdel(8) |
| アーカイバ (dxarchiver(8)) | なし | tar(1) , pax(1) , cpio(1) |
| 監査マネージャ (dxaudit(8)) | なし | なし |
| ディスプレイ・ウィンドウ (dxdw(8)) | なし | iostat(1) , netstat(1) , vmstat(1) , who(1) |

表 A-2: システム管理の日常管理アプリケーション (続き)

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|------------------------------------|---|---|
| イベント・ビューア | イベントの参照 [event_viewer] | evmget(1), evmshow(1), evmpost(1)。 その他の関連コマンド については, EVM(5) を参照。 |
| ファイル共有 (dxfileshare(8)) | ローカル・ディレクトリ (/etc/exports)の共有 [export] | mount(8), automount(8), exports(4) |
| ACL の取得/設定 (dxsetacl(8)) | なし | なし |
| ホスト・マネージャ (dxhosts(8)) | なし | なし |
| ライセンス・ マネージャ (dxlicenses(8)) | ライセンス・データの登 録 [lmfsetup] | lmf(8), lmfsetup(8) |
| メール・ユーザ管理 (mailusradm(8)) | なし | なし |
| パワー管理 (dxdpower(8)) | なし | sysconfig(8) |
| SysMan シャット ダウン | システムのシャットダ ウン [shutdown] | shutdown(8) |
| システム情報 (dxsysinfo(8)) | なし | du(1), df(1), swapon(8) |

表 A-3: システム管理のモニタリング/チューニング・アプリケーション

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|--------------------|--|----------------|
| クラス・スケジューラ | クラス・スケジュー リング [class_sched, class_setup, class_start, class_stop] | class_admin(8) |
| HP Insight Manager | HP Insight Manager の 設定 [imconfig] | なし |

表 A-3: システム管理のモニタリング/チューニング・アプリケーション (続き)

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|-------------------------------|-------------------------------|--|
| 構成レポート | 構成レポートの作成 [config_report] | sys_check(8) |
| エスカレーション・レポート | エスカレーション・レポートの作成 [escalation] | sys_check(8) |
| カーネル・チューナ (dxkernel-tuner(8)) | なし | sysconfig(8) , sysconfigdb(8) |
| プロセス・チューナ (dxproctuner(8)) | なし | nice(1) , renice(8) , ps(1) , kill(1) |

表 A-4: システム管理のソフトウェア管理アプリケーション

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|--------------------------|---|--------------|
| ソフトウェア・インストール | インストレーション [install, setldload, setldlist, setldd] | setld(8) |
| アップデート・インストレーション・クリーンアップ | OS アップデート後のク リナップ (updadmin) [updadmin] | updadmin(8) |

表 A-5: システム管理のストレージ管理アプリケーション

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|-----------------------------------|--|-----------------------------------|
| Advanced File System (dtadvfs(8)) | Advanced File System (AdvFS) ユーティリティ [advfs] | advfs(4) を参照 |
| ブート・テープ | ブータブル・テープの作成 [boot_tape] | btcreate(8) , btextract(8) |
| ファイル・システム管理 | ファイルシステム管理ユーティリティ [filesystems] | mount(8) , newfs(8) , fstab(4) |

表 A-5: システム管理のストレージ管理アプリケーション (続き)

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|---|---|----------------|
| Logical Storage Manager (lsmsa(8) , dxmlsm(8X)) | Logical Storage Manager (LSM) の管理 [lsm, volsetup] | volsetup(8) |
| Prestoserve I/O Accelerator (dxpresto(8X)) | Prestoserve ソフトウェアの設定 [presto] | prestosetup(8) |

表 A-6: システム管理ツール

| CDE 管理タスク | SysMan Menu オプション | コマンド行インタフェース |
|----------------------|---------------------------|--------------|
| I/O 統計 (dxdw(8)) | 入出力 (I/O) 統計の参照 [iostat] | iostat(1) |
| ネットワーク統計 (dxdw(8)) | なし | netstat(1) |
| システム・メッセージ (dxdw(8)) | イベント・ビューアを参照 | syslogd(8) |
| 仮想メモリ統計 (dxdw(8)) | 仮想メモリ (VM) 統計の参照 [vmstat] | vmstat(1) |

A.2 SysMan Menu タスクと関連ユーティリティ

A.1 節の表は、各種の SysMan Menu タスクと同等の機能を持つ GUI およびコマンド行ユーティリティを示しています。以下の SysMan Menu ユーティリティが使用できます。[accounts] のようなアクセラレータ・キーワードは、sysman コマンドを使用してコマンド行からユーティリティを起動する場合に使用します。以下に示すのは、SysMan Menu ユーティリティのカテゴリの初期状態のリストです。それぞれのカテゴリを独立した項で説明します。

- アカウント [accounts]
- ハードウェア [hardware]
- メール [mail]
- モニタリング/チューニング [monitoring]
- ネットワーク [network]

- プリント [printers]
- セキュリティ [security]
- ソフトウェア [software]
- ストレージ [storage]
- サポートとサービス [support]
- 一般的なタスク [general_tasks]

A.2.1 アカウント

アカウント [accounts] を使用すると、ユーザ・アカウントの保守を行い、システム・リソースを管理することができます。ユーザ・アカウントの管理については第 7 章を参照してください。「アカウント」では、次の作業を行うことができます。

ローカル・ユーザの管理 [users]

ユーザ・アカウント・データを記録する `/etc/passwd` ファイルを管理します。

ローカル・グループの管理 [groups]

ユーザ・リソース・アクセス・データを記録する `/etc/group` ファイルを管理します。

NIS ユーザの管理 [nis_users]

NIS ユーザ・アカウントを管理します。

NIS グループの管理 [nis_groups]

NIS ユーザ・グループを管理します。

LDAP ユーザの管理 [ldap_users]

LDAP ユーザ・アカウントを管理します。

LDAP グループの管理 [ldap_groups]

LDAP ユーザ・グループを管理します。

A.2.2 ハードウェア

ハードウェア [hardware] を使用すると、システム・ハードウェアと周辺デバイスに関する情報が表示できます。ハードウェアの管理については、『ハードウェア管理ガイド』を参照してください。「ハードウェア」では、次の作業を行うことができます。

ハードウェア階層の参照 [hw_hierarchy]

すべてのシステム・コンポーネントを階層構造で表示します。たとえば、CPU と、バスに接続されているすべてのデバイスが表示されます。詳細は、hwmgr(8) を参照してください。

クラスタの参照 [hw_cluhierarchy]

クラスタの全メンバを階層構造で表示します。

デバイス情報の参照 [hw_devices]

システムに接続されているすべてのデバイス（たとえばディスク）のリストを表示します。詳細は、hwmgr(8) を参照してください。

CPU (Central Processing Unit) 情報の参照 [hw_cpus]

システム上のプロセッサのタイプと、オンラインになっている時間などのシステム状態を表示します。

CPU の管理 [hw_manage_cpus]

マルチプロセッサ・システムの CPU を管理します。

Online Addition/Replacement (OLAR) ポリシ情報 [hw_olar_policy_info]

コンポーネントの削除および追加のポリシ情報を管理します。詳細は、olar_config(4) を参照してください。

A.2.3 メール

メール [mail] を使用すると、電子メールの構成と、メール・アカウントの管理を行うことができます。Mail では、次の作業を行うことができます。

メールの設定 [mailsetup]

システム上に電子メール・サービスを構成できます。

メール・アカウントの管理 [mailusradm]

システム・アカウント所有者の email をセットアップすることができます。

A.2.4 モニタリング/チューニング

モニタリング/チューニング [monitoring] を使用すると、システム・イベント報告とチューニング・ユーティリティを構成して使用できます。次のユーティリティが使用できます。

イベントの参照 [event_viewer]

イベント・マネージャ・ビューアを起動できます。詳細については、EVM(5) および第 13 章を参照してください。

HP Insight Manager の設定 [imconfig]

HP Insight Manager を構成できます。詳細については、第 1 章を参照してください。

クラス・スケジューリング [class_sched]

プロセス・グループに CPU 時間リソースを割り当てることができます。詳細については、class_admin(8) および第 3 章を参照してください。次のユーティリティを使用できます。

クラス・スケジューラの設定 [class_setup]

プロセスが使用するシステム・リソース (CPU 時間など) を管理するスケジューリング・データベースを作成することができます。現在のスケジュールを設定することもできます。

クラス・スケジューラの起動/再起動 [class_start]

現在選択されているスケジュールを実行するためのスケジューリング・デーモンを起動します。

クラス・スケジューラの停止 [class_stop]

スケジューリング・デーモンを停止し、リソースの共用をオフにします。

仮想メモリ (VM) 統計の参照 [vmstat]

仮想メモリの統計情報をモニタリングできます。詳細については、`vmstat(1)` を参照してください。

入出力 (I/O) 統計の参照 [iostat]

I/O (入力/出力) の統計情報をモニタリングできます。詳細については、`iostat(1)` を参照してください。

UPTIME 統計の参照 [uptime]

システムの使用時間をモニタリングし、システムが動作している時間と、最後にブートしてからの平均作業負荷を確認できます。詳細については、`uptime(1)` を参照してください。

A.2.5 ネットワーク

ネットワーク [network] を使用すると、ネットワーク・リソースの設定と管理を行うことができます。次のユーティリティを使用できます。

ネットワーク・セットアップ・ウィザード [net_wizard]

ネットワーク環境を手順に従って設定できるようにするユーティリティです。概要については、第 1 章を参照してください。また、ネットワーク構成オプションの詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。

基本ネットワーク・サービス [networkbasic]

最も一般的に使用される個々のネットワーク機能を構成するためのタスクの集まりです。次のユーティリティを使用できます。

ATM (Asynchronous Transfer Mode) の設定 [atm]

ATM サービスをセットアップし、ATM アダプタを構成します。詳細は、『ネットワーク管理ガイド：接続編』を参照してください。

ネットワーク・カードの設定 [interface]

TCP/IP アドレスやネットワーク・マスクなどの情報を指定して、ネットワーク・デバイスを構成することができます。

スタティック・ルートの設定 (/etc/routes) [route]

静的経路を使用するためのネットワーク設定と、ルータ・ノードの定義を行うことができます。静的経路は、ローカル・ネットワークおよびリモート・ネットワークとの最も一般的な通信形態です。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。

ルーティング・サービスの設定 (gated, routed, IP ルータ) [routing]

特定のルーティング方式を使用するように、ネットワークを構成できます。ゲートウェイ・ルーティング・デーモン (gated)、ルーティング・デーモン (routed)、またはインターネット・プロトコル (IP) ルータの中から選択できます。詳細は、『ネットワーク管理ガイド：サービス編』を参照してください。

ホスト・ファイルの設定 (/etc/hosts) [host]

/etc/hosts ファイルにリモート・ホスト・システムを追加できます。これにより、ローカル・システムはそのホストを認識でき、ネットワーク通信が可能になります。

等価ホスト・ファイルの設定 (/etc/hosts.equiv) [hosteq]

リモート・ホスト・システムとユーザを、/etc/hosts.equiv ファイルに追加できます。これにより、リモート・ホストのユーザがローカル・システムのリソースを使用できるようになります。等価ホストに関連するセキュリティの問題については、『セキュリティ管理ガイド』を参照してください。

リモート who サービスの設定 (rwhod) [rwhod]

ローカル・ネットワークのユーザに関する情報を取得できます。

ネットワーク・ファイルの設定 (/etc/networks) [networks]

ネットワークの情報をローカル・システムに知らせることができます。

追加ネットワーク・サービス [networkadditional]

他のネットワーク機能を構成するために使用できるユーティリティの集まりです。次のユーティリティを使用できます。

Domain Name Service (DNS (BIND)) [dns]

ローカル・システムのドメイン名サービスを構成します。次のユーティリティを使用できます。

- DNS サーバとしてシステムを設定 [dns_server]
- DNS クライアントとしてシステムを設定 [dns_client]
- このシステムの DNS の設定解除 [dns_deconfigure]

シリアル・ライン・ネットワーク [serial_line]

次のシリアル・ライン・ネットワーク・オプションを構成できます。

Point-to-Point Protocol (PPP) [ppp]

次の作業を含む、PPP の構成を行うことができます。

- オプション・ファイルの作成 [ppp_options]
- pap-secrets ファイルの修正 [pap]
- chap-secrets ファイルの修正 [chap]

詳細は、『ネットワーク管理ガイド：接続編』を参照してください。

UNIX-to-UNIX コピー (uucp) 接続の設定 [uucp]

モデム、TCP/IP、またはケーブル接続上の UUCP を構成できます。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。

NTP (Network Time Protocol) [ntp]

サーバと時刻値を比較することによってシステムの内部クロックを自動調整するか、クライアント・システムの時刻サーバとして動作するようにします。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。次のユーティリティを使用できます。

- NTP クライアントとしてシステムを設定 [ntp_config]
- NTP デーモンのステータス参照 [ntp_status]
- NTP デーモンの起動/再起動 [ntp_start]
- NTP デーモンの停止 [ntp_stop]

IPsec (Internet Protocol Security) の設定 [ipsec]

システム上に IPsec (Internet Protocol Security) を構成し管理を行うことができます。

NFS (Network File System) [nfs]

ネットワーク・ファイル・システムを構成し、ホスト間でファイル・システムを共用できるようにします。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。次のユーティリティを使用できます。

- NFS 設定状態の参照 [nfs_config_status]
- NFS クライアントとしてシステムを設定 [nfs_client]
- NFS クライアントとしてのシステム設定を解除 [nfs_deconfig_client]
- NFS サーバとしてシステムを設定 [nfs_server]
- NFS サーバとしてのシステム設定を解除 [nfs_deconfig_server]
- NFS デーモン・ステータスの参照 [nfs_daemon_status]
- NFS デーモンの起動/再起動 [nfs_start]
- NFS デーモンの停止 [nfs_stop]

NIS (Network Information Service) の設定 [nis]

異なるホストへログインするなど、ネットワーク・システムのリソースをユーザが使用できるようにします。ユーザ名とパスワードが、ホスト間で分散されます。詳細については、『ネットワーク管理ガイド：サービス編』を参照してください。

LAT (Local Area Transport) の設定 (LAT) [lat]

LAT の設定を行うことができます。詳細については、『ネットワーク管理ガイド：接続編』を参照してください。

DHCP サーバ (joind) としてのシステムの設定 [joind]

DHCP をセットアップできます。詳細は、『ネットワーク管理ガイド：接続編』を参照してください。

ネットワーク・デーモンのステータスの参照 [dmnstatus]

gated や rwhod などの各種ネットワーク・デーモンの状態を調べることができます。

ネットワーク・サービスの起動/再起動 [inet_start]

停止している gated や rwhod などのネットワーク・デーモンを起動または再起動できます。

ネットワーク・サービスの停止 [inet_stop]

すべてのネットワーク・サービスを停止できます。

A.2.6 プリント

プリンタ [printers] を使用すると、システムのプリント機能を構成できます。このタスクでは、次のユーティリティが起動されます。

ライン・プリンタの設定 [lprsetup]

ローカルおよびリモート(ネットワーク)のプリント・デバイスを使用可能デバイスのリストに追加し、これらのリソースをユーザが使用できるようにします。

A.2.7 セキュリティ

セキュリティ [security] を使用すると、システムのセキュリティ、システム監査、および管理ユーティリティにアクセスできる特権ユーザを管理できます。詳細については、『セキュリティ管理ガイド』を参照してください。次のユーティリティを使用できます。

DOP (Division of Privilege) の設定 [dopconfig]

特権を必要とするプログラム (SysMan Menu タスクのなど) に、ユーザがフル・アクセスできるようにします。詳細については、dop(8) を参照してください。

DOP アクションの管理 [dopaction]

DOP アクションとそれらに対応する特権を作成、変更、または削除できます。

セキュリティの設定 [secconfig]

基本セキュリティまたはエンハンスド・セキュリティを構成できます。

監査の設定 [auditconfig]

セキュリティ監査の設定と起動を行うことができます。

A.2.8 ソフトウェア

ソフトウェア [software] を使用すると、オペレーティング・システムおよびレイヤード・ソフトウェアのインストールとアップデートを管理できます。次のユーティリティを使用できます。

インストール [install]

オペレーティング・システムとコンポーネントのインストールができます。このタスクは、次のユーティリティを提供しています。

ソフトウェアのインストール [setldload]

RIS サーバまたは配布メディア (CD-ROM) からシステムにソフトウェアを追加できます。

インストールされているソフトウェアの一覧 [setldlist]

システムに現在インストールされているソフトウェアをリストできます。

インストールされているソフトウェアの削除 [setldd]

システムから永久的にソフトウェアを削除できます。

OS アップデート後のクリンナップ [updadmin]

スペースを節約するためにシステムから不要なファイルを削除したり、アップデート・インストール (updateinstall) 後にファイルをテープに保管することができます。

ライセンス・データの登録 [lmfsetup]

ソフトウェアの製品登録キー (PAK: Product Authorization Key) を登録できます。

A.2.9 ストレージ

ストレージ [storage] を使用すると、ファイル・システムとデータ・ストレージを管理できます。次のユーティリティを使用できます。

ファイルシステム管理ユーティリティ [filesystems]

ディスク・ストレージの基本的な管理を行うことができます。詳細については、第 6 章を参照してください。次のユーティリティを使用できます。

一般的なファイルシステム・ユーティリティ [generalfs]

UFS と AdvFS のどちらでも使用できるユーティリティを提供しています。このタスクは、次のユーティリティを提供しています。

- ファイルシステムのアンマウント [dismount]
- 現在マウントされているファイルシステムの表示 [df]
- ファイルシステムのマウント [mount]
- ローカル・ディレクトリの共用 (/etc/exports) [export]
- ネットワーク・ディレクトリのマウント (/etc/fstab) [net_mount]

Advanced File System (AdvFS) ユーティリティ [advfs]

AdvFS ドメインに関する基本的な管理作業を実行できます。詳細については、advfs(4) および『*AdvFS 管理ガイド*』を参照してください。次のユーティリティを使用できます。

- AdvFS ドメインの管理 [domain_manager]
- AdvFS ファイルの管理 [file_manager]
- AdvFS ドメインのデフラグメント [defrag]
- 新しい AdvFS ドメインの作成 [mkfdmn]
- 新しい AdvFS ファイル・セットの作成 [mkfset]
- AdvFS ドメインのファイルの修復 [salvage]
- AdvFS ドメインの修復 [verify]

UNIX File System (UFS) ユーティリティ [ufs]

UFS 上での基本的な管理作業を実行できます。UFS 管理の詳細については、第 6 章を参照してください。次のユーティリティを使用できます。

新しい UFS ファイルシステムの作成 [newfs]

ディスクのパーティションに新規のファイル・システムを作成することができます。

Logical Storage Manager (LSM) の管理 [lsm]

LSM (Logical Storage Manager) ボリュームの基本的な管理を実行できます。詳細については、『*Logical Storage Manager*』を参照してください。次のユーティリティを使用できます。

- Logical Storage Manager (LSM) の初期化 (LSM) [volsetup]
- Logical Storage Manager (LSM) 管理 [lsmmgr]

ブータブル・テープの作成 [boot_tape]

障害回復時に利用できるスタンドアロン・カーネルを、ブート可能テープに作成できます。詳細については、btcreate(8)、btextract(8)、および第 9 章を参照してください。

SAN アプライアンス・ウィザードの識別 [idsanappl]

記憶領域ネットワーク (SAN) 管理アプリケーションを識別し、SysMan Station に追加することができます。

A.2.10 サポートとサービス

サポートとサービス [support] を使用すると、トラブルシューティングおよびエラー回復の一環として、またはテクニカル・サポート担当者へ問題を報告する必要がある場合に、事前に構成された sys_check システム調査作業を実行できます。詳細については、第 3 章と、sys_check(8) を参照してください。次のユーティリティを使用できます。

ダンプの設定 [dumpconfig]

クラッシュ・ダンプのプロパティを構成できます。

ダンプ・スナップショットの作成 [onlinedump]

手動でクラッシュ・ダンプ・ファイルを作成できます。

エスカレーション・レポートの作成 [escalation]

テクニカル・サポート担当者に提出するシステム調査レポートを作成できます。

構成レポートの作成 [config_report]

ベースライン、トラブルシューティング、またはチューニングの目的で、システム調査レポートを作成することができます。

A.2.11 一般的なタスク

一般的なタスク [general_tasks] には、以下のようなさまざまな管理ユーティリティがあります。

Shut down the System [shutdown]

システムを安全にシャットダウンできるようにします。詳細については、`shutdown(8)` および第 2 章を参照してください。

クイック・セットアップ [quicksetup]

基本システム・セットアップ・ウィザードを実行して、代表的な基本システムの構成手順を示します。機能説明については、第 1 章を参照してください。

Prestoserve ソフトウェアの設定 [presto]

Prestoserve を構成することができます。詳細については、『*Guide to Prestoserve*』を参照してください。

X ディスプレイ・マネージャの設定 [xsetup]

省略時のウィンドウ環境として、CDE または XDM を選択できます。

クローニング・セットアップ情報 [cloneinfo]

`sysman -clone` コマンドの使用についての情報を表示します。このコマンドは、システム構成のクローニングを行い他のシステムに適用します。システムのクローニングについては、『インストレーション・ガイド — 上級ユーザ編』を参照してください。

コマンド行インタフェース情報 [sysmancli]

システム・プロンプトまたはシェル・プログラミングから SysMan タスクを実行するコマンド行インタフェース `sysman -cli` の使用についての情報が表示されます。

LDAP 構成のセットアップ [ldap]

LDAP (Lightweight Directory Access Protocol) を構成できます。



A

acctcms コマンド..... 10-32
acctcom コマンド 10-29
acctcon1 コマンド..... 10-22
acctdisk コマンド 10-39
acctdusg コマンド..... 10-38
acctmerg コマンド 10-46
accton コマンド 10-27
acctprc1 コマンド..... 10-33
acctprc2 コマンド..... 10-35
acctwtmp コマンド..... 10-20
ac コマンド 10-21
adduser ユーティリティ 7-22, 7-38
AdvFS ファイル・システム 5-5, 6-2
 diskconfig GUI による作成.... 5-1
 グラフィカル・インタフェー
 ス 6-36
 ドメインのディスマウント... 6-40
 ドメイン名..... 6-40
 マウントされたドメインの表
 示 6-41
 ユーティリティ 6-39, A-18
archiver..... 9-42
 cpio , pax , および tar コマン
 ド 9-42
 グラフィカル・インタフェース 9-6
ASU..... 7-12
 Windows のユーティリティ ... 7-3

ATM

 SysMan Menu..... A-12
 設定 1-23
atmconfig ユーティリティ 1-23
at コマンド 3-5
audit_setup ユーティリティ... 1-22
autofs コマンド..... 6-41
automount コマンド 6-51
 NFS マウントされているファイル・
 システムの表示..... 6-41
autosysconfig コマンド 4-14

B

bcheckrc スクリプト 3-3
BIND/DNS
 (DNS/BIND を参照)
bindconfig ユーティリティ 1-20
binlog 13-2
 Event Manager 13-2
binlog.conf ファイル 12-12, 14-15
binlog_data.c ファイル..... 12-20
binlogd デーモン 12-4,
 12-19, 14-15
 Event Manager 13-15
 UDP ソケットの使用不可 .. 12-19
 停止 12-20
boot_osflags 変数 2-15

bootdef_dev 2-19
Bourne シェル..... 7-8, 7-18
bcreate コマンド 9-6, 9-45
bextract コマンド 9-45
bttape.pid ロック・ファイル .. 9-46
bttape ユーティリティ... 9-6, 9-45

C

CDE..... A-1
 アプリケーション・マネー
 ジャ 1-14
 管理グラフィカル・インタフェー
 ス 1-14
 設定 1-23
 設定ユーティリティ..... 1-16
 電力管理 3-32
 日常管理 A-2
 フロント・パネル 1-15
 モニタリング/チューニング ... A-2
CDSL..... 1-13, 12-6
 確認 6-20
 管理 6-19
 共用ファイル 6-17
 クラスタ 6-15
 構造 6-18
 修正 6-20
 シンボリック・リンク 6-18
 定義 6-16
 ファイル・システム階層 6-15
cdslinvchk コマンド 6-20
cfgmgr.auth ファイル 4-20
cfgmgr デーモン 4-11
chfn コマンド..... 7-8
chmod コマンド 6-12

chsh コマンド 7-8
ckpacct シェル・スクリプト. 10-28
class_admin コマンド A-11
cmx エクササイザ..... 11-28
collect ユーティリティ 11-10,
 11-12
Compaq Analyze 13-3, 13-15
 binlogd 13-15
 エラー・レポート..... 12-1
Configure System Dump
 GUI..... 14-5
consvar コマンド
 コンソール変数の取得 2-4
 コンソール変数の設定 2-7
cpio コマンド 9-6, 9-40
crash コマンド 14-31
Create Dump Snapshot 14-8
cron
 ログ・ファイル管理..... 3-18
 ログ・ファイルのクリーンアップ 3-18
crontabs ディレクトリ 3-16
crontab コマンド..... 3-16
cron デーモン
 自動課金機能の設定..... 10-13
 デーモンへのコマンドの発行 3-16
 ログ・ファイルの保守 12-22
csh
 (C シェル を参照)
C シェル 7-8

D

date コマンド..... 2-27
dd コマンド..... 5-10

| | | | |
|--|--------------------------|--|--------------|
| データ・ディスクのクローニン グ | 5-10 | 設定 | 1-22 |
| DECevent | 13-3, 13-15 | dsfmgr コマンド | |
| エラー・レポート..... | 12-1 | デバイス特殊ファイルのリス ト | 6-44 |
| df コマンド | 6-63 | リカバリでの使用..... | 9-37 |
| DHCP サーバ..... | 1-20 | dump_user_pte_pages システム属 性..... | 14-24 |
| SysMan Menu..... | A-15 | dumpfs コマンド, 空きディスク・ス ペースの確認 | 6-63 |
| diskconfig GUI..... | 5-1, 6-6, 6-43 | dumpsys コマンド... | 14-19, 14-30 |
| ファイル・システムの作成... | 6-43 | dump コマンド | 9-6 |
| diskconfig の GUI | 1-19 | du コマンド..... | 6-64 |
| disklabel コマンド | 5-5, 6-6 | dxaccounts GUI | |
| 省略時のパーティション・テー ブルの作成..... | 9-21 | 省略値 | 7-19 |
| 省略時ラベルの書き込み | 5-8 | dxaccounts の GUI | 1-21 |
| ディスクへのラベル付け | 9-21 | dxarchiver GUI..... | 9-42 |
| パーティション・サイズの表示 | 5-7 | dxfileshare GUI | |
| パーティション・パラメータの編 集 | 5-8 | 共用ファイル・システムのマウント (インポート)..... | 6-51 |
| ラベルをゼロに復元..... | 5-11 | ファイル・システムの共用... | 6-49 |
| リカバリでの使用..... | 9-35 | dxfileshare の GUI | 1-20 |
| disktab ファイル | 6-6 | dxkerneltuner GUI..... | 11-9 |
| diskusg コマンド | 10-37 | dxlicenses の GUI..... | 1-19 |
| Division of Privileges (DOP を参照) | | dxpower の GUI..... | 3-29 |
| dn_setup ユーティリティ..... | 9-35 | dxshutdown の GUI..... | 2-31 |
| DNS/BIND | 1-20 | dxsysinfo GUI..... | 11-10 |
| SysMan Menu..... | A-13 | | |
| クイック・セットアップ | 1-18 | E | |
| doconfig プログラム..... | 4-8, 4-27, 4-32, 4-35 | ECU | 1-43 |
| dodisk シェル・スクリプト.. | 10-36 | edquota エディタ | |
| DOP | | 起動 | 6-58 |
| SysMan Menu..... | A-16 | 猶予期間の設定 | 6-58 |
| | | ed エディタが使用可能..... | 2-15 |

| | | | |
|---------------------------------|--------------|---------------------------------|----------------|
| envconfig ユーティリティ ... | 11-18 | evmd の構成 | 13-17 |
| envmond デーモンの停止と起 動 | 11-22 | evmget..... | 13-8 |
| envmond デーモン .. | 11-18, 11-22 | evmlogger | 13-7 |
| rc.config ファイルの読み込み | 11-22 | evmreload | 13-17 |
| 構成 | 11-22 | evmtemplate ファイル | 13-34 |
| /etc/evmlogger.conf | | evmviewer..... | 13-58 |
| (evmlogger.conf ファイル を参 照) | | evmwatch | 13-4 |
| /etc/exports ファイル | | get サーバ | 13-8 |
| ファイル・システムの共用 ... | 6-48 | SysMan Menu..... | A-11 |
| /etc/fstab ファイル | | アーカイブされた (zip された) ロ グ | 13-33 |
| (fstab ファイル を参照) | | イベント・チャネル..... | 13-2 |
| /etc/gettydefs ファイル | | イベント・テンプレート ... | 13-33 |
| (gettydefs ファイル を参照) | | イベントの自動処理..... | 13-65 |
| /etc/group ファイル | | イベントの抑制 | 13-65 |
| (group ファイル を参照) | | イベントのロギング..... | 13-65 |
| /etc/inittab ファイル | | イベントへの応答..... | 13-64 |
| (inittab ファイル を参照) | | 管理 | 13-15 |
| /etc/networks ファイル | | 管理での使用 | 13-39 |
| (ネットワーク・ファイル を参 照) | | 管理ユーティリティ..... | 13-9 |
| /etc/passwd ファイル | | 機能 | 13-3 |
| (passwd ファイル を参照) | | クライアントのインストール | 13-34 |
| /etc/rc.config ファイル | | 権限ファイル | 13-26 |
| (rc.config ファイル を参照) | | 構成 | 13-17 |
| /etc/routes ファイル | | コマンド行ユーティリティ ... | 13-8 |
| (ルート・ファイル を参照) | | コンポーネント | 13-6 |
| /etc/securettys ファイル | | システム・ファイル..... | 13-11 |
| (securettys ファイル を参照) | | シャットダウン | 2-5 |
| /etc/sysconfigtab ファイル | | セキュリティについての留意事 項 | 13-25 |
| (sysconfigtab ファイル を参照) | | チャネル構成 | 13-19 |
| Event Manager | 13-1 | チャネル・マネージャ | 13-7, 13-32 |
| API | 13-10 | トラブルシューティング ... | 13-68 |
| evmchmgr コマンド..... | 13-32 | ユーザの認証 | 13-26 |
| evmd デーモン | 13-7 | | |

- ユーティリティ 13-5
- リモート・アクセス..... 13-28
- ロガーの構成 13-21
- ログ取得済みのイベントの表
示 13-58
- ログ・ファイルの管理 13-32
- evm.auth** ファイル 13-26
- evmchmgr** コマンド..... 13-7,
13-9, 13-32
- evmd** デーモン..... 13-7, 13-9
- evmget_srv** プロセス..... 13-8
- evmget** コマンド 13-9
- evmlogger.conf** ファイル ... 13-65
- evmlogger** コマンド. 13-10, 13-64
- evmpost** コマンド 13-9
- evmreload** コマンド. 13-10, 13-17
- evmshow** コマンド 13-9
- evmsort** コマンド 13-9
- evmstart** コマンド..... 13-10
- evmstop** コマンド 13-10
- evmtemplate** ファイル..... 13-34
- evmviewer** ユーティリティ . 13-58
- evmwatch** コマンド 13-4, 13-9
- expected_dump_compression** 変
数..... 14-26
- extendfs** コマンド 6-31

F

- fastboot** コマンド 2-40
- fasthalt** コマンド 2-39
- FFM** 6-2
- file-on-file** ファイル・システム
マウント状況の表示..... 6-41

- finger** コマンド..... 7-8
- fsck** コマンド 6-62
 - 重なっているパーティション 6-62
 - 構文 6-62
 - シャットダウン 2-7
 - ファイル・システムの修復... 6-61
 - ファイル・システムのチェッ
ク 6-61
- fstab** ファイル 6-25, 6-27
 - 現在マウントされているファイル・
システムの表示..... 6-42
 - 編集 6-27
- fsx** コマンド 11-26
- fwtmp** コマンド 10-19
 - wtmp ファイルの修正..... 10-19

G

- gated** デーモン 1-20
 - SysMan Menu..... A-13
- gecos** データ..... 7-18
- genvmunix** 2-7, 2-10, 2-21
- get_info** 関数 11-20
- gettydefs** ファイル 3-4
- getty** コマンド 2-24, 2-26, 3-9
 - 設定 1-44
- GID** 7-20
 - passwd ファイルに..... 7-17
 - 最大数 7-15
 - 制限 7-21
- group**..... 7-13
- group_id**
(GID を参照)
- groupadd** コマンド..... 7-12

groupdel コマンド 7-12
groupmod コマンド 7-12
group ファイル... 7-13, 7-20, A-9
grpck コマンド 7-8
GUI A-1
 Advanced File System 6-36
 Configure System Dump 14-5
 Create Dump Snapshot..... 14-8
 diskconfig 5-1, 6-43
 dxaccounts 1-21, 7-9
 dxarchiver..... 9-42
 dxfileshare 6-49, 6-51
 dxkerneltuner..... 11-9
 dxlicenses 1-19
 dxdpower ユーティリティ 3-29
 dxshutdown..... 2-31
 dxsysinfo 11-10
 LSM 6-37
 Mail Configuration ... 1-21, A-10
 mailusradm A-10
 カスタム・セットアップ 1-9
 クイック・セットアップ 1-9
 ブート可能テーブの作成 6-36

H

halt コマンド 2-36, 2-39
HP Insight Manager 11-8
 設定 1-23
 ポート 1-40
hwmgr コマンド
 SysMan Menu..... A-10
 ディスクが使用可能かのチェック 6-43
 デバイスの表示 6-54

I

init.d ディレクトリ 3-3
 構造 3-11
inittab ファイル 2-7, 2-10, 2-23, 3-2, 3-5, 3-10
 boot エントリ 2-24
 initdefault エントリ..... 2-24
 rc スクリプト 2-25
 実行レベルの変更 2-24
 端末回線の使用可能化 2-26
init コマンド..... 2-10, 2-24, 2-31
 inittab ファイルの再読取り .. 2-26
 実行レベルの変更 2-24
 マルチユーザ実行レベル 2-25
Insight Manager
 (HP Insight Manager を参照)
iostat コマンド 11-5, A-12
IP
 (インターネット・プロトコル
 を参照)
i ノード 6-11
 番号 6-12
 ブロック..... 6-11
 未使用 6-11

J

joind..... 1-20

K

killall コマンド 2-31
kopt コマンド 4-32
Korn シェル 7-8
ksh

(Korn シェル を参照)

L

lastcomm コマンド 10-35
lastlogin シェル・スクリプト 10-24
last コマンド 10-24
LAT
 SysMan Menu A-15
 設定 1-21
latsetup ユーティリティ 1-21
LBN 6-5
LDAP 7-4, A-9
 構成 A-21
 ユーザ・アカウントの管理... 7-21
License Manager
 (dxlicenses の GUI を参照)
Lightweight Directory Access Protocol
 (LDAP を参照)
lineuse ファイル 10-22
lmfsetup コマンド A-17
 SysMan Menu A-17
Logical Storage Manager
 (LSM を参照)
lpc コマンドの引数 8-34
lpd デーモン 8-48
lpd フィルタ 8-50
lprsetup ユーティリティ 8-16, 8-26
 メイン・メニュー 8-27
lpctest コマンド 8-59
LSM A-19
 UFS ファイル・システムの作
 成 6-54

ユーティリティ 6-37

M

Mail Configuration GUI 1-21, A-10
mailsetup ユーティリティ A-10
mailusradm GUI A-10
max-per-proc-address-space 3-42
max-per-proc-data-size 3-42
max-per-proc-stack-size 3-42
{memb} パス要素 6-18
memx コマンド 11-26
 スワップ領域の制約 11-27
MFS 6-22
mkdir コマンド 6-43
mkfdmn コマンド 5-5
Monitoring Performance History
 (MPH ユーティリティ を参照)
mount コマンド 6-25, 6-30
 SysMan Menu 6-44
 状態の変更 6-26
 ファイル・システム 6-43
 マウントされているファイル・シス
 テムの表示 6-40
MPH ユーティリティ 11-15
msgbuf.savecore ファイル.. 14-15

N

netstat コマンド 11-6
newfs コマンド 5-5
 SysMan Menu で使用するオブショ
 ン 6-54

ファイル・システムの作成... 6-22

NFS

autofs ユーティリティ 6-41

automount 6-51

SysMan Menu..... A-15

SysMan によるマウント 6-50

クイック・セットアップ 1-18

設定 1-20

ファイル・システムのマウント (イ
ンポート) 6-50

マウントされているファイル・シス
テムの表示 6-41

NIC 1-19

NIS..... A-9

SysMan Menu..... A-15

クイック・セットアップ 1-18

設定 1-20

nissetup ユーティリティ 1-20

NTP

SysMan Menu..... A-14

クイック・セットアップ 1-18

設定 1-21

O

osf_boot コマンド 2-21

P

pac コマンド..... 10-40

PAK

dxlicenses ユーティリティ ... 1-19

クイック・セットアップ 1-18

登録 A-17

param.c ファイル 4-41

partial_dump 変数..... 14-25

passwd コマンド 7-8

passwd ファイル . 7-14, 7-17, A-9

管理 A-9

破損のパッチ 7-8

ファイルのチェック..... 7-8

pax コマンド..... 9-6, 9-40

PC

SysMan の使用 1-40

管理 1-23

グループ..... 7-38

PCF..... 8-56

pcfof プリント・フィルタ..... 8-56

PCL..... 8-56

PC からの管理 1-23

per-proc-address-space 3-42

per-proc-data-size..... 3-42

per-proc-stack-size 属性..... 3-42

POSIX シェル 7-8

PostScript のプリント 8-56

pppd デーモン 1-21

PPP の構成

SysMan Menu..... A-14

PPP の設定 1-21

prctmp シェル・スクリプト . 10-23

prdaily シェル・スクリプト . 10-48

printcap ファイル..... 8-41

プリンタ特性データベース... 8-37

printconfig ユーティリティ ... 8-16

proc サブシステム 3-42

prtacct シェル・スクリプト . 10-47

pwck コマンド..... 7-8

Q

quotacheck コマンド

ファイル・システム・クォータの確
 認 6-60
 ブロック使用量の確認 6-60
quotaoff コマンド
 ファイル・システム・クォータの停
 止 6-59
quotaon コマンド 7-8
 ファイル・システム・クォータの有
 効化 6-59
quota コマンド 7-8
 ブロック使用量の確認 6-60
quot コマンド
 使用中のブロック数の表示... 6-65

R

rc.config.common ファイル . 2-15,
 6-17
rc.config ファイル. 2-15, 3-4, 6-17
 envmond デーモンによる使
 用 11-22
rc0.d ディレクトリ 3-12
rc0 スクリプト 2-7, 3-4
rc1 スクリプト 2-7
rc2.d ディレクトリ 3-13
rc2 スクリプト 3-4
rc3.d ディレクトリ 3-15
rc3 スクリプト 2-7, 3-4
rcmgr コマンド.. 2-16, 3-4, 14-22
rcn.d ディレクトリ 3-3
rc ディレクトリ構造 3-11
rdump コマンド 9-6
reboot コマンド 2-39
 自動 2-4

すぐにリブートする..... 2-39

Remote Installation Service

(RIS を参照)

restore

ファイル・システムの取り出
 し 9-21
restore コマンド..... 9-6
 データの取り出し..... 9-20
 ファイルの対話式取り出し... 9-23
 ファイルの取り出し..... 9-22
 リカバリでの使用..... 9-37
 リモート・ファイルの取り出
 し 9-26

RIS..... 9-31

root ログイン, 端末での許可 .. 3-11

routed デーモン 1-20

rrestore コマンド 9-6

runacct シェル・スクリプト. 10-42

runsyscheck

(sys_check ユーティリティ を
参照)

rwhod デーモン 1-20

SysMan Menu..... A-13

S

SAS

(スタンドアロン・システム を
参照)

SAVECORE_DIR 変数, 設定 14-22

savecore コマンド..... 14-4

クラッシュ・ダンプ・ファイルの作
成 14-13

sa コマンド 10-30

/sbin/kopt コマンド
 (**kopt** コマンド を参照)
securettys ファイル 3-3, 3-11
sendmail ユーティリティ 1-21
Server System MIB 11-21
setld コマンド A-17
sh
 (Bourne シェル を参照)
shmx エクササイズ 11-27
show devices コマンド 2-19
shutacct コマンドの構文 10-15
shutdown コマンド 2-1,
 2-30, 2-36
 halt フラグの使用 2-38
 SysMan Menu A-20
 緊急 2-38n
 コンソール 2-36
 シャットダウンとリブート... 2-37
 シングルユーザ・モードへの変
 更 2-25
 停止フラグの使用 2-36
 電源を切る 2-36
 バックアップのため 9-14
 メッセージ 2-35
 ユーザへの警告 2-34
 リブート・フラグの使用 2-37
sizer プログラム 4-8
skel ファイル 7-14
SLIP
 SysMan Menu A-14
 設定 1-21
SMP 2-26
 cpu-enable-mask 属性 2-20
 CPU の追加 2-27
 CPU を有効にする 2-20
 自動リブート 2-27
 障害が発生したプロセッサのリブ
 ート 2-27
snmp_request コマンド 11-22
SRM コンソール 2-3, 9-29
stty, 設定 1-46
SWCC
 インストレーション
 (StorageWorks コマンド・コ
 ンソール を参照)
sync コマンド 2-31, 6-61
sys_check ユーティリティ... 11-11,
 11-14
 cron 11-15
 エスカレーション・レポート 11-8
 システム構成レポート 11-8
sysconfigdb コマンド
 サブシステム・エントリの削
 除 4-25
 属性値のリスト 4-22
 属性定義の更新 4-23
 属性定義の削除 4-24
 属性定義のマージ 4-22
 属性の追加 4-22
sysconfigtab ファイル . 1-44, 4-20
 DCD タイムアウト値 1-44
 除外メモリの設定 14-28
 スワップ 6-29
 スワップ領域 3-41
 複数のバージョン 4-21
sysconfig コマンド 3-31, 4-12
 dump_sp_threshold 属性を調整する
 ために使用 14-23
 dump_user_pte_pages 属性を設定す
 るために使用 14-24

| | | | |
|-------------------------------|------------|---------------------|----------|
| 構文 | 4-11 | LSM ユーティリティ | 6-39 |
| サブシステムのタイプの判断 | 4-13 | NFS | A-15 |
| 動的サブシステムをアンロー | | NIS | A-15 |
| ド | 4-14 | NIS アカウント | A-9 |
| リモート・サブシステム管理に使 | | NTP | A-14 |
| 用 | 4-19 | PPP の構成 | A-14 |
| syslog | 12-6, 13-2 | rwhod | A-13 |
| Event Manager | 13-2 | shutdown コマンド | A-20 |
| 省略時の syslog.conf ファイル | 12-7 | SLIP | A-14 |
| セキュリティ | 12-15 | Storage オプション | 6-38 |
| syslog.auth ファイル | 12-15 | UFS ファイル・システム ... | A-18 |
| syslog.conf ファイル | 12-6 | UFS ファイル・システムの作 | |
| syslogd デーモン | 12-3 | 成 | 6-53 |
| Event Manager | 13-14 | vmstat コマンド | A-11 |
| 起動 | 12-17 | アカウント・オプション | 7-10 |
| コンソール・メッセージ | 1-46 | アカウント管理 | 7-1, A-9 |
| 停止 | 12-19 | 管理ユーティリティ | 1-14 |
| sysman -cli コマンド | 1-26 | クラス・スケジューラ | A-11 |
| SysMan Menu | 1-1 | シャットダウン | 2-3 |
| AdvFS | A-18 | ストレージ | A-18 |
| AdvFS Utilities | 6-39 | 静的経路 | A-12 |
| ATM | A-12 | セキュリティ | A-16 |
| CDE フロント・パネルからのアク | | ソフトウェア管理 | A-17 |
| セス | 1-15 | 等価ホスト | A-13 |
| DHCP | A-15 | ドキュメント | 1-17 |
| DNS/BIND | A-13 | ネットワーク管理 | A-12 |
| DOP | A-16 | バックアップのためのシャットダウ | |
| fstab ファイル | 6-45 | ン | 9-14 |
| gated | A-13 | ハードウェアの管理 | A-10 |
| iostat コマンド | A-12 | ファイル・システム | A-18 |
| IP ルータ | A-13 | ファイル・システムの管理 ... | 6-35 |
| LAT | A-15 | ファイル・システムのデスマウン | |
| LDAP アカウント | A-9 | ト | 6-39 |

ファイル・システムのマウン
 ト 6-44
 ファイル・システム・ユーティリ
 ティ 6-38
 プリンタ A-16
 ブート可能テープ A-19
 ホスト・ファイル A-13
 マウントされているファイル・シス
 テムの表示 6-40
 メールの構成 A-10
 モニタリングとチューニング A-11
SysMan Station
 AdvFS ファイル・システムの
 ビュー 1-33
 CDE フロント・パネルからのアク
 セス 1-15
 CDE フロント・パネルのアイコ
 ン 1-15
 「Physical Filesystems」
 ビュー 6-39
 status オプション 1-32
 機能 1-32
 シャットダウン 2-3
 デバイスの表示 6-54
 電力管理 3-34
 ドキュメント 1-17
 ハードウェア・ビュー 1-34
 メイン・ウィンドウ 1-31
 メニュー・オプション 1-37
**SysMan クライアント, PC からの起
 動** 1-40
System Server MIB デーモン 11-18

T

tacct ファイル・エラー, 修正 10-47
tar コマンド 9-6, 9-40
TCP/IP 1-20
 プリントの構成 8-17
TERMINFO 環境変数 3-10
terminfo データベース 3-9
tic コマンド 3-10
tip 接続 1-42
tunefs コマンド 6-66
turnacct シェル・スクリプト 10-28

U

uerf コマンド, システム・エクササイ
 ゼとの併用 11-26
UFS ファイル・システム
 diskconfig GUI による作成 5-1
 LSM ボリューム上の 6-54
 クォータ・ファイル 7-16
 検査 6-61
 構造 6-10
 作成 6-21, 6-53
 スパース・ファイル 7-16
 バージョン 6-7
 ファイル・システム・クォータの設
 定 6-55
 ユーティリティ A-18
UFS ファイル・システムの拡張 6-31
UID 7-57
 /usr/include/limits.h 7-17
 最大数 7-15
 省略値 7-19
 制限 7-17
 複製 7-35

UID の複製 7-35
umount コマンド 6-31
 SysMan の代替 6-39
Unix to Unix Copy
 (UUCP の設定 を参照)
UNIX ファイル・システム
 (UFS ファイル・システム を
 参照)
uptime コマンド A-12
user_id
 (UID を参照)
useradd コマンド 7-12
userdel コマンド 7-12
usermod コマンド 7-12
/usr/include/limits.h ファイ
 ル 7-17, 7-21
/usr/skel ファイル
 (skel ファイル を参照)
/usr ファイル・システムのリスト
 ア 9-39
/usr ファイル・システム, リスト
 ア 9-39
utmp ファイル構造体 10-17
uucpsetup ユーティリティ 1-22
UUCP の設定 1-22
uugetty コマンド
 設定 1-44

V

/var/adm/crash ディレクトリ
 (クラッシュ・ディレクトリ を
 参照)
/var/adm/messages

 (メッセージ・ログ を参照)
/var/yp/src/* ファイル 7-15
/var ファイル・システムのリスト
 ア 9-39
/var ファイル・システム, リスト
 ア 9-39
vipw コマンド 7-7
vm-swap-eager 3-41
vmstat コマンド 11-5, A-11
vmunix ファイル 2-7

W

wall コマンド 2-24, 2-31
WBEM 1-39
who コマンド 7-8, 11-6
Windows NT ドメイン・サーバの
 サービス 7-12
Windows NT ドメインのアカウン
 ト 1-21
wtmpfix コマンド 10-18
wtmp ファイル
 fwtmp コマンドによる修正 10-19
wwpsof プリント・フィルタ... 8-56

X

X11 アプリケーション A-1
XDM の設定 1-23

あ

アカウント
 dxaccounts ユーティリティ .. 7-31

| | |
|---------------------------------------|------------------|
| gecos..... | 7-18 |
| GID..... | 7-20 |
| group ファイル | 7-20 |
| LDAP..... | 7-4 |
| NIS | 7-4 |
| NIS ファイル | 7-15 |
| passwd ファイル..... | 7-14, 7-17 |
| UID..... | 7-17 |
| Windows 2000 シングル・サイン・ オン | 7-54 |
| エンハンスド・セキュリティ . | 7-4 |
| クォータ..... | 7-8 |
| グループ・パスワード | 7-20 |
| シェル | 7-18 |
| 自動追加..... | 7-22 |
| 手動追加..... | 7-38 |
| 省略値 | 7-19 |
| 設定 | 1-21 |
| ディレクトリ | 7-18 |
| 電子メール..... | 1-21 |
| ユーザ・アカウント情報の変更 | 7-8 |
| ユーティリティとコマンド.... | 7-7 |
| ロック・ファイル..... | 7-5 |
| アカウント管理 | 7-13 |
| ASU | 7-12 |
| GID..... | 7-20 |
| group ファイル | 7-13 |
| passwd ファイル..... | 7-13 |
| SysMan Menu..... | 7-10, A-9 |
| クイック・スタート..... | 7-9 |
| コマンド行ユーティリティ... | 7-12 |
| システム・セットアップ時.... | 7-9 |
| システム・ファイル..... | 7-13 |
| スケルトン・ファイル | 7-14 |
| セキュリティ | 7-13 |
| ユーザ・アカウント..... | 7-21 |
| アカウント・マネージャ (dxaccounts GUI を参照) | |
| アップグレード | 2-29 |
| アップデート・インストレーショ ン..... | 1-14 |
| アドレス空間..... | 3-42 |
| アプリケーション・マネージャ | 1-11, 1-14, 1-23 |
| システム・モニタリング・ツ ール | 11-9 |
| シャットダウン | 2-3 |
| 「ストレージ管理」フォルダ | 6-35 |
| アーカイブ・サービス..... | 9-1 |

い

| | |
|----------------------------|------------|
| イベント | |
| 起動 | 13-7 |
| 定義されている | 13-1 |
| モデル | 13-4 |
| 抑制 | 13-65 |
| レポート..... | 12-1 |
| イベント管理..... | 13-1, A-11 |
| イベント・チャンネル..... | 13-2 |
| イベント・ロギング | |
| syslogd デーモン..... | 12-3 |
| syslogd デーモンの停止 | 12-19 |
| クラッシュの回復..... | 12-21 |
| 構成ファイル | 12-6 |
| システム・イベント・ロギング機 能 | 12-3 |
| デーモンの起動と停止 | 12-17 |
| 特殊ファイルの作成..... | 12-17 |
| 日次ファイルの作成..... | 12-10 |

| | |
|-------------------------|-------|
| バイナリ・イベント・ログの構成 | 12-20 |
| バイナリ・イベント・ロギング機能 | 12-4 |
| バイナリ構成ファイル | 12-12 |
| ファイルの保守 | 12-22 |
| イベント・ログ | |
| ログ・ファイルの保護 | 12-3 |
| インターネット・プロトコル, ルータ..... | 1-20 |

う

| | |
|-------------------|------|
| ウィンドウ・マネージャ | 1-23 |
|-------------------|------|

え

| | |
|-----------------------|------|
| エスカレーション・レポート ... | 11-8 |
| エラー | |
| イベント..... | 13-1 |
| システム・シャットダウンの警告 | 2-30 |
| エラー・ロギング | 12-1 |
| エラー・ログ | |
| プリンタ..... | 8-49 |
| 延期モードのスワッピング | 3-40 |
| エンハンスド・セキュリティ | |
| ユーザ・アカウントの変更.... | 7-4 |

お

| | |
|-------------------------|------|
| オンライン・ヘルプ | 1-6 |
| オーバコミット・モードのスワッピング..... | 3-40 |

か

| | |
|---------------------|-------|
| 回復 | |
| ディスク..... | 9-20 |
| ディレクトリ | 9-1 |
| データ | 9-20 |
| 課金 | |
| utmp ファイル構造体 | 10-17 |
| オンやオフにする..... | 10-23 |
| ファイル | |
| (課金ファイル を参照) | |
| プロセス..... | 10-25 |
| 課金機能 | 10-2 |
| crontab コマンドの使用.... | 10-14 |
| cron へのコマンドの発行... | 10-14 |
| rc.config ファイル..... | 10-12 |
| エラー・メッセージ..... | 10-45 |
| 開始と停止..... | 10-15 |
| 月次レポート | 10-10 |
| コマンド | |
| (課金コマンド を参照) | |
| サービス料..... | 10-39 |
| 自動 | 10-14 |
| 接続セッション | 10-15 |
| 設定 | 10-11 |
| ディスク使用量 | 10-36 |
| ディスクのサンプル..... | 10-46 |
| 日次レポート | 10-10 |
| プリンタ..... | 10-12 |
| 課金コマンド | |
| ac | 10-21 |
| acctcms..... | 10-32 |
| acctcom..... | 10-29 |
| acctcon1 | 10-22 |

| | | | |
|---------------------|-------|--------------------------|-------|
| acctdisk..... | 10-39 | カスタム・セットアップの GUI | 1-19 |
| acctdusg..... | 10-38 | 仮想メモリ | |
| acctmerg..... | 10-46 | アドレス空間 | 3-42 |
| accton..... | 10-27 | スタック制限 | 3-42 |
| acctprc1 | 10-33 | スワップ領域 | 3-42 |
| acctprc2 | 10-35 | 説明 | 3-34 |
| acctwtmp..... | 10-20 | データ制限..... | 3-42 |
| ckpacct | 10-28 | プロセス毎..... | 3-42 |
| diskusg..... | 10-37 | 各国語サポート (NLS) | |
| dodisk | 10-36 | LOCPATH 変数 | 3-24 |
| fwtmp | 10-19 | NLSPATH 変数 | 3-24 |
| last | 10-24 | メッセージ・カタログ | 3-24 |
| lastcomm..... | 10-35 | ロケール・カテゴリの変更... 3-22 | |
| lastlogin..... | 10-24 | ロケールの設定 | 3-21 |
| prctmp..... | 10-23 | 環境構成ユーティリティ | |
| prdaily..... | 10-48 | (ECU を参照) | |
| prtacct..... | 10-47 | 環境モニタリング | 11-17 |
| runacct | 10-42 | envmond デーモン | 11-22 |
| sa | 10-30 | envmond デーモンの使用 .. 11-18 | |
| shutacct | 10-15 | get_info 関数..... | 11-20 |
| turnacct | 10-28 | HP Insight Manager..... | 11-18 |
| wtmpfix | 10-18 | カーネル・モジュール・コンポーネ | |
| スタートアップ | 10-15 | ントの使用 | 11-18 |
| リスト | 10-3 | 構成要素..... | 11-17 |
| 課金ファイル | | シャットダウン | 2-30 |
| adm..... | 10-13 | 監査の設定 | 1-22 |
| 管理ファイル | 10-6 | 管理ユーティリティ | A-1 |
| 休日 | 10-13 | カーネル | |
| 月次 | 10-10 | インストレーション後の構成 . 4-8 | |
| データベース・ファイル | 10-6 | 環境モニタリングの属性 ... 11-19 | |
| 日次 | 10-7 | 構成 | 4-1 |
| 付随的なファイル..... | 10-6 | 構成ファイル・エントリ | 4-42 |
| プリンタの使用 | 10-40 | 静的構成..... | 4-25 |
| ルート・ファイル..... | 10-13 | 代替カーネルのブート | 2-19 |
| カスタム・セットアップ GUI ... | 1-9 | 代替ブート..... | 2-10 |

| | |
|------------------------|-------|
| ターゲット..... | 4-8 |
| 動的構成..... | 4-11 |
| 汎用..... | 4-8 |
| リモート・デバッグ..... | 1-45 |
| カーネル構成マネージャ..... | 11-19 |
| カーネル・サブシステム..... | 3-4 |
| 構成変数の設定..... | 3-4 |
| タイプの判断..... | 4-13 |
| 動的にアンロード..... | 4-14 |
| 動的ロードが可能..... | 4-9 |
| カーネル・サブシステムの属性..... | 4-15 |
| 値のリスト..... | 4-22 |
| 識別..... | 4-18 |
| 属性値の確認..... | 4-16 |
| リモート管理..... | 4-19 |
| カーネル・チューナ GUI | |
| (dxkerneltuner ユーティリティ | |
| を参照) | |
| カーネル・モジュール | |
| サポートされるパラメータ..... | 11-19 |
| ロードとアンロード..... | 11-19 |

き

| | |
|--------------------------------|-------|
| 共用ディレクトリ, 非共用..... | 6-50 |
| 共用メモリ, shmx によるテスト..... | 11-27 |

く

| | |
|----------------------------|-----------|
| クイック・セットアップ GUI ... | 1-9, 1-18 |
| クォータ | |

| | |
|-----------------------------|-------------|
| (ファイル・システム・クォータを参照) | |
| クォータ制限..... | 6-56 |
| クラス・スケジューラ..... | A-11 |
| クラスタ | |
| CDSL..... | 6-15 |
| 共用ファイル..... | 6-17 |
| メンバ..... | 6-18 |
| クラッシュからの回復..... | 2-13 |
| クラッシュ・ダンプ..... | 12-21 |
| 圧縮..... | 14-7 |
| 圧縮された..... | 14-26 |
| 圧縮率..... | 14-9 |
| アーカイブ..... | 14-32 |
| 概要..... | 14-1 |
| 継続..... | 14-30 |
| サイズの見積り..... | 14-19 |
| 作成..... | 14-10 |
| 省略時の位置の変更.. | 14-9, 14-22 |
| 除外メモリ..... | 14-27 |
| ディスク・スペース..... | 14-20 |
| 内容と方式..... | 14-22 |
| ハングしたシステム上での強制..... | 14-31 |
| ファイル..... | 14-1 |
| 部分とフルの選択..... | 14-25 |
| フルまたは部分..... | 14-20 |
| フルまたは部分の選択 | 14-7, 14-9 |
| ヘッダ..... | 14-17 |
| 有効にする..... | 14-6 |
| ユーザ・ページ・テーブルを部分ダンプに含める..... | 14-24 |
| リモート・ホストへ..... | 14-29 |

領域の割り当て 14-21
 クラッシュ・ディレクトリ, 変更 14-21
 クラッシュの回復 12-21
 グラフィカル・ユーザ・インタフェース
 (GUI を参照)
 グループ A-9
 PC 7-38
 管理 7-38
 グループ・メンバ 7-38
 コマンド 7-12
 削除 7-38
 省略値 7-38
 追加 7-38
 パスワード 7-38
 ファイルのチェック 7-8
 変更 7-38
 グループ識別子
 (GID を参照)
 グループの削除 7-38
 グループの作成 7-38
 グループ・ファイル
 1 行の長さの制限 7-21
 グループの追加 7-20
 グループ名 7-20
 グループ・メンバ 7-38
 クローニング 9-2
 情報 1-10
 ディスク 9-27

け

継続ダンプ 14-30
 権限ファイル

Event Manager 13-26

こ

構成ファイル

NAME.list ファイル 4-39
 param.c ファイル 4-41
 イベント・ロギング 12-6
 エントリ 4-42
 機能拡張 4-39
 キーワード 4-42
 デバイスの追加 4-27

国際化 3-19, 3-24
 プリント 8-56

コマンド実行スクリプト 2-23

コンソール

auto_action リブート 2-4
 boot_osflags 2-15
 boot_osflags オプション 2-18
 show devices コマンド 2-19
 環境変数 2-17
 スタンドアロン・ブート 2-16
 ブート 2-15
 ブート環境変数 2-17
 ブート・コマンド 2-15
 ブート・デバイス 2-19
 変数の設定 2-7
 変数の変更 2-20

コンソール環境変数

設定 1-44
 定義 2-17
 コンソール・プロンプト 2-36
 コンソール変数 2-4
 コンソール・ポート 1-42, 1-45
 設定 1-43

コンソール・メッセージ 1-46

さ

災害からの復旧 9-1

し

時間帯..... 3-24

時刻，設定 2-27

システム

カスタマイズ 3-1

クラッシュ

(システム・クラッシュを参照)

スタートアップ・ファイル.... 3-2

性能 11-12

セットアップ

(システムのセットアップを参照)

ハードウェア障害..... 2-13

メッセージ..... 11-9

モニタリングとテスト 11-1

モニタリング・ユーティリ

ティ 11-4

リブート中に作成されるクラッ

シュ・ダンプ情報 14-10

システム・イベント 12-1, 13-1

システム・エクササイザ 11-24

診断 11-25

システム管理

SysMan Menu の概要 1-2

SysMan Station..... 1-1

システム設定ユーティリティ A-4t

システムのセットアップ 1-18

ストレージ・アプリケーション

ン A-7t

ソフトウェア管理アプリケーション

ン A-7t

チューニング・アプリケーション

ン A-6t

日常管理アプリケーション... A-5t

日常管理ツールのフォルダ.... A-2

ファイル・システム・アプリケー

ション..... A-7t

メソッド..... 1-11

モニタリング・アプリケーション

ン A-6t

リモート..... 1-42

システム管理ツール A-1

SysMan へのアクセス..... 1-15

「システム管理」フォルダ 1-11

システム・クラッシュ.. 2-13, 12-21

回復 2-13

クラッシュ・ダンプを書き込

む 14-16

ロギング..... 14-14

システム・クロック，設定 2-27

システム構成

カーネル・サブシステム 4-1

静的 4-25

動的 4-11

ドキュメント 4-3

ユーティリティ A-2

システム構成レポート..... 11-8

システム・シャットダウン 2-1

fsck コマンド 2-7

| | | | |
|---------------------|-------|-------------------------|------------------|
| SysMan の使用 | 2-3 | カスタム・セットアップ | 1-19 |
| 関連するシステム・ファイル .. | 2-7 | 初期 | 1-7 |
| 関連ユーティリティ..... | 2-7 | ネットワーク・インタフェース・ | |
| 高いしきい値レベル..... | 11-17 | カード..... | 1-19 |
| 方法とオプション | 2-3 | システムのセットアップ .. | 1-7, 1-18 |
| リモート・システム..... | 1-46 | 概要 | 1-18 |
| システム初期化 | 2-24 | システムの停止 .. | 2-31, 2-36, 2-40 |
| システム・セットアップ | | システムのテスト | 11-1 |
| アカウント管理 | 7-9 | システムの電源を切る..... | 2-36 |
| システム・チューニング | 4-9 | システムのモニタリング | |
| システムの開始 | 2-40 | (モニタリング を参照) | |
| システムの構成 | | システム・ファイル | |
| group ファイル | A-9 | Event Manager | 13-11 |
| passwd ファイル..... | A-9 | アーカイブ・サービス | 9-9 |
| アカウント..... | A-9 | 初期化 | 3-2 |
| システムのシャットダウン | | プリンタ..... | 8-8 |
| fastboot..... | 2-40 | システム・メモリ | 11-26 |
| fasthalt コマンド | 2-39 | システム・モニタリング | 11-12 |
| fsck を実行するように警告... | 2-37 | collect ユーティリティ | 11-10 |
| halt フラグ | 2-38 | HP Insight Manager..... | 11-8 |
| エラーまたはイベントによる | 2-30 | iostat コマンド..... | 11-5, 11-9 |
| 環境モニタリング..... | 2-30 | MPH..... | 11-15 |
| 緊急 | 2-38n | netstat コマンド | 11-6, 11-9 |
| グラフィカル・インタフェー | | uptime コマンド | 11-6 |
| ス | 2-31 | vmstat コマンド | 11-5 |
| システムの停止 | 2-36 | who コマンド | 11-6 |
| 自動リブート | 2-37 | コマンドとユーティリティ... | 11-4 |
| 性能の低下..... | 2-30 | メッセージ・ログ..... | 11-9 |
| トラブルシューティングのため | | システム・リファレンス・マニユア | |
| の | 2-29 | ル・コンソール | 2-3 |
| ファイル・システムの破壊... | 2-30 | システム・ログ, 表示..... | 13-58 |
| マルチユーザ・モードから... | 2-30 | 実行コマンド・スクリプト | 2-10 |
| システムのスタートアップ | 2-8 | 実行レベル | 2-23 |
| システムの設定 | | init コマンドを使用 | 2-24 |
| DNS/BIND | 1-18 | wait エントリ | 3-8 |

| | |
|---|-------------|
| 確認 | 2-23 |
| コンソール..... | 3-9 |
| 省略時設定..... | 3-6 |
| 初期化 | 3-8 |
| シングルユーザ | 2-23 |
| プロセス..... | 3-10 |
| 変更 | 2-24 |
| マルチユーザ | 2-23 |
| 省エネ..... | 3-28 |
| 除外メモリ | 14-7, 14-27 |
| 初期化..... | 2-9 |
| シリアル回線インターネット・プロ トコル (SLIP を参照) | |
| シングルユーザ・ブート | 2-9 |
| シングルユーザ・モード , 課金 | 10-23 |
| シンボリック・リンク..... | 6-18 |
| シンメトリック・マルチプロセシン グ..... | 2-20 |

す

| | |
|--|-------|
| スタック制限..... | 3-42 |
| スタンドアロン・システム | 9-45 |
| スタンドアロン・ブート | 2-14 |
| ファイル・システムのマウン ト | 2-15 |
| スタートアップ・シェル・スクリプト の構文 | 10-15 |
| ストレージ | A-18 |
| ストレージ管理のフォルダ | A-2 |
| 「ストレージ管理」フォルダ ファイル・システム・ユーティリ ティ | 6-35 |

| | |
|------------------------------------|-------|
| スパース・ファイル | 7-16 |
| スプール キュー | 8-49 |
| プリンタ..... | 8-48 |
| スワップ・パーティション | 6-29 |
| スワップ領域 クラッシュ・ダンプ・ファイ ル | 14-16 |
| 枯渇 | 3-41 |
| 説明 | 3-35 |
| 追加 | 3-34 |
| 必要量の見積り | 3-40 |
| 不足を修正..... | 3-42 |
| 割り当て..... | 3-38 |
| 割り当て方法の選択..... | 3-41 |
| スーパブロック | 6-11 |

せ

| | |
|------------------------------|-------|
| 西紀 , 設定 | 2-27 |
| 静的経路 SysMan Menu..... | A-12 |
| 静的構成 | 4-25 |
| 静的ルート | 1-19 |
| 性能 , 性能低下によるシャットダウ ン..... | 2-30 |
| 性能のモニタリング | 11-12 |
| 性能モニタリング 現在のシステム状態..... | 11-12 |
| 製品登録キー (PAK を参照) | |
| セキュリティ syslog..... | 12-15 |
| SysMan Menu..... | A-16 |

| | |
|--------------------------|-------|
| UDP ソケットを使用不可にする | 12-19 |
| イベント管理 | 13-25 |
| エンハンスト (C2) セキュリティ | 7-4 |
| 設定 | 1-22 |
| リモート・ホストのメッセージ | 12-3 |
| セクタ | 6-5 |
| 接続セッション | 10-23 |
| 接続タイプ, プリンタ | 8-44 |
| 接続方法, プリンタ | 8-2 |

そ

| | |
|----------------------------|-------|
| 総合課金レコード | 10-35 |
| 即時モードのスワッピング | 3-40 |
| 属性, 構成可能なカーネル・サブシステム | 4-3 |
| ソフトウェア管理 | |
| SysMan Menu | A-17 |
| ソフトウェア・ライセンス | 1-18 |

た

| | |
|--------------------------------------|-------|
| 対話式ブート | 2-21 |
| 端末 | 3-9 |
| 端末回線の保護 | 3-11 |
| ターゲット・カーネル | 4-8 |
| ターミナル通信システム, cmx によるテスト | 11-28 |

ち

| | |
|---------------|------|
| チェックリスト | 1-18 |
|---------------|------|

て

| | |
|--|-------|
| ディスク | |
| クローニング | 5-10 |
| コピー | 5-10 |
| バックアップ | 9-4 |
| パーティション・テーブル .. | 6-24t |
| 複製とコピー | 9-27 |
| モニタリング | 6-62 |
| ラベル | 6-6 |
| ディスク管理 | 5-1 |
| ディスク・クォータ | 6-55 |
| クォータ超過状態からの回復 | 6-56 |
| ディスク・スペース | |
| 空きスペースの確認 | 6-63 |
| 使用中のブロック数の表示 .. | 6-65 |
| 使用量のチェック | 6-64 |
| ディスク設定ユーティリティ (diskconfig の GUI を参照) | |
| ディスクの回復 | 9-20 |
| ディスクのコピー | 9-27 |
| ディスク・パーティション | 6-3 |
| 重なっているパーティション . | 5-9 |
| サイズ | 6-6 |
| 省略時ラベルの書き込み | 5-8 |
| 定義 | 6-6 |
| パーティション・サイズの表示 | 5-7 |
| ディレクトリ | 6-13 |
| 階層 | 6-4 |
| 回復 | 9-1 |
| タイプ | 6-4 |
| バックアップ | 9-1 |
| リンク | 6-12 |
| ディレクトリの回復 | 9-1 |

デバイス , カーネルへのサポートの追加 4-27
デバイス定義キーワード 4-27
デバイス特殊ファイル 6-14
 名前の表示 6-40
 ファイル・システム 6-40
 プリンタ 8-43
デバイスのパス名 , 説明 8-43
電源を切る 2-36
電子メール A-10, A-11
 (Mail Configuration GUI;
 mailusradm GUI も参照)
 クイック・セットアップ 1-18
 設定 1-21
電力管理 3-32
データ制限 3-42
データの回復 9-20
データ・ブロック 6-13
テープ
 バックアップ 9-4
 ブート可能 9-6
テープ・ドライブ , インストールされていない 9-35

と

等価ホスト
 SysMan Menu A-13
動的サブシステム
 カーネル構成 4-12
 状態の判断 4-12
ドキュメント 1-17
UFS ファイル・システム 6-16
アカウント管理 7-6

アーカイブ・サービス 9-8
クラッシュ・ダンプ 14-2
システム構成 4-3
システムのカスタマイズ 3-36
システムのシャットダウンとリブート 2-4
システムのモニタリングとテスト 11-11
プリンタおよびプリント・サービス 8-5
特権
 (DOP を参照)
ドメイン・ネーム・サービス
 (DNS/BIND を参照)
トラブルシューティング
 sys_check の使用 11-14
 イベント管理 (Event
 Manager) 13-68
シャットダウン 2-29
ファイルとファイル・システム 6-66
プリンタ 8-58
ブート操作 2-28

に

日常管理ツールのフォルダ A-2

ね

ネットワーク・インタフェース・カード , 設定 1-19
ネットワーク情報サービス
 (NIS を参照)

ネットワーク・タイム・プロトコル
(NTP を参照)
ネットワーク・デバイス 9-31
ネットワークの設定
 Network Setup ウィザード... 1-19
 クイック・セットアップ 1-18
ネットワーク・ファイル 1-20
ネットワーク・ファイル・システム
(NFS を参照)
年, 設定 2-27
ネーム・サーバ
(DNS/BIND を参照)

は

バイナリ課金レコード..... 10-35
パスワード 7-20
 グループ 7-38
 設定 A-9
バックアップ
 AdvFS 9-2
 archiver 9-42
 cpio, pax, および tar コマン
 ド 9-40
 dxarchiver ユーティリティ... 9-42
 LSM 9-2
 LSM ミラー..... 9-2
 UFS クォータ・ファイル 7-16
 UFS スパース・ファイル 7-16
 アプリケーション 9-3
 クローニング 9-2
 構成のクローニング..... 9-3
 スクリプトの使用..... 9-19
 スケジューリング 9-5
 増分 9-18

ツールおよびユーティリティ 9-40
手順 9-4
バックアップ・データの損失の回
 避 9-15
 フル 9-16
 メディア・チェンジャ 9-9
 リモート..... 9-19
 ログ・ファイルに cron を使用 3-18
バックアップ・デバイスのインストー
 ル..... 9-35
パニック文字列 14-32
汎用カーネル..... 4-8
ハードウェア, カーネルへのサポート
 の追加 4-27
ハードウェアの管理 A-10

ひ

非同期転送モード
(ATM を参照)

ふ

ファイル 6-4
 回復 9-1
 バックアップ 9-1
 保護 (モード)..... 6-12
ファイル共有, 設定 1-20
ファイル・システム 6-1
 AdvFS ドメインの表示..... 6-41
 automount 6-51
 diskconfig による作成..... 6-43
 dxfileshare GUI による共用.. 6-49
 dxfileshare GUI によるマウン
 ト 6-51

| | | | |
|-------------------------------------|-------|-----------------------------|------------------|
| file-on-file , 表示 | 6-41 | チューニング | 6-66 |
| fstab ファイル | 6-26 | ディスク・パーティションの確 認 | 6-43 |
| hwmgr によるディスクのチェッ ク | 6-43 | ディスマウント | 6-39, 6-42 |
| i ノード・ブロック | 6-11 | ディレクトリ階層 | 6-4 |
| newfs コマンドによる作成 ... | 6-22 | ディレクトリの管理 | 6-13 |
| NFS マウントされているファイル・ システムの表示 | 6-41 | デバイス特殊ファイル名 | 6-40 |
| SysMan Menu ユーティリ ティ | 6-38 | データ・ブロック | 6-13 |
| SysMan による NFS のマウン ト | 6-50 | トラブルシューティング | 6-66 |
| SysMan による管理 | 6-35 | 非共用 | 6-50 |
| SysMan による作成 | 6-53 | ファイル・システム満杯のメッセー ジ | 6-31 |
| SysMan によるマウント状況の表 示 | 6-40 | ファイルの管理 | 6-13 |
| アンマウント | 6-31 | ファイル破壊のためシャットダウ ン | 2-30 |
| エクササイズ | 11-26 | ブート・ブロック | 6-11 |
| エクスポート (共用) | 6-48 | 保護 (モード) | 6-12 |
| グループ用のクォータ | 6-55 | マウント | 6-25, 6-30, 6-44 |
| 現在のマウント状況リストの再表 示 | 6-42 | マウント・ポイントの確認 ... | 6-43 |
| 現在のマウント・ポイントの表 示 | 6-41 | モニタリング | 6-62 |
| 構造 | 6-4 | ユーザ・アカウント用のクォー タ | 6-55 |
| サポートされた | 6-2 | 容量の増加 | 6-31 |
| サポートされているブロック・サイ ズ | 6-25 | リンク数 | 6-12 |
| 使用量の制限 | 6-55 | ファイル・システム・クォータ .. | 6-55 |
| スタンドアロンのマウント ... | 2-15 | edquota エディタの起動 | 6-58 |
| スーパブロック | 6-11 | 確認 | 6-60 |
| 対話形式で修復 | 6-62 | 自動設定 | 6-60 |
| チェック | 6-61 | 停止 | 6-59 |
| | | 有効化 | 6-59 |
| | | 猶予期間の設定 | 6-58 |
| | | ファイル・システムのアンマウン ト | 6-31 |

| | | | |
|---------------------------|-------|---------------------|-------|
| ファイル・システムのインポート..... | 6-50 | エラー・ログ・ファイル | 8-49 |
| ファイル・システムのエクスポート..... | 6-48 | 課金機能..... | 10-40 |
| ファイル・システムの共用 | 6-48 | 削除 | 8-32 |
| dxfileshare GUI による | 6-49 | システムの日常保守..... | 8-31 |
| dxfileshare GUI の使用 | 6-51 | システム・ファイル..... | 8-8 |
| 共用の変更..... | 6-49 | 使用量の報告 | 10-40 |
| マウント (インポート)..... | 6-50 | ジョブとキューの制御 | 8-33 |
| ファイル・システムのディスマウント..... | 6-39 | 直接接続でインストール | 8-21 |
| ファイル・システム・ユーティリティ | | 追加 | 8-31 |
| CDE ストレージ管理..... | 6-35 | デバイス特殊ファイル名 | 8-43 |
| SysMan Menu..... | A-18 | トラブルシューティング | 8-58 |
| ファイル・タイプ | 6-13 | フラグ・ビット | 8-52 |
| シンボリック・リンク・ファイル | 6-13 | ユーティリティ | 8-10 |
| デバイス..... | 6-13 | リモート・インストール | 8-24 |
| ドメイン・ソケット..... | 6-13 | リモート・サーバ..... | 8-55 |
| 名前付きパイプ | 6-13 | プリンタ・サービス, 課金 | 8-35 |
| ファンの故障..... | 11-17 | プリンタ制御ファイル..... | 8-56 |
| ファームウェア | | プリント・サービス | 8-1 |
| 情報の入手先 | 9-29 | pac コマンド | 10-40 |
| 変数の設定..... | 2-7 | 課金機能..... | 10-40 |
| 復元 | | 設定 | 1-22 |
| 手順 | 9-1 | プリント・フィルタ | 8-56 |
| 復旧 | | 汎用 | 8-56 |
| 災害 | 9-1 | プロセス | |
| プリンタ | | 仮想メモリ..... | 3-42 |
| ASU | 8-25 | 初期化 | 2-9 |
| /etc/printcap ファイル..... | 8-41 | プロセス毎のメモリ制限 | 3-42 |
| lpc コマンド | 8-33 | ブート..... | 2-8 |
| lprsetup ユーティリティ | 8-26 | flags -i | 2-21 |
| printconfig ユーティリティ .. | 8-16 | genvmunix | 2-10 |
| TCP/IP プリント..... | 8-17 | コンソール..... | 2-15 |
| | | 準備, システム・クラッシュ | |
| | | 後 | 2-13 |
| | | 準備, シングルユーザ・モード | |
| | | へ | 2-12 |

準備, 停止しているシステム 2-12
準備, 電源が切られているシステム 2-11
スタンドアロン 2-14
設定コマンドの変更 2-20
代替カーネル 2-21
対話式 2-21
トラブルシューティング 2-28
ブート可能テープ 9-6, 9-45
SysMan Menu A-19
グラフィカル・ユーザ・インタ
フェース 6-36
テープの必要条件 9-46
ロック・ファイル 9-46
ブート・コマンド, オプション 2-16
ブート・ディスク, 複製 9-38
ブート・デバイス 2-19
ブート・ドライブ, 代替 9-36
ブート・ブロック 6-11
ブート・ログ・メッセージ ... 12-23

へ

ページングの説明 3-35

ほ

ポイント・ツー・ポイント・プロト
コル
(PPP を参照)
保護, ファイル 6-12
ホスト・ファイル
SysMan Menu A-13

ま

マウント・コマンド
スタンドアロン・ブート 2-15
マウント・ポイント 6-40
一時的 6-41
確認 6-43
現在の使用状況の表示 6-41
作成 6-43
マルチバス・フェイルオーバー ... 2-19
マルチユーザ・ブート 2-9

み

ミラーリング 9-2

め

メッセージ・カタログ, NLS... 3-19
メッセージ, システムからの受
信 11-22
メッセージ・ログ 11-9, 12-23
メディア・チェンジャ 9-9
メモリ
エクササイズ 11-26
共用メモリのテスト 11-27
ダンプ 14-27
メモリ・ファイル・システム 6-2

も

モデム 1-42
設定 1-43
トラブルシューティング 1-47

モニタリング..... 11-1, A-11
 sys_check の使用..... 11-14
モニタリング/チューニング・ツールの
 フォルダ..... A-2

ゆ

ユーザ..... 7-21
ユーザ・アカウント A-9
 管理 7-21
ユーザ・アカウントの削除 7-34
ユーザ識別子
 (UID を参照)
ユーザ・ページ・テーブル
 部分クラッシュ・ダンプに含め
 る 14-24
ユーザ名 7-17

ら

ライセンス 1-18
ライン・プリンタ・デーモン ... 8-48
ラベル, ゼロに復元 5-11

り

リブート手順..... 2-37
リモート **who** ユーティリティ . 1-20
リモート・イベントの監視 ... 13-28
リモート・システム 1-42
 管理用の機能 1-45
リモート・システム管理
 (システム管理 を参照)
リモート接続..... 1-42
リモート・プリンタ用の **printcap** シ
 ンボル 8-40

リモート・ホスト
 共用ファイル・システムへのアクセ
 スの拒否 6-50
 共用ファイル・システムへのアクセ
 スを可能とする..... 6-49
 ダンプ 14-29
リンク..... 6-12
 CDSL..... 6-18

る

ルーティング, 設定 1-20
ルート・ディスク
 回復 9-27
 複製 9-38
ルート・ディスクの回復 9-27
ルート・ディスクの複製 9-27
ルート・ファイル 1-19
ルート・ファイル・システム
 検証 2-21
 シングルユーザ・モードからの読み
 取り/書き込みマウント ... 2-10,
 6-62

れ

レコード
 オーバオール接続セッション 10-23
 日次課金情報 10-35
 バイナリ課金情報..... 10-35

ろ

ロギング, クラッシュ・ダンプ 14-14
ログイン
 シェル 7-18

使用不可 2-35
ディレクトリ 7-18
ログ取得済みのイベント, 表示 13-58
ログ・ファイル
cron によるクリーンアップ .. 3-18
cron によるバックアップ..... 3-18
/var/adm/messages..... 11-9
保守 12-22
ロケール 3-19
サポート・ファイル..... 3-19

ロック
アカウント管理 7-5
論理ブロック番号 6-5
ローカル・エリア・トランスポート
(LAT を参照)

わ

ワールドワイド・サポート 3-24



Tru64 UNIX ドキュメントの購入方法

Tru64 UNIX ドキュメントのご購入については、弊社担当営業または日本ヒューレット・パッカートの各営業所/代理店にお問い合わせください。

各ドキュメント・キットの注文番号は以下のとおりです。ドキュメント・キットに含まれるマニュアルの内容については『ドキュメント概要』を参照してください。

| キット名 | 注文番号 |
|---|-------------|
| Tru64 UNIX Documentation CD-ROM | QA-6ADAA-G8 |
| Tru64 UNIX Documentation Kit | QA-6ADAA-GZ |
| End User Documentation Kit | QA-6ADAB-GZ |
| - Startup Documentation Kit | QA-6ADAC-GZ |
| - General User Documentation Kit | QA-6ADAD-GZ |
| - System and Network Management Documentation Kit | QA-6ADAE-GZ |
| Developer's Documentation Kit | QA-6ADAF-GZ |
| Reference Pages Documentation Kit | QA-6ADAG-GZ |
| TruCluster Server Documentation Kit | QA-6BRAA-GZ |
| Tru64 UNIX 日本語ドキュメント・キット | QA-6ADJB-GZ |
| スタートアップ・ドキュメント・キット | QA-6ADJC-GZ |
| 一般ユーザ・ドキュメント・キット | QA-6ADJD-GZ |
| システム/ネットワーク管理ドキュメント・キット | QA-6ADJE-GZ |
| プログラミング・ドキュメント・キット | QA-6ADJF-GZ |
| CDE 翻訳ドキュメント・キット | QA-6ADJG-GZ |
| TruCluster Server 日本語ドキュメント・キット | QA-05SJA-GZ |
| Advanced Server for UNIX 日本語ドキュメント・キット | QA-5U2JA-GZ |



マニュアルに対するご意見

Tru64 UNIX
システム管理ガイド
AA-RK3PE-TE

弊社のマニュアルに関して、ご意見、ご要望、または内容の不明確な部分など、お気づきの点がございましたら、下記にご記入の上、弊社社員にお渡しくださるようお願い申し上げます。

マニュアルの採点：

| | 大変良い | 良い | 普通 | 良くない |
|-------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 正確さ(説明どおりに動作するか) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 情報量(十分か) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 分かり易さ | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| マニュアルの構成 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 図(役立つか) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 例(役立つか) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 索引(項目の検索性) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ページ・レイアウト(情報の検索性) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

内容の不明確な部分がありましたら、以下にご記入ください：

ペ ー ジ

| | |
|--|--|
| | |
| | |
| | |
| | |

その他お気づきの点がございましたら、以下にご記入ください：

| |
|--|
| |
| |
| |
| |

ご使用のソフトウェアのバージョン： _____

貴社名/部課名 _____

御名前 _____

記入日 _____

(注) 当用紙を受け取った弊社社員は、すみやかに下記にお送りください。

ビジネスクリティカルシステム統括本部 **BCS** 技術本部 **Alpha** ソフトウェア技術部