

Tru64 UNIX

ハードウェア管理ガイド

Part Number: AA-RTFJA-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、ハードウェア・コンポーネントとストレージ・デバイスの管理を担当する UNIX システム管理者を対象とします。本書では、AlphaServer システムで稼動する HP Tru64 UNIX オペレーティング・システムによって制御されるシステム・ハードウェアを管理するためのタスクについて説明しています。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って、コンポーネントの配置換え、ストレージの命名、ハードウェア環境の構成などの管理タスクを実施します。

© 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: Hardware Management (AA-RSFCA-TE)
© 2002 Hewlett-Packard Company

目次

まえがき

1 ハードウェアの管理

1.1	ハードウェアの理解	1-1
1.2	参照情報	1-7
1.2.1	ドキュメント	1-7
1.2.2	Web リソース	1-9
1.2.3	ソフトウェアとアプリケーション	1-9
1.2.4	関連コマンドとユーティリティ	1-13
1.3	ハードウェア管理のためのシステム・ファイル	1-14
1.4	WWID と共用デバイス	1-16
1.5	デバイスの命名とデバイス特殊ファイル	1-17
1.5.1	関連ドキュメントとコマンド	1-19
1.5.2	デバイス特殊ファイルのディレクトリ	1-19
1.5.2.1	擬似デバイスと非ストレージ・デバイス	1-20
1.5.2.2	旧式のデバイス特殊ファイル名	1-23
1.5.2.3	新しいデバイス特殊ファイル名	1-25
1.5.2.4	デバイス特殊ファイル名の変換	1-26
1.5.3	デバイス特殊ファイルの管理	1-27
1.5.3.1	一般的な操作の実行 (dn_setup)	1-28
1.5.3.2	デバイス・クラスとカテゴリの表示	1-29
1.5.3.3	データベースの検証と修復	1-31
1.5.3.4	デバイス特殊ファイルの削除	1-32
1.5.3.5	デバイス特殊ファイル名の移動と交換	1-33
1.5.3.6	デバイス特殊ファイルの番号の付け替え	1-42

2 SysMan Menu と SysMan Station の使用

2.1	SysMan Menu ハードウェア・タスクの使用	2-2
2.1.1	ハードウェア階層の参照	2-3
2.1.2	クラスタの参照	2-5
2.1.3	デバイス情報の参照	2-6
2.1.4	CPU 情報の参照	2-7
2.2	SysMan Station の使用	2-8

3 hwmgr を用いたハードウェア管理

3.1	ハードウェア管理モデル	3-2
3.2	hwmgr コマンド・オプション	3-3
3.3	hwmgr 環境の構成	3-5
3.4	hwmgr を用いたハードウェア管理	3-6
3.4.1	SCSI ハードウェアの位置の特定	3-7
3.4.2	システム階層の表示	3-7
3.4.3	コンポーネント・カテゴリの表示	3-8
3.4.4	コンポーネント属性の取得	3-9
3.4.5	コンポーネント属性の設定	3-11
3.4.6	クラスタの表示	3-13
3.4.7	デバイスの表示	3-13
3.4.8	トランザクションの表示	3-15
3.4.9	ユーザ定義 SCSI デバイス名の作成	3-16
3.4.10	SCSI デバイスの削除	3-19
3.4.11	RAID アレイのディスクの構成設定	3-21
3.4.12	故障した SCSI ディスクの交換	3-22
3.4.13	故障した SCSI テープ・ドライブ (またはハード・ディスク) の交換	3-23

3.4.14	hwmgr によるクラスタ・メンバのブート・ディスクの交換	3-25
3.4.15	name サブシステムの永続データベースの表示	3-26
3.4.16	name 永続データベースからのコンポーネントの抹消と登録解除	3-27
3.4.17	ハードウェア・データベースの最適化	3-28
3.4.18	コンポーネント名の変更	3-29
3.4.18.1	コンポーネントの識別	3-30
3.4.18.2	コンポーネント名の変更	3-30
3.4.18.3	名前変更の検証	3-31
3.4.19	コンポーネントの再配置 (移動)	3-33
3.4.19.1	コンポーネント名の探索	3-34
3.4.19.2	コンポーネントの再配置	3-35

4 DDR (動的デバイス認識)

4.1	カーネルの再構成	4-1
4.2	動的メソッドの使用	4-2
4.2.1	DDR (動的デバイス認識)	4-2
4.2.1.1	標準に対する準拠	4-3
4.2.1.2	DDR メッセージ	4-3
4.2.2	データベース・エントリの例	4-4
4.2.2.1	DDR エントリの例, StorageTek 9840	4-4
4.2.2.2	DDR エントリ例, EMC Symmetrix	4-6
4.3	DDR データベースの手作業による変更	4-7
4.4	カスタマイズ済みの cam_data.c 情報の変換	4-7
4.5	DDR を使用せずに擬似端末とデバイスを追加する	4-8
4.5.1	擬似端末の追加	4-9
4.5.2	他のデバイスの追加	4-11

5	デバイス・コマンドとデバイス・ユーティリティ	
5.1	デバイス・ユーティリティに関する情報	5-1
5.2	SCSI デバイスおよびデバイス・ドライバのユーティリティ ..	5-4
5.2.1	SCSI 構成ユーティリティ scu の使用	5-4
5.2.2	デバイス・スイッチ・マネージャ devswmgr の使用	5-5
5.3	diskconfig を使ったディスクのパーティショニング	5-6
5.4	手作業によるディスク・パーティションの作成	5-9
5.4.1	パーティションの重なりを検索	5-13
5.5	ディスクのコピー	5-14
5.6	ディスク使用量の監視	5-15
5.6.1	利用可能な空き領域の確認	5-16
5.6.2	ディスク使用量の確認	5-17
5.6.3	ディスク・クォータの確認	5-18
6	プロセッサ固有の情報	
6.1	旧式のプロセッサ	6-1
6.2	AlphaServer TS202c	6-2
6.2.1	オペレーティング・システム機能	6-2
6.2.2	オペレーティング・システム機能の制限事項	6-3
6.2.3	mksas コマンドの使用	6-6
6.2.3.1	構成ファイルの要件	6-6
6.2.3.2	su コマンドの使用に関する制限事項	6-7
6.2.3.3	省略時の構成ファイルに ftp が含まれていない	6-7
6.2.3.4	mksas コマンドが警告メッセージを発する	6-7
6.2.3.5	addlist ファイルのサンプルを提供	6-7
6.2.3.6	処理の終了時に作業ディレクトリを削除する mksas の -t オプション	6-8
6.2.4	除外メモリの割り当て	6-8

6.2.5	AlphaServer TS202c の構成	6-9
6.2.5.1	ネットワーク・ブート可能なカーネルの作成	6-9
6.2.5.2	サーバとクライアントの構成	6-10
6.3	AlphaServer GS140 論理パーティション	6-13
6.3.1	ハードウェアの要件	6-13
6.3.2	論理パーティションのインストレーションと操作の準備 .	6-16
6.3.2.1	一般に使用される用語の定義	6-16
6.3.3	論理パーティションの構成とインストレーション・タスク	6-19
6.3.3.1	システムのハードウェア構成の確認	6-19
6.3.3.2	ファームウェアのリビジョン・レベルの確認	6-22
6.3.3.3	論理パーティションの構成	6-22
6.3.3.4	環境変数の選択と設定	6-24
6.3.3.5	コンソール環境変数の表示	6-26
6.3.3.6	コンソール環境変数の訂正	6-26
6.3.3.7	自動ブート・リセットの無効化	6-26
6.3.3.8	メモリ・インターリーブ・モードの設定	6-27
6.3.3.9	オペレーティング・システム・タイプの UNIX への設 定	6-27
6.3.3.10	auto_action コンソール環境変数の設定	6-27
6.3.4	パーティションの初期化	6-27
6.3.5	インターリーブ・モード・エラーの修正	6-28
6.3.6	オペレーティング・システムのインストール	6-29
6.3.7	パーティショニングされたシステムの管理	6-30
6.3.7.1	運用特性	6-30
6.3.7.1.1	コンソール init コマンド (P##>>>init)	6-30
6.3.7.1.2	オペレーティング・システムのシャットダウンと リブート	6-30
6.3.7.2	中断されたオペレーティング・システムのブートの回 復	6-31

6.3.7.3	プロセッサの停止	6-31
6.3.7.4	電源 OFF/ENABLE スイッチの位置	6-32
6.3.7.5	コンソール環境変数の変更によるパーティションの再 構成	6-32
6.3.7.6	ブート前の他のコンソール環境変数の確認	6-33
6.3.7.7	ブート時の論理パーティショニング情報メッセージ ..	6-34
6.3.8	ハードウェアの管理と保守	6-34
6.3.8.1	技術サポートの利用	6-35
6.3.8.2	ハードウェア管理と保守タスクの実行	6-36
6.3.9	UNIX カーネルの再構築が必要なハードウェア変更	6-37
6.3.9.1	パーティションごとの UNIX カーネルの再構築方法 .	6-38
6.3.10	回復不能なハードウェア・エラーのマシン・チェックへの 対応	6-40
6.3.11	論理パーティショニングのエラー・メッセージ	6-42
6.3.12	コンソール・ファームウェアのエラーまたは情報メッセー ジ	6-43
6.4	AlphaServer 1000 および 1000A の構成情報	6-45
6.4.1	EISA Configuration Utility Version 1.10	6-45
6.4.2	グラフィックス解像度	6-45
6.5	AlphaServer GS シリーズの構成情報	6-46
6.5.1	1 次 CPU での OLAR エラーの可能性	6-46
6.5.2	CPU の電源切断と投入を連続的に繰り返さないこと	6-47
6.5.3	ホット追加に関する制限事項	6-47
6.6	Personal Workstation 433au , 500au , および 600au システ ム	6-48
6.6.1	64ビット PCI オプション・カード	6-48
6.6.2	省略時のキーボード・マップの誤り	6-48

7 個々のデバイス・タイプの管理

7.1	ストレージ・デバイス	7-2
-----	------------------	-----

7.1.1	パラレル・スキャニング	7-3
7.1.2	コンソール・レベルのマルチパス・サポートに関する制限 事項	7-5
7.1.3	ファイバ・チャネルのダンプの構成	7-6
7.1.4	アレイからのディスクの取り外し	7-7
7.1.5	故障した HSZ40 コントローラと HSZ50 コントローラの交 換	7-8
7.1.6	故障した HSZ70 コントローラの交換	7-10
7.1.7	HSZ80 コントローラと HSG80 コントローラの下のディス ク上にある永続的予約のクリア	7-12
7.1.8	CD-R/W の書き込みオプション	7-14
7.1.9	フロッピー・ディスクの構成	7-14
7.2	ホスト・バス・アダプタ (HBA) の管理	7-16
7.2.1	AlphaServer 1000A および 2100A システム上の KZPSA ..	7-16
7.2.2	KZPBA (Qlogic ISP1040B) の構成	7-17
7.3	グラフィックス・アダプタ	7-18
7.3.1	関連ドキュメントとリソース	7-18
7.3.2	グラフィックス・コントローラに関する情報の取得	7-19
7.3.3	3Dlabs OXYGEN VX1 PCI/AGP グラフィックス・コント ローラ	7-20
7.3.4	Radeon 7500 PCI/AGP グラフィックス・コントローラ ...	7-21
7.3.4.1	Radeon 7500 の識別	7-21
7.3.4.2	使用上の制限事項	7-22
7.3.4.3	Radeon 7500 のビデオ・モード	7-23
7.3.4.4	Xserver の構成設定	7-24
7.3.4.5	CDE Xserver の構成設定	7-25

8 旧式のハードウェアの管理

8.1	PCMCIA (PC カード) のサポート	8-1
8.2	CalComp グラフィックス・タブレット	8-2

8.3	CI および HSC ハードウェアのサポート	8-3
8.3.1	ハードウェアの設定, 制限事項, およびリビジョン・レベル	8-3
8.3.2	ソフトウェアのインストールおよび制約	8-4
8.3.3	構成ファイルのエントリ	8-4
8.3.4	HSC 制御装置および HSC ディスクのブート	8-5
8.3.5	複数のホスト間でのディスク/テープ・ユニットの共用 ...	8-5

索引

表

1-1	SAN ソフトウェア	1-12
1-2	旧式のデバイス特殊ファイルに使うテープ・デバイスの接尾語	1-24
1-3	新しいデバイス特殊ファイル名の例	1-25
1-4	デバイス名の変換例	1-27
5-1	本書で説明されているデバイス・ユーティリティ	5-2
5-2	リファレンス・ページで説明されているデバイス・ユーティリティ	5-2
7-1	Radeon 7500 のビデオ・モード (解像度)	7-23

まえがき

本書は『ハードウェア管理ガイド』の第 1 版です。以前のリリースでは、本書の情報は『システム管理ガイド』に含まれていました。

本書では、AlphaServer システムで稼動する HP Tru64 UNIX オペレーティング・システムが制御するハードウェア・コンポーネント (PCI バス・オプション・カードなど) やストレージ・デバイス (ディスクやテープなど) の管理作業について説明します。

本書の対象読者

本書は、次のようなスキルを持つシステム管理者を対象としています。

- UNIX オペレーティング・システムの概要と、特権コマンドとユーティリティに関する知識があること。
- UNIX システム管理操作に慣れていること。
- UNIX システムの可用性を高く保つために必要な手段をすべて知っていること。

本書は、UNIX 管理者の教育を目的としたマニュアルではありません。

注意

本書は、AlphaServer プロセッサのハードウェア・コンポーネントと、プロセッサにローカルなデバイス (シングルスピンドル・ディスクなど) 上のハードウェア・コンポーネントの管理だけを対象としています。ストレージ・アレイの構成と保守を含む、ストレージ構成の管理についての重要な情報は、お使いのハードウェア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) などのソフトウェア・アプリケーションも使います。

前リリースからの更新

古いバージョンのオペレーティング・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が必要です。この情報は、Tru64 UNIX ドキュメント CD-ROM 上の HTML ファイル (特に、『新機能および変更された機能』) にあります。さらに、次のオンライン・ドキュメントも利用できます。

- 次の Web サイトで、Tru64 UNIX のオンライン・ドキュメントを参照できます。

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

- 次の Web サイトで、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参照できます。

http://www.tru64unix.compaq.com/docs/pub_page/update_list.html

本書の構成

本書は次の章から構成されています。

- | | |
|-------|---|
| 第 1 章 | Tru64 UNIX のハードウェア管理モデルの概要を説明します。 |
| 第 2 章 | SysMan Menu と SysMan Station を使用して、一般的なハードウェア管理作業を実行する方法を説明します。 |
| 第 3 章 | ディスクやテープなどのハードウェアのクラスを hwmgr コマンドを使用して管理する一般的な方法を説明します。 |
| 第 4 章 | 動的デバイス認識 (DDR) を使用してデバイスをシステムに手動で追加する方法を説明します。 |
| 第 5 章 | 特定のタイプのデバイスを管理したり、ディスクのパーティション作成 (フォーマット) などの特定の作業を実行したりするのに使用するハードウェア管理ツールとユーティリティについて説明します。 |
| 第 6 章 | プロセッサ固有のオペレーティング・システム機能に関する情報について説明します。たとえば、プロセッサ・モデルには、ソフト・パーティションをサポートするものもあります。 |

- | | |
|-------|--|
| 第 7 章 | ストレージ・アレイ，ホスト・バス・アダプタ，グラフィックス・コントローラなど，特定のデバイス・クラスや個々のデバイスの管理方法を説明します。 |
| 第 8 章 | Tru64 UNIX によってサポートされているけれども廃止されている旧式のハードウェア・デバイスの管理手順を説明します。 |

関連ドキュメント

以下のドキュメントは，ライセンスされたソフトウェア・キットに付属のドキュメント CD-ROM に収められている，標準の Tru64 UNIX オペレーティング・システムのドキュメントです。これらのドキュメントには重要な関連情報が含まれています。

- 『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，重要な情報について説明しています。
- 『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レーションについて説明しています。ストレージ・デバイスの初期構成，命名，コンポーネントの割り当てなど，重要なハードウェア管理作業について説明します。
- 『*Managing Online Addition and Removal*』 – システム管理アプリケーションを使用したシステム・コンポーネントのオンライン追加と削除 (OLAR) の使用のための管理および設定技術について説明します。システム管理アプリケーションには，SysMan Station，SysMan Menu，hwmgr コマンドなどがあります。コンポーネントの不具合，自動割り当て解除，関連サービス・ツールに関連するトピックが含まれています。
- 『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。AlphaServer システムに搭載されているネットワーク・インタフェース・カード (NIC) などのネットワーク・コンポーネントの設定とトラブルシューティングに関する情報が含まれています。
- 『システム管理ガイド』，『AdvFS 管理ガイド』，『*Logical Storage Manager*』 – AdvFS (Advanced File System) と UNIX ファイル・システム (UFS) の管理に関する情報が含まれています。

本書の表記法

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。



ハードウェアの管理

本章では、ハードウェア管理モデルの概要を説明するとともに、利用可能な参考資料について解説します。以下のトピックについて取り上げます。

- ハードウェア管理の大まかな概念を説明するとともに、それが本書のどこに関連するかを示します (1.1 節)。
- ハードウェアの管理に用いる他のドキュメント資料 (コマンドやユーティリティのリファレンス・ページなど) を示します。また、ハードウェアの管理に関係するユーティリティに関するドキュメントの参照先についても説明します (1.2 節)。
- システム・デバイスに関する情報が格納されているシステム・ファイルとデータベースについて説明します (1.3 節)。
- 最近のほとんどのデバイスで、デバイス固有の名前がファームウェアにエンコードされていることについて説明します。オペレーティングシステムは、このワールドワイド識別子 (WWID) を使用してデバイスを管理します (1.4 節)。
- 特定の種類のデバイスの論理的な表現であるデバイス特殊ファイルについて説明します (1.5 節)。

`hwmgr` コマンドを使うことによって、CPU のホットスワップのようなサービス・タスクを実施できます。この機能の詳細については、`hwmgr_ops(8)` と『*Managing Online Addition and Removal*』を参照してください。

1.1 ハードウェアの理解

ハードウェア・コンポーネントは、CPU、ネットワーク・カード、またはハードディスクのような、システムの個別の部品のことです。システムは、CPU (中央演算処理装置) を頂点とし、ディスク・ドライブやテープ・ドライブなどの周辺機器を最下層に置く階層構造になっています。これを、システム・トポロジと呼ぶこともあります。以下に述べるようなコンポーネン

トは、これがすべてではありませんが、ほとんどのコンピュータ・システムにある一般的なデバイスです。

- CPU には、シングル・プロセッサ・システム、マルチプロセッサ・システム、およびクラスタ・システムがあります。ハードウェアの管理では、システムをホストと呼ぶことがあります。また、システムには、ホスト名とホスト・アドレス (ネットワーク上にある場合) が割り当てられています。ホスト名を指定してコマンドを使うこともよくあります。CPU はシステムのハードウェア階層の最上位にあり、その他のシステム・コンポーネントは CPU の下に位置しています。

CPU に関する一般的な管理作業は、CPU のオンラインへの移行、起動および停止、CPU リソースの共用など、多岐にわたります。追加情報については、『システム管理ガイド』と『*Managing Online Addition and Removal*』を参照してください。

- バス - システムには主要な内部通信バスがいくつかあり、それらのバスを通してシステムのコンポーネント間でデータを転送します。アダプタとコントローラは物理的にバスへ差し込まれます。こうしたデバイスには物理アドレスと論理アドレスの両方があります。

物理バスによっては特定のソフトウェアが必要なこともありますが、そうしたソフトウェアは、通常、UNIX オペレーティング・システムで管理されます。たとえば、PCI バスにサウンド・カードやネットワーク・カードなどのオプション・カードを追加するときには、システムをシャットダウンしてハードウェアを追加し、リブートしなければなりません。このようなコンポーネントは、通常、リブート時に自動的に認識されシステム構成に追加されますが、ファームウェア・ユーティリティを使ってそのデバイスのドライバをインストールしなければならないこともあります。そのようなコンポーネントの追加についての詳細は、そのシステムのマニュアルと、カードに添付されているマニュアルを参照してください。

- コントローラとアダプタ - システムに、SCSI (Small Computer System Interface) コントローラのように、1 つ以上のストレージ・デバイスを制御するコントローラがいくつか接続されていることがあります。また、1 種類のディスクをサポートし、通常は物理ディスクが 1 台だけ接続されたフロッピー・ディスク・インタフェース (fdi) のようなコントローラが接続されていることもあります。さらに、ネットワーク・アダプタ

がバスに接続されることもあります。しかし、それより下の階層には、ネットワーク・ケーブル以外のコンポーネントはありません。

各アダプタはバス上で自分用の物理スロットを 1 つ占有し、管理用の論理アドレスと物理位置の両方が割り当てられます。アダプタを通してその他のコンポーネントがスロットを占有することもあり、その場合、そのコンポーネントにも物理アドレスと論理アドレスの両方が割り当てられます (コントローラとアダプタは、しばしば HBA または ホスト・バス・アダプタと呼ばれます)。

- SCSI ディスク、CD-ROM リーダなどのストレージ・デバイスは、システム階層の最下位の要素です。これらは一般にコントローラまたはアダプタに取り付けられ、管理用の物理位置と論理アドレスの両方が割り当てられます。

ストレージ (およびその他の) デバイスは、コンポーネント間またはクラスタのメンバ間で共用されることがあります。これは、コンポーネントのアクセス方法によっては、1 つのコンポーネントに複数の異なる名前および ID が割り当てられている可能性があるということです。コンポーネントの識別方法と、階層の他の部分からそのコンポーネントがどのように見えるかを理解することは、ハードウェア管理での重要な側面です。管理者はコンポーネントの論理位置および物理位置の両方を知っていなければなりません。

このマニュアルでは、SCSI デバイスについて説明する場合の例として、SCSI ディスクを最も多く使用しています。これは、SCSI デバイスが最も頻繁にハードウェア管理作業の対象となるデバイスであり、システムから見て、シングル・デバイスにも、グループやアレイにもなるからです。例を次に示します。

シングル SCSI ディスク - RZ デバイス

オペレーティング・システムは、SCSI インタフェース仕様に従うストレージ・デバイスをサポートします。しかし、すべての SCSI デバイスがこの標準に厳密に従っているわけではないので、システムはこのようなデバイスを自動的に検出して追加することができない場合もあります。このようなデバイスは、第 4 章で説明するように、`ddr_config` を用いて追加します。

SCSI RAID - HSZ デバイスと HSV デバイス

RAID (Redundant Array of Inexpensive Disks または Redundant Array of Independent Disks) 技術。これは、複数の SCSI ディスクを連結してボックスに収納したデバイスであり、システムには単一デバイスとして見えます。ホット・スワッピング、フェイルオーバー、および冗長性などの機能をサポートしていることがあります。また、ファイバ・チャネル制御装置を使ってシステムに接続されていることがあります。このようなストレージ・アレイは、ストレージ・エリア・ネットワーク (SAN) 内のシステム間で共用できます。たとえば、HSV110-Enterprise Virtual Array HSV Controller はその例です。

SWCC (StorageWorks Console) のようなアプリケーションを使い、ストレージ・アレイとストレージ・エリア・ネットワークを管理します。このような構成では、`hwmgr` コマンドのようなオペレーティング・システムの機能を使って、ストレージ管理作業の一部分だけを実行することができます。ストレージ・アレイの構成と管理方法の完全な説明は、StorageWorks のドキュメントを参考にしてください。

デバイス特性についての詳細は、RAID(7)、SCSI(7)、および `rz(7)` を参照してください。テープ・デバイスについての詳細は、`tz(7)` を参照してください。現在サポートされている RAID および SCSI の標準については、『*Tru64 UNIX 概要*』と Tru64 UNIX 『*Quickspecs*』を参照してください。

ハードウェアを管理するには、すべてのコンポーネントがお互いにどのように関連し合っているか、また、システム・トポロジの中でコンポーネントが論理的および物理的にどのように位置しているか、さらにはシステム・ソフトウェアがコンポーネントをどのように認識し、それらとどのように通信しているか、を理解していなければなりません。システム・コンポーネントの位置関係をより詳しく知るためには、『システム管理ガイド』で SysMan Station の紹介を参照してください。このユーティリティは、システム・コンポーネントの位置関係を表示し、その表示内容を操作できるようにするグラフィカル・ユーザ・インタフェースです。

大部分のハードウェア管理作業は自動化されています。サポートされている SCSI ディスクをシステムに追加接続してリブートすると、ディスクが自動的

に検出され、システム内に構成されます。オペレーティング・システムは、必要なドライバを動的にロードし、デバイス特殊ファイルを作成します。管理者が行う必要のある作業は、必要に応じてディスクのパーティションを作成し、そのパーティションにファイル・システムを作成して(『システム管理ガイド』で説明)、データを格納できるようにすることだけです。ただし、定期的に手動で行わなければならないハードウェア管理作業はあります。たとえば、ディスクがクラッシュした場合には、代替ディスクを同じ論理位置でオンラインにしなければなりません。また、稼働中のシステムにコンポーネントを手動で追加したり、ディスクの入出力を他のディスクにリダイレクトしなければならないこともあります。

それら以外のハードウェア管理作業の多くは、ディスク・パーティションの作り直しやバスへのアダプタの追加など、標準的なシステムの操作と保守です。このような作業については、通常、コンポーネントに付属しているハードウェア・マニュアルで十分に説明されていますが、新しいコンポーネントの物理/論理位置を最適な(あるいは、より適した)位置にするためにシステムをチェックするというような作業が必要になることもあります。

ハードウェア管理の重要な側面に、予防を目的とした保守およびモニタリング作業があります。システム環境を健全に保つには、次のようなオペレーティング・システム機能を使用します。

- Event Manager (EVM) – システム・イベントをフィルタにかけ、選択されたイベントについて知らせる、イベント・ロギング・システム。電子メールやポケットベルで問題を警告する高度な機能もあります。EVMの構成については、『システム管理ガイド』を参照してください。
- SysMan Station – モニタリングのためにシステム(またはクラスタ)のハードウェア全体を表示させたり、コンポーネントの管理作業のためにアプリケーションを起動させたりできる、グラフィカル・ユーザ・インタフェース。これらのアプリケーションは、SysMan Menu から起動することもできます。アプリケーションの例をいくつか、この章で後述します。SysMan タスクの使い方についての詳細は、2.1 節を参照してください。また、『システム管理ガイド』を参照してください。
- sys_check システム調査ツール – システムの現在の構成に関するデータを HTML ドキュメントにして、Web ブラウザで読めるようにするコマンド行ユーティリティ。このデータをシステムの基準として使用したり、チューニングに使用したり、また、すべてのログ・ファイル

をチェックするために使用したりすることができます。Storage 構成セクションには、ストレージ・デバイスとファイル・システムについての情報があります。このユーティリティの実行方法、および、これを定期的に行うように構成する方法については、『システム管理ガイド』の中でシステムのカスタマイズについて書いてある箇所と、`sys_check(8)` を参照してください。

- HP Insight Manager Agents for Tru64 UNIX – 企業全体を扱える Web ベースの管理ツール。ローカル・エリア・ネットワーク内のどこからでも、システムとコンポーネントの状態を見ることができます。このツールから、SysMan Station、SysMan Menu、およびシステム調査ユーティリティ `sys_check` を起動することができます。詳細は、`insight_manager(5)` を参照してください。

このマニュアルの構成は、次のように、管理するハードウェア・コンポーネントやデバイスを反映したものになっています。

- 汎用のハードウェア管理ツール – これらのツールを使って、同じタイプの全コンポーネントに対しても、SCSI テープ・ドライブなど特定の種類のコンポーネントに対しても、また、個別のコンポーネントに対しても操作を行うことができます。場合によっては、このツールをクラスタに属するすべてのシステムに対して実行させることもできます。このようなツールの例としては、SysMan Station があります。このツールを使えば、クラスタの全メンバについて、そのコンポーネントの階層全体を視覚的に表示させることができます。
- ソフトウェア管理 – これには、システムのハードウェア・コンポーネントに対応するソフトウェアの管理が含まれます。主な管理対象はデバイス特殊ファイルです。デバイス特殊ファイルは、ハードウェア・コンポーネントに結び付いており、どのアプリケーションからでもそのコンポーネントのデバイス・ドライバまたは擬似ドライバにアクセスできるようにします。
- 対象ハードウェア別管理ツール – これらのツールを使えば、特定のコンポーネントを対象とした操作や特定のタスクを行うことができます。ツールの例としては、ディスク構成のコマンド行インタフェース `disklabel` や、これと同じような機能を持つグラフィカル・ユーザ・インタフェース Disk Configuration (`diskconfig`) があります。これらのツールを使うことによって、標準的なレイアウトや独自のレイアウトでディスクのパーティションを作成することができます。

- デバイス固有またはシステム固有のタスク – これらの操作は、特定のデバイスやプロセッサ・プラットフォームを対象とします。このようなタスクの例としては、論理パーティションの構成があります。論理パーティションは、特定の種類のプロセッサだけでサポートされています。

観点を変えれば、汎用ツールを使って多数のコンポーネントを対象にタスクを実行できる一方で、ハードウェア別ツールを使って単一のコンポーネントのみを対象にタスクを実行することもできます。特に明記しない限り、ほとんどの操作は 1 つのシステムまたは 1 つのクラスタを対象とします。クラスタ・ハードウェアの管理についての詳細は、TruCluster サーバのドキュメントを参照してください。

1.2 参照情報

以降の各項で、ドキュメント、システム・ファイル、関連のソフトウェア・ツールに関する参照情報を示します。ここで説明するツールの中には、将来のリリースで廃止される予定のものもあります。廃止される予定のオペレーティング・システム機能の一覧を『リリース・ノート』で参照し、できるだけ早く新しいツールへ移行してください。サイト固有のシェル・スクリプトの中でこうした廃止予定のコマンドを起動していないか、チェックしてください。

1.2.1 ドキュメント

以下のドキュメントに、ハードウェア管理の情報が記載されています。

- マニュアル (オンラインまたはハードコピーで提供)
 - 『*Managing Online Addition and Removal*』 – システム管理アプリケーションを使用したシステム・コンポーネントのオンライン追加と削除 (OLAR) で使う管理および構成の方法について説明します。システム管理アプリケーションには、SysMan Station、SysMan Menu、hwmgr コマンドなどがあります。コンポーネントの不具合、自動割り当て解除、関連サービス・ツールなどのトピックが含まれています。
 - 『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』 – ネットワーク・コンポーネントの構成や接続に関する情報を提供します。

- Device Driver Documentation キット - 『*Writing PCI Bus Device Drivers*』や『*Writing Device Drivers: Reference*』のような関連ドキュメントが含まれています。
- 『*Logical Storage Manager*』 - LSM の概念とコマンドに関する情報が含まれています。LSM (Logical Storage Manager) は、物理ディスク・デバイス、論理エンティティ、およびそれらを相互に関連付けるマッピングからなります。
- リファレンス・ページ
 - hwmgr(8) - ハードウェア管理コマンド /sbin/hwmgr の構文と使用法についての概要です。
 - hwmgr_ops(8) - /sbin/hwmgr コマンド用のシステム操作オプションを説明します。これらのオプションを使って、CPU ホット・スワップなどの手順を実行します。
 - hwmgr_show(8) - /sbin/hwmgr コマンド用のハードウェア情報オプションを説明します。これらのオプションを使って、ハードウェア・データベースからの情報を表示します。
 - hwmgr_get(8) - /sbin/hwmgr コマンド用のコンポーネント属性情報オプションを説明します。これらのオプションを使って、コンポーネント属性を取得、構成します。
 - hwmgr_view(8) - /sbin/hwmgr コマンド用の状態情報オプションを説明します。これらのオプションを使って、コンポーネントとシステム状態を表示します。
 - dsfmgr(8) - デバイス特殊ファイル管理コマンドのコマンド構文について説明します。このコマンドを使って、/dev ディレクトリにデバイス特殊ファイルを作成します。1.5 節も参照してください。
 - mknod(8), MAKEDEV(8), scu(8), ddr_config(8), devswmgr(8) - デバイスを管理するときに使う可能性のある、さまざまなコマンドとユーティリティを説明します。

コマンド行とグラフィカル・ユーザ・インタフェースには、詳細なオンライン・ヘルプもあります。

1.2.2 Web リソース

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。ハードウェア固有のアプリケーション・ソフトウェアの使い方に関して情報が必要な場合には、ハードウェアのオーナーズ・マニュアルも参照してください。

次の Web サイトでは、ドライバのアップデート情報のような情報やリソースを提供しています。

- Alpha 用のファームウェア・ダウンロードを始めとするソフトウェアやドライバは、次の Web サイトにあります。

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

- AlphaServers に関する一般的な情報は、次のサイトにあります。

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

- 次の Web サイトは、ファームウェアやドライバを含め、ストレージに関するサポートを提供します。

<http://www.compaq.com/storage/diskarrays-support.html>

このサイトには、次のタイプのストレージ・デバイスに関する情報があります。

- ストレージ・アレイ (ファイバ・チャネルと SCSI ディスク)
- ストレージの相互接続 (アダプタ、ハブ、およびスイッチ)
- ストレージ・アダプタ (KZPSA などのホスト・バス・アダプタ)

テープ・ストレージ (SDLT160/320) に関する情報は、次の URL にあります。

<http://www.compaq.com/storage/diskarrays-support.html>

1.2.3 ソフトウェアとアプリケーション

ローカル・システムの構成に応じて、1 つ以上のアプリケーションを使ってシステム・コンポーネントを管理します。多くの場合、これらのアプリケーションでは、システムに接続されているファイバ・チャネル・スイッチのような他のデバイスとそのローカルなコンポーネントも管理できます。利用できるアプリケーションは、対象となるストレージ・コンポーネントと、アプリケーションがサポートする Tru64 UNIX のバージョンによって異なります。プログラムの中には、ネットワークに接続された PC システム上だけで動作し、Web 接続を使ってストレージにアクセスするものもあります。ま

た、ハードウェアにソフトウェアがバンドルされておらず、プログラムのライセンスを別に購入する必要があるかもしれません。

複雑な管理手順では、多くの場合、Tru64 UNIX のコマンドや `hwmgr` のようなユーティリティとともに、これらのアプリケーションを 1 つ以上使う必要があります。このような複雑な手順の例としては、マルチパス・フェールオーバーのための冗長ファイバ・チャネル・ストレージの構成があります。

システム・コンポーネントは、以下のアプリケーションを使って管理します。

システム・コンソール

システム・コンソールは、AlphaServer の電源を入れたときに最初に現れるコマンド・インタフェースです。電源投入後の初期テストが正しく完了すると、システムによってコンソール・プロンプト (`>>>`) が表示されます。このインタフェースは、SRM (System Reference Manual) コンソールと呼ばれます。これは、システム・ファームウェアと呼ばれることもあります (ただし、ファームウェアには SRM に属さない他のコンポーネントも含まれます)。

コンソールを使って、システム・コンポーネントに関する情報を表示させることができます。たとえば、次のコマンドは、ローカルの SCSI バスに接続されているディスク・ドライブのバス、ターゲット、論理ユニット番号 (lun) アドレスなど、システムのハードウェア・コンポーネントに関する情報を表示します。

```
>>>show device
```

AlphaServer のオーナーズ・マニュアルでは、SRM コンソール・コマンドについて説明しています。AlphaServer システムによっては、SRM コマンドのオプションに若干の違いがあります。システムのハードウェア・マニュアルが手元になれば、次に示す Alpha Systems Technology Web サイトから印刷可能な PDF 版をダウンロードできます。
<http://www.compaq.com/alphaserver/technology/index.html>

使用するプロセッサのコンソールに固有の制限については、Tru64 UNIX の『リリース・ノート』を参照してください。

アレイ・コントローラ・ソフトウェアと階層性記憶域オペレーティング・ファームウェア

以下のソフトウェアは、特定のいくつかのモデルのストレージ・アレイ・コントローラで動作し、管理タスクを実行できます。

- アレイ・コントローラ・ソフトウェア (ACS)

ACS はホストからの I/O リクエストを処理し、リクエストに応えるために必要なデバイス・レベルの操作を実行します。ACS は、クローニングやフォーマットなどのユーティリティを提供します。ACS オペレーティング・ファームウェアは、コントローラと同梱の PCMCIA プログラム・カードに格納されています。

- 階層性記憶域オペレーティング・ファームウェア (HSOF)

HSOF は機能コード、診断機能、ユーティリティ、および、特定の HS シリーズ・アレイ・コントローラの動作試験をする機能で構成されています。HSOF のコマンド行インタフェースを使って、構成やクローニングなどのコントローラ操作を実行できます。HSOF は、コントローラと同梱の PCMCIA プログラム・カードにも格納されています。

ACS または HSOF オペレーティング・ファームウェアにアップデートがあるたびに、コントローラのモデルごとに新しい PCMCIA プログラム・カードが提供されます。リリースごとにアップデート・カードを個別に購入することも、アップデート・サービス契約の一環として自動的に受け取ることもできます。ファームウェアのアップデート情報については、次の Web サイトを参照してください。

<http://www.compaq.com/products/storageworks/softwaredrivers/acs/index.html>

ACS と HSOF のオンライン・ドキュメントは、次の Web サイトで参照できます。

<http://www.compaq.com/products/storageworks/array-and-scsi-controllers/HSxuserdocs.html>

SWCC (StorageWorks Command Console)

SWCC (StorageWorks Command Console) は、EMA12000 と EMA16000 RAID アレイを対象とした、ストレージの構成とモニタリングのための視覚的なツールです。このツールは、ストレージ管理をポイント・アンド・クリック式の単純な操作で行えるようにし、単独の管理コンソールからストレージを視覚的に構成したりモニタリングしたりできるようにします。SWCC 2.3 は、Tru64 UNIX Version 4.0F 以降を対象とするエージェントを提供します。

SWCC の情報については、次の Web サイトを参照してください。

<http://www.compaq.com/products/storageworks/swcc/index.html>

ストレージ・エリア・ネットワーク (SAN)

HP は、数が増えているいくつかのソフトウェア・アプリケーションを通じて SAN の管理をサポートしています。表 1-1 に、本書発行時点で入手可能なアプリケーションのいくつかを示します。いくつかのアプリケーションは、ネットワークに接続された SAN アプライアンスで動作し、別のソフトウェア・コンポーネントに依存することがあります。他の SAN ツールは、特定のバージョンのオペレーティング・システムを対象としたエージェントだけを提供します。その場合には、Tru64 UNIX で動作するエージェント・プログラムにコマンドを発行する、アプライアンス (または PC) 上で動作するインタフェースを通じてハードウェアを管理することになります。

表 1-1: SAN ソフトウェア

名称	説明
SANworks Command Scripter	HSJ80, HSG60, HSG80, HSZ70 および HSZ80 アレイ・コントローラをコマンド制御できるようにします。CLI (Command Line Interpreter) コマンドを含むスクリプト・ファイルの作成、編集、実行ができます。この製品は、ローカルな LAN 接続と、リモート接続のためのブラウザ・インタフェースの両方を提供します。このアプリケーションは、SWCC との組み合わせで動作します。
SANworks Data Replication Manager (DRM)	ハードウェア冗長性と、距離の離れた複数のサイトの間でデータを複製することによるデータ・レプリケーション機能を提供することによって、耐災害性を実現します。距離の離れた DRM サイトの間は、光ファイバ・ケーブルまたは ATM (非同期転送モード) で結ばれます。DRM は、ファイバ・チャネル・ギガビット・スイッチを使ってサイト間でデータを転送します。
SANworks Element Manager for StorageWorks HSG	HSG コントローラでストレージを一元管理するための、構成とモニタリングを行う Web ベースの視覚的なツールです。
SANworks Enterprise Volume Manager	この Web ベースのアプリケーション・ソフトウェアを使って、コントローラでのクローン操作とスナップショット操作を管理できます。
SANworks Network View	SAN のトポロジを自動的にマッピングし、アベイラビリティを監視し、ストレージ環境のマップを表示するブラウザ・ベースのアプリケーションです。
SANworks Open SAN Manager	Open SAN 向けに、集中化された、アプライアンス・ベースのモニタリングと管理のためのインタフェースを提供します。このインタフェースを使って、SAN 内の一箇所からストレージの編成、表示、構成、および監視ができます。また、SANworks アプリケーションを起動するための手段と、SAN 内のストレージ・コンポーネントを直接管理するためのリンクも提供します。

SAN ソフトウェアとドライバに関する情報については、次の Web サイトを参照してください。

<http://www.compaq.com/storage/sanworks-support.html>

Media Robot Utility (MRU)

DLT や、4mm (DAT) または TKZ シリーズのローダなどのオートローダを制御するためのアプリケーションです。

1.2.4 関連コマンドとユーティリティ

デバイスの管理に使用できるコマンドは、次のとおりです。

- デバイスが正しく動作しているかどうかをテストするために使用します。diskx(8)、tapex(8)、cmx(8)、fsx(8)、および memx(8) を参照してください。
- scu コマンド。SCSI 周辺機器および CAM I/O サブシステムの保守と問題の診断を可能にします。scu(8) と、コマンドのオンライン・ヘルプを参照してください。
- sysconfig コマンド。カーネル・サブシステムの構成を照会または変更する場合に使用します。このコマンドを用いれば、稼働しているカーネルへのサブシステムの追加、カーネルに現在あるサブシステムの再構成、カーネル内のサブシステムについての情報要求 (照会)、そして、カーネルからのサブシステムの構成除外および削除ができます。sysconfig コマンドを用いてコンポーネントの属性値を設定できます。sysconfig コマンドの使い方については、『システム管理ガイド』を参照してください。同書には、Kernel Tuner (dxkerneltuner) についての説明もあります。Kernel Tuner は、属性値を変更するためのグラフィカル・ユーザ・インタフェースです。
- CDE のアプリケーション・マネージャ - 「SysMan アプリケーション」ポップアップ・メニューと「システム管理」フォルダには、いくつかのハードウェア管理ツールがあります。たとえば、次のようなツールです。
 - システム設定 - ATM、ディスク・デバイス、ネットワーク・デバイス、PPP (モデム) デバイス、および LAT デバイスなどのハードウェアを構成するグラフィカル・ユーザ・インタフェース。
 - 日常管理 - 電力管理のためのグラフィカル・ユーザ・インタフェース。特定デバイスの電力属性の設定に使用します。

- SysMan Checklist , SysMan Menu , SysMan Station – どのツールも、システム・デバイスの構成、モニタリング、および保守を行うためのインタフェースです。SysMan Menu と SysMan Station は、パーソナル・コンピュータや X11 ベースの環境など、さまざまなプラットフォームから起動できます。この機能により、リモート・モニタリングとリモート・デバイス管理が実行できます。詳細については、『システム管理ガイド』を参照してください。

1.3 ハードウェア管理のためのシステム・ファイル

次に示すシステム・ファイルには、コンポーネントを構成してカーネルへ組み込む場合にシステムで使用する静的情報または動的情報が含まれています。これらのファイルは、ASCII テキスト・ファイルであっても、編集しないでください。ファイルの中には、『システム管理ガイド』で説明するようなコンテキスト依存シンボリック・リンク (CDSL) もあります。リンクが切れると、そのリンクを検証して再度作成するまでは、クラスタ・システムでそのファイルを使用できなくなります。CDSL の再作成については、`cdslinvcchk(8)` および `mkcdsl(8)` を参照してください。

注意

ハードウェア・データベースの中にはテキスト形式のものもありますが、データベースを編集してはなりません。データベースに誤りがあると、システムがデバイスにアクセスできなくなったり、データが損傷を受けたり、システムが起動できなくなったりすることがあります。デバイスの管理には、適切なコマンドとユーティリティだけを使うようにしてください。

- `/dev` – このディレクトリには、デバイス特殊ファイルが置かれています。詳細については、1.5 節を参照してください。
- `/etc/ddr_dbase` – 動的デバイス認識 (DDR) デバイス情報データベース。このファイルの内容は、コンパイルされてバイナリ・ファイル `/etc/ddr.db` にされます。システムはそのバイナリ・ファイルを用いてデバイス情報を取得します。
- `/etc/dec_devsw_db` – カーネルの `dev` スイッチ・コードを所有者とするバイナリ・データベース。このデータベースには、ドライバのメ

ジャー・デバイス番号とスイッチ・テーブル・エントリの最新情報を記録します。

- `/etc/disktab` – ディスクのジオメトリとパーティション・レイアウト・テーブルを指定するファイル。このファイルは、ディスクのデバイス名と特定のディスク・デバイス属性を知るのに役立ちます。
- `/etc/dvrdevtab` – データベース名と、ドライバ名から特殊ファイル・ハンドラへのマッピングを指定するファイル。
- `/etc/gen_databases` – データベース名をデータベース・ファイルの位置とデータベース・ハンドラに変換するために必要な情報を含むテキスト・ファイル。
- `/etc/dec_hw_db` – ハードウェアの永続的な情報を含むバイナリ・データベース。一般に、バスやコントローラなどのハードウェアが該当します。
- `/etc/dec_hwc_ldb` – クラスタ・メンバから見てローカルにあるハードウェア・コンポーネントの情報を含む、バイナリ・データベース。
- `/etc/dec_hwc_cdb` – クラスタの全メンバによって共用されるハードウェア・コンポーネントの情報を含む、バイナリ・データベース。一意のクラスタ名を持つが、`dev_t` にマッピングされたハードウェア・コンポーネントが、このデータベースに格納されています。
- `/etc/dec_scsi_db` – SCSI/CAM が所有するバイナリ・データベース。SCSI デバイスのワールドワイド識別子 (WWID) が格納されています。CAM は、このデータベースを使用して、システムが認識しているすべての SCSI デバイスの最新情報を把握します。
- `/etc/dec_unid_db` – ハードウェア・コンポーネントに割り当てられているハードウェア ID (HWID) の最高値を格納するバイナリ・データベース。オペレーティング・システムはこのデータベースを使って、新しくインストールするハードウェア・コンポーネントに自動的に割り当てる次の HWID を生成します。システムは HWID を再使用しません。たとえば、ディスクをシステムに追加したときに、124 という HWID が割り当てられたとします。システムからそのディスクを恒久的に取り外した場合でも、HWID 124 は代替のディスクやその他のデバイスに割り当てられることはありません。HWID の番号付けシーケンスをリセットする唯一の方法は、オペレーティング・システムを新たにインストールすることです。

1.4 WWID と共用デバイス

SCSI デバイスの命名は、デバイスの論理識別子 (ID) に基づいて行われます。SCSI デバイスのデバイス特殊ファイル名と物理位置の間に相関関係はありません。UNIX はデバイスからの情報を用いてワールドワイド識別子という ID (通常は WWID と表記) を作成します。

デバイスの WWID が一意に定まり、システムに取り付けられた各 SCSI デバイスの ID を識別できるのが理想です。しかし、一部の古いディスクでは、情報が不足していて、特定のデバイスに一意の WWID を作成できないことがあります (他社製の新しいディスクでもこうしたものがあります)。このようなデバイスに対しても、オペレーティング・システムは WWID を生成しようとします。極端な場合にはデバイスの接続情報 (SCSI バス/ターゲット/LUN) を用いてデバイスの WWID を作成します。

したがって、一意の WWID を持たないデバイスは共用バスで使わないようにしてください。一意の WWID を持たないデバイスを共用バスに接続すると、そのデバイスへの異なるバスごとに異なるデバイス特殊ファイルが作成されます。このような場合、オペレーティング・システムが 2 つの異なるデバイス特殊ファイルを用いて同じデバイスに同時にアクセスして、データが損傷を受ける可能性があります。デバイスがクラスタの中で一意な WWID を持っているかどうかは、次のコマンドで調べます。

```
# /sbin/hwmmgr show components
```

デバイスの `FLAGS` フィールドに `c` フラグがセットされていれば、クラスタ内で一意の WWID があるので、共用バスに置くことができます。このようなデバイスはクラスタ内の共用バスに置くことができるため、「クラスタ内共用可」といいます。

注意

HSZ デバイスにはこの規則はあてはまりません。HSZ デバイスがクラスタ内共用可とマークされていても、HSZ のファームウェアの中には、複数のイニシエータがデバイスを同時にアクセスできないようにしているバージョンがあります。現在の制限については、HSZ デバイスのオーナーズ・マニュアルと Tru64 UNIX 『リリース・ノート』を参照してください。

次の例は、ハードウェア・コンポーネントのうち、クラスタ内で一意の WWID を持つものをすべて表示させたものです。

```
# /sbin/hwmgrr show component -cshared
HWID:HOSTNAME FLAGS SERVICE COMPONENT NAME
-----
35:  pmoba  rcd--  iomap  SCSI-WWID:0410004c:"DEC  RZ28      ..."
36:  pmoba  -cd--  iomap  SCSI-WWID:04100024:"DEC  RZ25F     ..."
42:  pmoba  rcd--  iomap  SCSI-WWID:0410004c:"DEC  RZ26L     ..."
43:  pmoba  rcds-  iomap  SCSI-WWID:0410003a:"DEC  RZ26L     ..."
48:  pmoba  rcd--  iomap  SCSI-WWID:0c000008:0000-00ff-fe00-0000
49:  pmoba  rcd--  iomap  SCSI-WWID:04100020:"DEC  RZ29B     ..."
50:  pmoba  rcd--  iomap  SCSI-WWID:04100026:"DEC  RZ26N     ..."
```

一意の WWID を持っていないデバイスでも、共用バス上で利用可能にしなければならないことがあります。このようなデバイスは共用バスで使用しないほうが良いのですが、共用バス上でこれを構成する方法があります。hwmgrr edit scsi コマンド・オプションを使って一意の WWID を作成する方法については、第 3 章を参照してください。

1.5 デバイスの命名とデバイス特殊ファイル

デバイスは、/dev ディレクトリにあるデバイス特殊ファイルを通して、システムの他の部分から使用できるようになっています。デバイス特殊ファイルにより、アプリケーション (データベース・アプリケーションなど) はデバイス・ドライバを通してデバイスにアクセスできます。個々のデバイス・ドライバは、特定の種類のハードウェア・コンポーネントを 1 つまたは複数制御するカーネル・モジュールです。ここで言うデバイスとは、たとえば、ネットワーク・コントローラ、グラフィックス・コントローラ、ディスク (CD-ROM デバイスを含む) などが該当します。

システムは、ハードウェア・コンポーネントを制御しない擬似デバイス・ドライバへアクセスするのにもデバイス特殊ファイルを使います。たとえば、擬似端末 (pty) の端末ドライバは端末デバイスをシミュレートします。端末ドライバ pty は文字型ドライバであり、一般に、リモート・ログインに使用します。これについては、第 4 章で説明しています。デバイス・ドライバの詳細な情報については、次の URL にあるデバイス・ドライバのドキュメントを参照してください。 <http://www.tru64unix.compaq.com/docs/>

通常、デバイス特殊ファイルの管理はシステムが自動的に行います。たとえば、UNIX オペレーティング・システムの新バージョンをインストールする場合、あるタイミングでシステムがすべてのバスとコントローラを調べ、システム・デバイスをすべて検出します。次に、システムは検出したデバイスを記述するデータベースを構築し、デバイス特殊ファイルを作成して、

ユーザが使用できるようにします。デバイス特殊ファイルを使用する最も一般的な方法は、それをシステムの `/etc/fstab` ファイルに UFS ファイルの位置として指定することです。

デバイス特殊ファイルの操作を手動で実行しなければならないのは、システムに問題がある場合と自動的に処理できないデバイスをサポートしなければならない場合のみです。この節では、バージョン 5.0 以降でデバイスとデバイス特殊ファイルに名前を割り当て構成する方法について説明します。

次の点に注意してください。

- SCSI デバイスに対するデバイス特殊ファイルの名前の形式は、SCSI ディスクが `/dev/disk/disk13a`、そして SCSI テープ・ドライブが `/dev/ntape/tape0_d0` となっています。`/dev/rz10b` という形式の SCSI デバイス特殊ファイル名は、旧式のデバイス特殊ファイルです。この節では、現在のデバイス特殊ファイルと旧式のものを分けて説明します。また、一部のスクリプトおよびコマンドでは、この 2 つを `old` (または `legacy`) および `new` (または `current`) と表記して区別しています。日本語の説明では、それぞれ「古い」(または「旧式の」)と「新しい」(または「現在の」)で表記します。このオペレーティング・システムを初めて使うユーザは、現在の命名モデルをサポートしないサードパーティ製のドライバとデバイスを使うのであれば、旧式のデバイス特殊ファイル名について気にする必要はありません (デバイス特殊ファイルの構造については、この節で後述します)。
- 現在、SCSI ディスクおよびテープ・デバイスに対しては 1 つの共通のデバイス特殊ファイル命名モデルが、また、他のデバイスすべてに対してはこれと異なる命名モデルが適用されています。SCSI ディスクおよびテープ・デバイスに対する命名方法は、将来のリリースでは他のデバイスにも拡張される予定です。そうなれば、旧式のデバイスおよびデバイス名のサポートが、非クラスタ・システムでも継続できるようになります。アプリケーションとコマンドは、すべてのデバイス名をサポートするか、またはサポートされているデバイス名の形式をエラー・メッセージで通知するかのどちらかです。

旧式のデバイス名およびデバイス特殊ファイルはしばらく維持されます。廃止予定については、将来のリリースでお知らせします。

1.5.1 関連ドキュメントとコマンド

次のドキュメントに、デバイス名に関する情報があります。

- マニュアル：
 - コンテキスト依存シンボリック・リンク (CDSL) についての説明があります。デバイス特殊ファイルのあるディレクトリには、CDSL になっているものがあります。次に進む前に、この概念をよく理解してください。詳細については、『システム管理ガイド』を参照してください。
- リファレンス・ページとコマンド
 - デバイス特殊ファイルの管理についての説明は、`dsfmgr(8)` を参照してください。このコマンドは、`MAKEDEV` を置き換えるものです (`MAKEDEV(8)` を参照してください)。
 - ディスク・バック・ラベルの保守についての説明は、`disklabel(8)` を参照してください。
 - Disk Configuration GUI の起動方法についての説明は、`diskconfig(8)` を参照してください。`disklabel` より豊富な機能があり、ディスクのパーティションを作成してそこにファイル・システムを作る操作を 1 回で実行できます。Disk Configuration インタフェースは、CDE の「アプリケーション・マネージャ・システム管理」フォルダから実行することもできます。「ディスク」アイコンは「システム設定」フォルダ内にあります。このインタフェースの使い方は、オンライン・ヘルプに説明があります。

1.5.2 デバイス特殊ファイルのディレクトリ

デバイス特殊ファイルを入れるために、`/devices` ディレクトリがルート・ディレクトリ (`/`) の下に作成されています。このディレクトリにはサブディレクトリがあり、その中にそのデバイス・クラスのデバイス特殊ファイルが置かれます。デバイス・クラスは、同類のデバイスをまとめたタイプ (ディスクや非巻戻し型テープ・デバイスなど) に対応しています。たとえば、`/dev/disk` ディレクトリには、サポートされているすべてのディスクに対するファイルが置かれ、`/dev/ntape` ディレクトリには非巻戻し型テープ・デバイスに対するデバイス特殊ファイルが置かれます。このリリースの場合、作成されているサブディレクトリは一部のクラスのみです。どの操作の

場合でも、`/devices` ディレクトリではなく `/dev` ディレクトリを用いてパスを指定する必要があります。

注意

いくつかのデバイス特殊ファイル・ディレクトリは CDSL です。システムがクラスタの一部のとき、クラスタ単位でデバイスを使用可能にします。『システム管理ガイド』で示すファイル・システム階層 (特に CDSL のインプリメント) について良く理解しておく必要があります。

`/dev` ディレクトリには、`/devices` ディレクトリのサブディレクトリに対応するシンボリック・リンクがあります。たとえば、次のようになっています。

```
lrwxrwxrwx 1 root system 25 Nov 11 13:02 ntape ->
../../../../../../devices/ntape

lrwxrwxrwx 1 root system 25 Nov 11 13:02 rdisk ->
../../../../../../devices/rdisk

lrwxrwxrwx 1 root system 24 Nov 11 13:02 tape ->
../../../../../../devices/tape
```

このような構造になっていることによって、システムがクラスタのメンバになっているときに、特定のデバイスをホスト固有にできます。そしてそれ以外のデバイスを、クラスタの全メンバで共用できます。また、デバイス・ドライバの開発者やコンポーネント・ベンダが、新しいデバイス・クラスを追加することもできます。

1.5.2.1 擬似デバイスと非ストレージ・デバイス

デバイスと、それらに関連付けられているデバイス特殊ファイルを管理するにあたって、システムの `/dev` ディレクトリに何があるかを知っていると役立ちます。`/dev` ディレクトリには、次の出力例に示すように、多数のデバイスがあります。

```
# ls -l
total 47
drwx----- 2 root system 512 Jun 5 15:55 .audit
-rwxr-xr-x 1 bin bin 34777 Jun 19 00:02 MAKEDEV
-rw-r--r-- 1 root system 2238 Jun 28 16:42 MAKEDEV.log
-rwxr-xr-x 1 bin bin 1418 Jun 19 00:02 SYSV_PTY
-rwxr-xr-x 1 bin bin 1418 Aug 1 2001 SYSV_PTY.PreUPD
crw----- 1 root system 47, 0 May 30 17:16 atm_cmm
```

```

crw-rw-rw- 1 root system 87, 0 May 30 17:16 aud97
cr----- 1 root system 17, 0 May 30 16:45 audit
srw-rw---- 1 root system 0 Jul 25 13:40 binlogdmb
crw----- 1 root system 30, 0 May 30 16:45 cam
lrwxrwxrwx 1 root system 27 May 30 17:16 changer -> \
../../../../devices/changer
crw--w--w- 1 root daemon 0, 0 Jul 25 13:44 console
-rw-r--r-- 1 root system 12 Jul 25 13:39 console.gen
lrwxrwxrwx 1 root system 25 May 30 17:16 cport -> \
../../../../devices/cport
lrwxrwxrwx 1 root system 24 May 30 17:16 disk -> \
../../../../devices/disk
lrwxrwxrwx 1 root system 25 May 30 17:16 dmapi -> \
../../../../devices/dmapi
crw----- 1 root system 31, 0 Jul 26 13:43 kbinlog
crw----- 1 root system 3, 1 May 30 16:45 kcon
crw----- 1 root system 82, 0 May 30 17:16 kevm
crw----- 1 root system 82, 2 May 30 17:16 kevm.pterm
crw-rw---- 1 root system 25, 1 May 30 16:45 keyboard0
crw----- 1 root system 3, 0 May 30 16:45 klog
cr--r----- 1 root mem 2, 1 May 30 16:45 kmem
drwxr-xr-x 2 root system 512 May 12 18:14 lat
cr--r----- 1 root mem 45, 0 May 30 16:45 lockdev
srw-rw-rw- 1 root system 0 Jul 25 13:40 log
crw-rw-rw- 1 root system 34, 0 May 30 17:16 lp0
cr--r----- 1 root mem 2, 0 May 30 16:45 mem
crw-rw-rw- 1 root system 88, 0 May 30 17:16 mmsess0
crw-rw---- 1 root system 23, 1 May 30 16:45 mouse0
crw-rw-rw- 1 root system 52, 0 May 30 17:16 msb0
lrwxrwxrwx 1 root system 15 May 30 17:16 none -> \
../devices/none
lrwxrwxrwx 1 root system 25 May 30 17:16 ntape -> \
../../../../devices/ntape
crw-rw-rw- 1 root system 2, 2 Jul 26 13:28 null
cr--r--r-- 1 root system 26, 0 May 30 16:45 pfcntnr
crw-rw-rw- 1 root system 32, 58 May 30 17:16 pipe
crw-rw-rw- 1 root system 46, 0 May 30 16:45 poll
crw----- 1 root system 37, 0 May 30 16:45 prf
srwxrwxrwx 1 root system 0 Jul 25 13:43 printer
crw-rw-rw- 2 root system 32, 7 May 12 18:12 ptm
crw-rw-rw- 1 root system 32, 65 Jun 28 17:14 ptmx
crw-rw-rw- 2 root system 32, 7 May 12 18:12 ptmx_bsd
drwxr-xr-x 2 root system 512 May 12 18:12 pts
crw-rw-rw- 1 root system 7, 31 May 30 16:45 ptyqf
crw-rw-rw- 1 root system 89, 0 May 30 17:16 random
lrwxrwxrwx 1 root system 25 May 30 17:16 rdisk -> \
../../../../devices/rdisk
drwxr-xr-x 2 root system 512 May 30 17:16 sad
crw----- 1 root system 80,1048575 May 30 17:16 scp_scsi
crw-rw-rw- 2 root system 32, 49 May 12 18:12 snmpinfo
drwxr-xr-x 3 root system 512 May 30 17:21 streams
crw-r--r-- 1 root system 15, 0 May 30 16:45 sysdev0
lrwxrwxrwx 1 root system 24 May 30 17:16 tape -> \
../../../../devices/tape
crw-rw-rw- 1 root system 1, 0 May 30 16:45 tty
crw-rw-rw- 1 root system 35, 0 May 30 17:16 tty00
crw-rw-rw- 1 root system 89, 1 Jul 25 13:39 urandom
crw-r----- 1 root mem 2, 5 May 30 16:45 vmzcore
crw-rw---- 1 root system 33, 1 May 30 16:45 ws0
crw-rw-rw- 1 root system 38, 0 May 30 16:45 zero

```

上記のリストでは、重複するデバイスの種類は除いてあります。ディレクトリには、端末 (ttys)、擬似端末 (ptys)、ディスク・ストレージ・デバイス

(dsk) など多数のデバイスが含まれています。デバイスの数と種類は、システムのハードウェアおよびソフトウェアの構成によって異なります。これらのデバイスの大半は、オペレーティング・システムとアプリケーションが必要とするもので、ストレージに関係しません。最も頻繁に管理の対象になるのが、ディスク、テープ、ネットワーク・カードなど、データ入出力を伴うデバイスです。

頻繁に管理するデバイスとコントローラの一覧を表示するには、`hwmgr show devices` コマンドを使います。このコマンドの出力には、`dsfmgr` または `hwmgr` コマンドを使って管理できない入出力デバイスは含まれません。`hwmgr show devices` コマンドの出力には、SCSI カード、ファイバ・チャネル、ストレージ・アレイ・コントローラなどのデバイスに関連する `dev/cport/scp2` のようなストレージ・デバイスも含まれます。`dev/changer/mc*` のエントリは、テープ交換デバイスに対応します。

デバイスの一覧を表示するには、次の `hwmgr` コマンドを使用します。

```
# hwmgr view dev
HWID: Device Name           Mfg      Model      Location
-----
6: /dev/dmapi/dmapi
7: /dev/scp_scsi
8: /dev/kevm
53: /dev/cport/scp5          SWXCR     xcr0
54: /dev/disk/dsk1197c       SWXCR     ctrlr-0-unit-0
69: /dev/disk/floppy53c      3.5in floppy fdi0-unit-0
46: /dev/disk/dsk1155c       DEC       HSG80      bus-4-targ-2-lun-17
79: /dev/disk/dsk6c         DEC       HSZ22      (C) DEC bus-0-targ-0-lun-5
81: /dev/random
82: /dev/urandom
84: /dev/ntape/tape11        COMPAQ    SDT-10000  bus-5-targ-0-lun-0
90: /dev/disk/dsk1194c       COMPAQ    BD009122C6 bus-2-targ-0-lun-0
91: /dev/disk/dsk1195c       COMPAQ    BD009122BA bus-2-targ-1-lun-0
871: /dev/disk/dsk1180c      DEC       HSG80      bus-4-targ-2-lun-13
122: /dev/changer/mc1        TL800     (C) DEC bus-4-targ-0-lun-10
127: /dev/cport/scp2         DATA ROUTER bus-4-targ-0-lun-0
128: /dev/cport/scp3         HSG80CCL bus-4-targ-2-lun-0
129: /dev/cport/scp4         HSV110 (C) COMPAQ bus-4-targ-4-lun-0
926: /dev/ntape/tape0        COMPAQ    DLT8000    bus-4-targ-0-lun-8
927: /dev/ntape/tape1        COMPAQ    DLT8000    bus-4-targ-0-lun-9
942: /dev/disk/dsk1199c      COMPAQ    HSV110 (C) COMPAQ IDENTIFIER=1001
687: /dev/disk/cdrom53c      COMPAQ    CDR-8435   bus-6-targ-0-lun-0
```

上記の出力例からは、類似のディスク・ストレージ・デバイスは除いてあります。この出力には、`/dev/kevm` (カーネル・イベント・マネージャ) のように、一部のカーネル・サブシステム擬似デバイスが含まれていますが、無数にある端末と擬似端末は含まれていません。`/dev/kevm` のようなカーネル・サブシステムについての情報は、システム属性 (`sys_attrs*`) のリファレンス・ページにあります。`sys_attrs(5)` を参照してください。`/dev/random` のような他の擬似デバイスについては、リファレンス・ページに説明があ

ります。特定のデバイスに関するリファレンス・ページを探すには、次の例に示すように `apropos` コマンドを使います。

```
# apropos random
.
.
random, urandom (4)      - Kernel random number source devices
# apropos disk
.
.
disk, dsk, cdrom, rz (7) - SCSI disk interface
```

1.5.2.2 旧式のデバイス特殊ファイル名

旧式のデバイス命名規則に従えば、デバイス特殊ファイルはすべて `/dev` ディレクトリに置くことになります。デバイス特殊ファイル名は、デバイスの種類、物理的な位置、およびその他のデバイス属性を表してます。古い規則を使用するディスクやテープ・ドライブのデバイス特殊ファイル名は、たとえば、SCSI ディスクの `/dev/rz14f` やテープ・ドライブの `/dev/rmt0a` といったファイル名の形式をとります。名前には、次のような情報が含まれます。

パス	接頭語	タイプ	番号 (インスタンス)	接尾語
<code>/dev</code>	<code>r (raw)</code>	<code>rz (ディスク)</code>	0	<code>c (パーティション)</code>
<code>/dev</code>	<code>(巻戻し)</code>	<code>rmt (テープ)</code>	4	<code>a (密度)</code>
<code>/dev</code>	<code>n (非巻き戻し)</code>	<code>rmt (テープ)</code>	12	<code>h (密度)</code>

この情報は、次のように解釈されます。

パスは、デバイス特殊ファイルが置かれるディレクトリです。デバイス特殊ファイルはすべて `/dev` ディレクトリに置かれます。

接頭語は、次のように、同じ物理デバイスでもそのデバイス特殊ファイルとしての性質 (型) が異なる場合、その違いを区別します。

- `r` – 文字型 (raw) ディスク・デバイス。ブロック型デバイスのデバイス特殊ファイルには接頭語がありません。
- `n` – クローズ時非巻戻し型のテープ・デバイス。クローズ時巻き戻し型のテープ・デバイスには接頭語がありません。

タイプは、2 文字または 3 文字のドライブ名です。SCSI ディスク・デバイスは `rz`、テープ・デバイスは `rmt` になります。

次に示すように、番号はデバイスのユニット番号です。

- SCSI デバイスの場合、ユニット番号は次の式で計算します。

```
unit = (bus * 8) + target
```

ディスク・デバイス `HSZ40` および `HSZ10` の場合、ユニット番号の前に 1 文字付けて、LUN を表すことがあります。この場合、`a` は LUN 0 を表し、`b` は LUN 1 を表すという具合になります。LUN 0 に文字 `a` を付ける必要はありません。これは省略値だからです。

- テープ・デバイスの場合、接頭語は 0 ~ 7 の連番になります。

接尾語は、次のように同じ物理デバイスに対する複数のデバイス特殊ファイルを区別します。

- ディスクのパーティションは `a ~ h` の文字を使用して表します。ディスク・デバイスごとに合計 16 のファイルが作成されます。内訳は、文字型デバイス・パーティション `a ~ h` に対して 8 個、ブロック型デバイス・パーティション `a ~ h` に対して 8 個です。
- テープ・デバイスは接尾語を用いてテープの密度を表します。テープ・デバイスごとに 8 個までのファイルが作成されます。その場合、表 1-2 に示す接尾語を使って、密度ごとに 2 つのファイルが作成されます。

表 1-2: 旧式のデバイス特殊ファイルに使うテープ・デバイスの接尾語

接尾語	説明
<code>a</code>	密度 QIC-24 (SCSI QIC デバイスの場合)
<code>l</code>	デバイスがサポートする最低密度、または密度 QIC-120 (SCSI インタフェースの QIC デバイスの場合)
<code>m</code>	ドライブが 3 倍密度の場合の中間の密度、または密度 QIC-150 (SCSI インタフェースの QIC デバイスの場合)
<code>h</code>	デバイスがサポートする最高の密度、または密度 QIC-320 (SCSI インタフェースの QIC デバイスの場合)

スクリプトが期待どおりに動作するように、旧式のデバイス命名規則がサポートされています。しかし、新しいデバイス命名規則を通じて使用できる機能は、旧式の命名規則と組み合わせて使うとうまく動かないないことがあります。バージョン 5.0 以降をインストールする場合、インストール時に旧式のデ

バース特殊ファイル (rz13d など) が作成されることはありません。旧式のデバイス特殊ファイル命名規則が必要な場合は、dsfmgr(8) の説明にある適切なコマンドを用いて旧式のデバイス特殊ファイル名を作成しなければなりません。旧式のデバイス特殊ファイルをサポートしないデバイスもあります。

1.5.2.3 新しいデバイス特殊ファイル名

現在のデバイス特殊ファイルには、抽象的なデバイス名が使われており、デバイスの構造やデバイスへの論理パスについての情報は何も得られません。新しいデバイス命名規則は、デバイスの種類を表す名前とインスタンス番号で構成されています。表 1-3 に示すように、この 2 つが、デバイスのベース名の要素になります。

表 1-3: 新しいデバイス特殊ファイル名の例

/dev 内の位置	デバイス名	インスタンス	ベース名
/disk	dsk	0	dsk0
/rdisk	dsk	0	dsk0
/disk	cdrom	1	cdrom1
/tape	tape	0	tape0

デバイス名を、システムが割り当てるインスタンス番号と組み合わせて、dsk0 のようなベース名が作成されます。

現在のデバイス特殊ファイルはデバイスのベース名を使って命名され、対象デバイスに関するより詳しい情報を伝える接尾語が含まれます。接尾語は、次のようにデバイスの種類によって異なります。

- ディスク – デバイス・ファイル名はベース名と a ~ z の接尾語で構成されます。たとえば、dsk0a のようになります。ディスクのパーティションは、a ~ h を使って表します。特に指定しない限り、CD-ROM とフロッピー・ディスクは a と c のみを使用します。たとえば、cdrom1c や floppy0a のようになります。
- raw デバイスに対しては、クラス・レベルのディレクトリ /dev/rdisk に同じデバイス名があります。
- テープ – デバイス・ファイル名は、ベース名と、文字 _d の後に 1 個の整数を続けた接尾語で構成されます。たとえば、tape0_d0 のようになります。この接尾語は、/etc/ddr.dbase ファイルにおけるデバイ

スのエントリに従ってテープ・デバイスの密度を表わします。たとえば次のようになります。

デバイス	密度
tape0	省略時密度
tape0c	圧縮付き省略時密度
tape0_d0	/etc/ldr.dbase のエントリ 0 に対応する密度
tape0_d1	/etc/ldr.dbase のエントリ 1 に対応する密度

新しいデバイス特殊ファイル名の接尾語と旧式のテープ・デバイス名の接尾語には、次に示す直接の対応関係があります。

旧式のデバイス名の接尾語	新しい接尾語
l (low)	_d0
m (medium)	_d2
h (high)	_d1
a (alternate)	_d3

テープ・デバイスについては、次に示すように、2 つの異なるデバイス名のセットが新しい命名規則に従っています。

/dev/tape – このディレクトリには、巻き戻しテープ・デバイス用の名前が含まれます。

/dev/ntape – このディレクトリには、非巻き戻しテープ・デバイス用の名前が含まれます。

使用するデバイス特殊ファイルを判断するには、/etc/ldr.dbase ファイルを調べます。

1.5.2.4 デバイス特殊ファイル名の変換

デバイス特殊ファイルを対象とするコマンドを使用するシェル・スクリプトがある場合には、その中で使用しているオペレーティング・システム付属のコマンドやユーティリティごとに、新しいファイル名と旧式のファイル名に対するサポートが次のように異なるので、注意してください。

- どちらの形式のデバイス名もサポートする。

- 新しいデバイス名のみをサポートする。旧式のデバイス名を使用している場合、このタイプのコマンドは使用できません。
- 旧式のデバイス名のみをサポートする。新しいデバイス名を使用している場合、このタイプのコマンドは使用できません。

ただし、どのデバイスも形式の異なるデバイス名を同時に使用することはできません。シェル・スクリプトがデバイス命名方法に適合しているかテストしてください。コマンドのリファレンス・ページやオンライン・ヘルプを参照してください。

スクリプトを更新する場合は、旧式の名前から対応する新しい名前へ簡単に変換できます。旧式のデバイス名と、それに対応する新しいデバイス名の例をいくつか、表 1-4 に示します。インスタンス番号どうしは何も関係もありません。したがって、旧式のデバイス特殊ファイル `/dev/rz10b` を、現在のシステムの `/dev/disk/dsk2b` に対応付けることができます。

これらの名前を例として使用すれば、スクリプトにあるデバイスは変換できるはずです。dsfmgr(8) コマンドを使ってデバイス名を変換することもできます。

表 1-4: デバイス名の変換例

旧式のデバイス特殊ファイル名	新しいデバイス特殊ファイル名
<code>/dev/rmt0a</code>	<code>/dev/tape/tape0</code>
<code>/dev/rmt1h</code>	<code>/dev/tape/tape1_d1</code>
<code>/dev/nrmt0a</code>	<code>/dev/ntape/tape0_d0</code>
<code>/dev/nrmt3m</code>	<code>/dev/ntape/tape3_d2</code>
<code>/dev/rz0a</code>	<code>/dev/disk/dsk0a</code>
<code>/dev/rz10g</code>	<code>/dev/disk/dsk10g</code>
<code>/dev/rrz0a</code>	<code>/dev/rdisk/dsk0a</code>
<code>/dev/rrz10b</code>	<code>/dev/rdisk/dsk10b</code>

1.5.3 デバイス特殊ファイルの管理

ほとんどの場合、デバイス特殊ファイルの管理はシステム自身が行います。オペレーティング・システムを最初にフル・インストールする際、そのシステムで検出された SCSI ディスクと SCSI テープ・デバイスごとにデバイス特殊ファイルが作成されます。アップデート・インストレーションによってオペレーティング・システムを旧バージョンから更新した場合、新しいデバ

イス特殊ファイルと旧式のデバイス特殊ファイルが混在している可能性があります。ただし、その後で新しい SCSI デバイスを追加すると、特に指定しない限り、`dsfmgr` コマンドは新しいデバイス特殊ファイルのみを作成します。システムをリブートすると、ブート・シーケンスで `dsfmgr` コマンドが自動的に呼び出され、デバイスに対する新しいデバイス特殊ファイルを作成します。また、システムは、`pty` (擬似端末) のような擬似デバイスに必要なデバイス特殊ファイルも自動的に作成します。

SCSI ディスクまたはテープ・デバイスをシステムに追加すると、新しいデバイスが自動的に検出されて認識され、ハードウェア管理データベースに追加され、デバイス特殊ファイルが作成されます。新しいデバイスを追加した後に行う最初のリブートでは、ブート・シーケンスで `dsfmgr` コマンドが自動的に呼び出され、そのデバイスのデバイス特殊ファイルが作成されます。

旧式のデバイス名でしか動作しないアプリケーションをサポートする場合、既存のすべてのデバイスについて、または、最近追加したデバイスについてのみ、旧式のデバイス特殊ファイルを手動で作成する必要があるかもしれません。しかし、ファイバ・チャネルなどの機能をサポートする最近のデバイスには、新しい特殊ファイル命名規則しか使用できないものがあります。

以降の項で、`dsfmgr` コマンドの一般的な使い方を説明します。コマンド構文についての詳細は、`dsfmgr(8)` を参照してください。デバイス特殊ファイルを作成するためにブート時に実行されるシステム・スクリプト・ファイル `/sbin/dn_setup` は、`dsfmgr` コマンド・オプションの使い方の例を示します。

1.5.3.1 一般的な操作の実行 (`dn_setup`)

`/sbin/dn_setup` スクリプトは、デバイス特殊ファイルを作成するために、システムの起動時に自動的に実行されます。`/sbin/bcheckrc` ユーティリティも、システム起動時にデバイス特殊ファイルの確認をします。`/sbin/bcheckrc` によって何らかの問題が検出されると、コンソールに次のメッセージが表示されます。

```
bcheckrc: Device Naming failed initial check.  
Run dsfmgr -v, and if no errors are  
reported, exit or type CTRL-D to continue  
booting normally.
```

通常は `dn_setup` コマンドを実行する必要はありません。デバイス名に関する問題のトラブルシューティングや、壊れたデバイス特殊ファイル・ディレ

クトリあるいはデータベース・ファイルの復元に役立ちます (1.5.3.3 項も参照してください)。システム構成を頻繁に変更したり、別のバージョンのオペレーティング・システムをインストールした場合は、システムの起動中に、システム・コンソールにデバイス関連のエラー・メッセージが表示されることがあります。これらのメッセージは、デバイス特殊ファイル名を割り当てることができないことを示す場合があります。この問題は、保存されている構成を現在の構成に対応付けることができない場合に発生する可能性があります。また、インストレーションとインストレーションの間にデバイスの追加や削除を行った場合にもこの問題が発生することがあります。

管理者が使って便利なオプションは、`-sanity_check` だけです。デバイス名データベースを検証するには、次のコマンドを入力します。

```
# /sbin/dn_setup -sanity_check
Passed.
```

Passed 以外のメッセージが表示された場合には、`dsfmgr` コマンドを使ってデバイス名データベースの検証と修復を行います。問題が深刻で正常にブートできない場合には、技術サポート担当者によって、デバッグと問題解決のための他のコマンドを使うように指示されることがあります。

`dn_setup(8)` を参照してください。

1.5.3.2 デバイス・クラスとカテゴリの表示

システムにあるデバイスのタイプはすべて、「Category to Class-Directory, Prefix Database」ファイル `/etc/dccd.dat` で個々に識別できます。これらのデータベース内の情報は、`dsfmgr` コマンドを使って表示できます。この情報から、システムにどのようなデバイスがあるかがわかります。また、他の `dsfmgr` コマンド・オプションで利用できるデバイス識別属性を知ることできます。たとえば、ディスク・デバイスである、というように、物理的な特性が互いに関連するデバイスのクラスを見つけることができます。デバイスの各クラスは、`/dev` の下に独自のディレクトリを持っています (非巻き戻し型テープ・デバイスの `/dev/ntape` など)。デバイス・クラスは、「Device Class Directory Default Database」ファイル `/etc/dcdd.dat` に格納されています。

データベースのエントリを表示するには、次のコマンドを使用します。

```
# /sbin/dsfmgrp -s
dsfmgr:show all datum for system at /

Device Class Directory Default Database:
# scope mode name
```

```

-- -- -- --
1 1 0755 .
2 c 0755 disk
3 c 0755 rdisk
4 c 0755 tape
5 c 0755 ntape
6 1 0755 none

Category to Class-Directory, Prefix Database:
# category sub_category type directory iw t mode prefix
-- -- -- --
1 disk cdrom block disk 1 b 0600 cdrom
2 disk cdrom char rdisk 1 c 0600 cdrom
3 disk floppy block disk 1 b 0600 floppy
4 disk floppy char rdisk 1 c 0600 floppy
5 disk floppy_fdi block disk 1 b 0666 floppy
6 disk floppy_fdi char rdisk 1 c 0666 floppy
7 disk generic block disk 1 b 0600 dsk
8 disk generic char rdisk 1 c 0600 dsk
9 parallel_port printer * . 1 c 0666 lp
10 pseudo kevm * . 0 c 0600 kevm
11 tape * norewind ntape 1 c 0666 tape
12 tape * rewind tape 1 c 0666 tape
13 terminal hardwired * . 2 c 0666 tty
14 * * * none 1 c 0000 unknown

Device Directory Tree:
12800 2 drwxr-xr-x 6 root system 2048 May 23 09:38 /dev/.
166 1 drwxr-xr-x 2 root system 512 Apr 25 15:58 /dev/disk
6624 1 drwxr-xr-x 2 root system 512 Apr 25 11:37 /dev/rdisk
180 1 drw-r--r-- 2 root system 512 Apr 25 11:39 /dev/tape
6637 1 drw-r--r-- 2 root system 512 Apr 25 11:39 /dev/ntape
181 1 drwxr-xr-x 2 root system 512 May 8 16:48 /dev/none

Dev Nodes:
13100 0 crw----- 1 root system 79, 0 May 8 16:47 /dev/kevm
13101 0 crw----- 1 root system 79, 2 May 8 16:47 /dev/kevm.pterm
13102 0 crw-r--r-- 1 root system 35, 0 May 8 16:47 /dev/tty00
13103 0 crw-r--r-- 1 root system 35, 1 May 8 16:47 /dev/tty01
13104 0 crw-r--r-- 1 root system 34, 0 May 8 16:47 /dev/lp0
169 0 brw----- 1 root system 19, 17 May 8 16:47 /dev/disk/dsk0a
6627 0 crw----- 1 root system 19, 18 May 8 16:47 /dev/rdisk/dsk0a
170 0 brw----- 1 root system 19, 19 May 8 16:47 /dev/disk/dsk0b
6628 0 crw----- 1 root system 19, 20 May 8 16:47 /dev/rdisk/dsk0b
171 0 brw----- 1 root system 19, 21 May 8 16:47 /dev/disk/dsk0c
:
:

```

この表示から、他の dsfmgr コマンドで使える情報を得ることができます (データベースのフィールドについての完全な説明は dsfmgr(8) を参照してください)。たとえば、次のような情報があります。

- class – デバイス・クラス。disk (ブロック型デバイス), rdisk (文字型デバイス), tape (巻き戻し型テープ・デバイス) などがあります。この情報を dsfmgr -a (追加) または dsfmgr -r (削除) コマンド・オプションで使用して、クラスの追加または削除を行います。
- category – デバイス・カテゴリ。たとえば、SCSI ディスク, CD-ROM リーダ, フロッピー・ディスク・リーダーはすべて disk カテゴリに属し

ます。この情報を `dsfmgr -a` (追加) または `dsfmgr -r` (削除) コマンド・オプションで使用する、カテゴリの追加または削除を行います。

1.5.3.3 データベースの検証と修復

異常な状況では、デバイス・データベースが壊れたり、デバイス特殊ファイルが間違っシステムから削除されてしまうことがあります。デバイスを使用できないというエラーが表示されているにもかかわらず、デバイス自体は故障しているように見えないこともあります。デバイス特殊ファイルに問題がありそうな場合は、`dsfmgr -v` (検証) コマンド・オプションを用いてデータベースをチェックします。

注意

システムの起動時にデバイス名の問題を示すエラー・メッセージが表示された場合は、ブートを継続できるように、検証コマンド・オプションを使用しなければなりません。データベースの検証前と検証後に、システム構成をチェックしてください。検証手続きによって、大半の問題が解決され、ブートを継続できます。ただし、検証オプションによってデバイス自身の問題や構成の問題は解決できません。

このような問題はまれで、主に通常と異なる操作 (ブート・ディスクの切り替えなど) を行った場合に発生します。エラーは一般的に、システムが回復不能で、正しく動作する最新の構成のコピーが使用できなかったことを意味します。また、エラーが発生するのは、一般的に、現在のシステム構成がデータベースに合わなくなった場合です。

システムを壊す可能性のあるシステム操作をするときは、システムを直前の構成と同じ状態に戻せて、バックアップから直前のバージョンのオペレーティング・システムを復元できるようにしなければなりません。

たとえば、`mttools` コマンドが使えるようにフロッピー・ディスク・デバイスを構成しようとしたが、デバイスにアクセスできなくなったとします。その場合、次の `dsfmgr` コマンドを使用して、問題を診断するために役立てます。

```
# /sbin/dsfmgr -v
dsfmgr:verify all datum for system at /
```

```
Device Class Directory Default Database:
    OK.

Device Category to Class Directory Database:
    OK.

Dev directory structure:
    OK.

Dev Nodes:
    ERROR:device node does not exist: /dev/disk/floppy0a
    ERROR:device node does not exist: /dev/disk/floppy0c
    Errors:    2

Total errors:  2
```

この出力から、問題のフロッピー・ディスク・デバイスに対応するデバイス特殊ファイルがないことがわかります。この問題を解決するために、同じコマンドに `-F` (修復) フラグを指定して、次のようにエラーを修正します。

```
# /sbin/dsfmgr -v -F

dsfmgr:verify all datum for system at /

Device Class Directory Default Database:
    OK.

Device Category to Class Directory Database:
    OK.

Dev directory structure:
    OK.

Dev Nodes:
    WARNING:device node does not exist: /dev/disk/floppy0a
    WARNING:device node does not exist: /dev/disk/floppy0c
    OK.

Total warnings:  2
```

この例では、`ERROR` が `WARNING` に変わり、フロッピー・ディスクのデバイス特殊ファイルが自動的に作成されることを示しています。`dsfmgr -v` コマンドをもう一度実行しても、エラーはもう表示されません。

1.5.3.4 デバイス特殊ファイルの削除

デバイスをシステムから恒久的に取り外した場合、そのデバイスのデバイス特殊ファイルを削除して、他の種類のデバイスに再度割り当てることができ

ます。次の例に示すように、`dsfmgr -D` コマンド・オプションを使用してデバイス特殊ファイルを削除します。

```
# ls /dev/disk
cdrom0a  dsk0a    dsk0c    dsk0e    dsk0g    floppy0a
cdrom0c  dsk0b    dsk0d    dsk0f    dsk0h    floppy0c

# /sbin/dsfmgr -D /dev/disk/cdrom0*
-cdrom0a -cdrom0a -cdrom0c -cdrom0c
# ls /dev/disk
dsk0a    dsk0c    dsk0e    dsk0g    floppy0a
dsk0b    dsk0d    dsk0f    dsk0h    floppy0c
```

削除前の `ls` コマンドの出力にデバイス特殊ファイル `cdrom0` があることに注目してください。この例では、ワイルドカード文字 (*) を使うことによってすべての `cdrom` デバイスを対象にして `dsfmgr -D` コマンド・オプションを実行し、そのサブカテゴリに属するすべてのデバイス特殊ファイルを恒久的に削除しています。削除コマンドの後のメッセージには、ベース名 (`cdrom0`) が 2 回繰り返して現れています。これは、対象となる `raw` または文字型特殊ファイルがあった `/dev/rdisk` ディレクトリからもデバイス特殊ファイルを削除したためです。

ハードウェアの変更がないのに、デバイス特殊ファイルを誤って削除してしまった場合には、次の手順でファイルを再作成します。

```
# /sbin/dsfmgr -n cdrom0a

+cdrom0a +cdrom0a
# /sbin/dsfmgr -n cdrom0c

+cdrom0c +cdrom0c
```

1.5.3.5 デバイス特殊ファイル名の移動と交換

`dsfmgr -m` (移動) コマンド・オプションを使用して、デバイス特殊ファイルをデバイス間で移動する (再割当てする) ことができます。また、`-e` オプションを使用して、デバイス特殊ファイルを他のデバイスと交換することもできます。たとえば次のようにします。

```
# /sbin/dsfmgr -m dsk0 dsk10
# /sbin/dsfmgr -e dsk1 15
```

次の手順は、これらのコマンド・オプションを使用してテープ・デバイスの交換する方法を示す例です。この例では、故障した内蔵テープ・ドライブの緊急代替装置として、デスクトップ型の TLZ06 (DAT) SCSI ドライブを使いま

す。この手順は、内蔵テープ・ドライブの交換やテープ・チェンジャ (ジュークボックス) にマウントされているドライブの追加と削除にも適用できます。

この手順は、利用可能な SCSI アドレスと (必要に応じて) 適切な SCSI 空きポートを持つすべての構成に適用されます。空いている SCSI バスがなければ、故障したテープ・ドライブを先に取り外してから代替機を取り付ける必要があるかもしれません。テープ・ドライブを追加したら、システムを再起動する必要があります。一般的にシステムが再起動に要する時間だけファイル・システムが利用できないことをユーザに知らせてください。

以下の手順では、既存のデバイスの名前を新しくインストールするデバイスに付け直したいものとし、システムが新しいデバイスに自動的に割り当てる名前を使わないものとします。たとえば、システムに接続されている3つのデバイスのうち、テープ・デバイス `tape0` を使うバックアップ・スクリプトを作成したとします。使われていないのは、`tape1` と `tape2` です。`tape0` が故障し、これを新しいデバイスに置き換えると、新しいデバイスには `tape3` という名前が割り当てられます。しかし、バックアップ・スクリプトはなくなったデバイスを指定しているので、処理が失敗します。新しいデバイス名を指定するようにスクリプトを書き換えるか、デバイスの名前を変えて、故障したデバイスの名前にする必要があります。

または、デバイス名の自動作成機能の利点を考慮して、ローカルのスクリプトとユーティリティがデバイス名に依存しないようにします。そうすれば、スクリプトはハードウェア構成の変更にかかわらず動作し続けられます。`dsfmgr` と `hwmgr` コマンドを使えば、システム構成を照会して、デバイスの特性と属性を動的に知ることができます。この方法の場合、本来のターゲットが使用不可能になったときに正常なデバイスにフェールオーバーできるため、より効率的です。

この手順の実行に先立ち、必ず以下を実行してください。

- 代替のテープ・ドライブと適切な接続ケーブルを入手します。システムのオンライン仕様書を参照し、サポートされているオプションに関する情報を入手します。たとえば、『*AlphaServer ES40 QuickSpecs*』などを参照します。旧式のプロセッサの場合、サポートされているオプションのリストに、テープ・デバイスの最近のモデルが含まれていないことがあります。しかし、使用に関して制限があれば、デバイスに付属のインストール・ドキュメントに記載されているのが一般的です。

注意

テープ・ドライブが正しい SCSI モード (SCSI 2, Fast Wide など) をサポートしていて、使用する SCSI バス・アダプタに対応していることを必ず確認してください。よくわからないときは、技術サポート窓口までご連絡ください。

- システムから削除するデバイス、および、システムへ追加するデバイスのオーナーズ・マニュアルを用意します。オーナーズ・マニュアルには、ここに書かれていない、安全にかかわる重要な情報が書かれています。また、システムの損傷を防ぐための重要な情報が含まれていることもあります。手順の実施にあたっては、オーナーズ・マニュアルの説明に従ってテープ・ドライブの SCSI ターゲットを設定する必要があるかもしれません。
- 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、「*Firmware Updates*」Web ページで入手できます。
- hwmgr コマンドの使い方の詳細については、hwmgr(8) を読んでください。

手順は次のとおりです。

- hwmgr コマンドを使って、次のようにすべての SCSI デバイスに関する情報を表示します。

```
# /sbin/hwmgr show scsi
```

SCSI			DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICEID	HOSTNAME	TYPE	SUBTYPE	OWNER	PATH	FILE	VALID PATH
32:	0	f2394	disk	none	2	1	dsk0	[0/0/0]
33:	1	f2394	disk	none	2	1	dsk1	[0/1/0]
34:	2	f2394	cdrom	none	0	1	cdrom0	[0/4/0]
35:	3	f2394	disk	none	0	1	dsk4	[0/2/0]
41:	4	f2394	tape	none	0	1	tape0	[1/3/0]

既存の SCSI デバイスの記録をとっておくために、出力をパイプに通してファイルに書き出してください。

このコマンド出力は、この手順にとって重要な以下の情報を提供します。

- ハードウェア識別子 (HWID) は、システムによってデバイスに割り当てられる整数値です。上の例では、HWID 41 (tape0) が故障したデバイスです。

- SCSI DEVICEID (デバイス識別子) は整数値です。これは、`did`とも呼ばれます。この数値は、デバイスを SCSI データベースから削除するときに指定します。
- DEVICE FILE は、`device:instance` という形式の文字列です。たとえば、テープ・ドライブ 1 の場合は `tape1` となります。これは、巻き戻し型のテープ・デバイスを表すデバイス特殊ファイル `/dev/tape/tape1` の場所を示す省略形です。非巻き戻し型のデバイスを表すデバイス特殊ファイルは `/dev/ntape` ディレクトリにあります。
- FIRST VALID PATH は、SCSI バス、ターゲット、論理ユニット番号 (LUN) を表す文字列です。この文字列は、3 つの整数値で構成され、`N/N/N` という形式です。このデータは、デバイスへの最初の有効なパスを示します。

テープ・ドライブの追加または交換をするとき、新しいデバイスで利用できる空いている SCSI バス・ターゲットを知る必要があります。TLZ06 のようなテープ・デバイスは多くの場合、SCSI ターゲットを手作業で設定する方法があり、オーナーズ・マニュアルに記載されています。未使用のアドレスを書き留めておいてください。先の例では、バス 1 は、アドレス位置 3 にデバイスが 1 つだけあります。これが既存のテープ・デバイスです (1/3/0)。したがって、テープ・デバイスの SCSI ターゲットは 1, 2, または 4 ~ 6 に設定できます。

注意

`hwmgr` コマンドの出力で、VALID PATH 欄に空のフィールドがある場合、バス、ターゲット、LUN アドレスの欠如はデバイスの I/O に問題があります。これがデバイスの故障の原因かもしれません。この手順の終わりにあるトラブルシューティングの説明で、この可能性について取り上げます。

FIRST VALID PATH 欄の 1 つ以上のエントリが空の場合、残りの (割り当て済みの) アドレスを書き留めておいてください。どのアドレスがないかは、手順 3 に従って調べることができます。

2. sysman shutdown または shutdown コマンドを使って、全ユーザに通知を出し、システムをシャットダウンします。オペレーティング・システムを停止するには、-h (halt) オプションを使います。たとえば次のようにします。

```
# shutdown -h +10
"System shutting down to replace backup device \
Back in 15 minutes"
```

3. コンソールのプロンプトで、次のコマンドを実行して現在割り当てられているデバイスのアドレスを知ることができます。

```
>>> show config
```

このコマンドの出力には、各 SCSI バスに接続されているデバイスのリストが含まれています。たとえば、MKB300.3.0.13.0 は、バス B のターゲット 3 に接続されている TLZ06 テープ・デバイスに対応するエントリです。末尾の 3 つの数字は、バス、ターゲット、LUN アドレスです。

手順 1 で利用可能なバス・アドレスがわからなかったら、show config コマンドの出力から知ることができます。

4. 前面パネルのスイッチを使ってシステムの電源を切り、少なくとも 20 秒待ってからケーブルの取り外しを始めます。
5. 以下の手順は、次のいずれかに該当する場合にのみ実施してください。該当しない場合は、手順 6 へ進んでください。
 - テープ・ドライブをシステムから恒久的に取り外し、新しいテープ・ドライブと置き換えない。

(テープ・ドライブを別の場所で使うために、システムから一時的に取り外すことができます。システム・レコードは残るので、テープ・デバイスを後で置き換えてシステム・レコードを再割り当てできます。)
 - 新しいテープ・ドライブをインストールする前に、既存のテープ・ドライブを取り外す必要がある。理由としては、テープ・ドライブが内蔵ベイにインストールされている、場所不足のために論理バス・アドレスを再利用する必要がある、といったことが考えられます。

どちらかに該当する場合には、次のようにします。

- a. テープ・ドライブへの電源を切り、オーナーズ・マニュアルに書かれている手順に従って、システムからテープ・ドライブを物理的に取り外します。
- b. コンソールのプロンプトで、次のようにしてシステムをシングル・ユーザ・モードで起動します。

```
>>> boot -flags s
```

- c. テープ・ドライブを削除するには、次のコマンドを使用します。

```
# /sbin/hwmgrr delete scsi -did NV
```


-did オプションに対して指定している値は、手順 1 で調べた SCSI DEVICEID です。
- d. まれな状況で、削除が完全でないことがあります。デバイスが削除されたかどうか、手順 1 で説明したように hwmgrr show scsi コマンドを再度実行して調べます。また、/dev/tape ディレクトリと /dev/ntape ディレクトリの中を調べて、デバイス特殊ファイルがないことを確認します。

システムにデバイスのレコードが少しでも残っていたら、この手順の終わりにあるトラブルシューティングの説明を参照してください。

6. この手順は、テープ・デバイスを置き換える場合にのみ実施してください。

新しいテープ・ドライブを、オーナーズ・マニュアルの説明に従って接続します。テープ・ドライブの電源とシステムの電源を適切な順番で入れます。

システムをリブートし、次のようにしてシングルユーザ・モードにします。

```
>>> boot -flags s
```

7. ブート・プロシージャを監視し、メッセージに注意します。リブートの過程で、新しいテープ・ドライブは自動的に検出され、次のようなメッセージがコンソールに表示されます。

```
dsfmgr: Note creating device special files
+tape16...
```

このメッセージは、新しいテープ・デバイスが検出され、デバイス特殊ファイルが /dev/tape また /dev/ntape ディレクトリに作成されることを示します。このようなメッセージが表示されなかったら、デバイス特殊ファイルを手作業で作成する必要があります (ただし、先に移行の手順を最後まで実行してみることに)。

8. システムをリブートして、マルチユーザ・モードになったら、次のコマンドを入力して新しく追加したテープ・ドライブに関する情報を表示します。

```
# /sbin/hwmgrr show scsi
      SCSI
HWID: DEVICEID HOSTNAME  TYPE  SUBTYPE OWNER  PATH  FILE  VALID PATH
-----
48:   5           f2394   tape  none    0      1   tape16 [1/4/0]
```

このコマンド出力を、手順 1 の出力と見比べると、新しいテープ・ドライブが tape16、HWID 48 として表示されているのがわかります。新しいテープ・ドライブの DEVICE FILE (デバイス特殊ファイル) インスタンス番号が 16 で、元のテープ・ドライブの番号 (0) よりもかなり大きいことに注目してください。

9. デバイスを交換したら、次の作業も行う必要があるかもしれません。
 - a. 既存の (故障していると思われる) テープ・ドライブのデバイス名を交換ドライブに移す。それには、次のように dsfmgr コマンドに -e (交換) オプションを付けて、DEVICE FILE を指定します。

```
# /sbin/dsfmgr -e tape16 tape0
tape16<==>tape0      tape16_d0<==>tape0_d0 tape16_d1<==>tape0_d1
tape16_d2<==>tape0_d2 tape16_d3<==>tape0_d3 tape16_d4<==>tape0_d4
tape16_d5<==>tape0_d5 tape16_d6<==>tape0_d6 tape16_d7<==>tape0_d7
tape16c<==>tape0c    tape16<==>tape0      tape16_d0<==>tape0_d0
tape16_d1<==>tape0_d1 tape16_d2<==>tape0_d2 tape16_d3<==>tape0_d3
tape16_d4<==>tape0_d4 tape16_d5<==>tape0_d5 tape16_d6<==>tape0_d6
tape16_d7<==>tape0_d7 tape16c<==>tape0c
```

この出力には双頭の矢印 (<==>) が含まれています。これは、デバイス特殊ファイルの名前が交換されたことを示します。次のようにして変更を確認します。

```
# /sbin/hwmgrr show scsi
      SCSI
HWID: DEVICEID HOSTNAME  TYPE  SUBTYPE OWNER  PATH  FILE  VALID PATH
-----
48:   5           f2394   tape  none    0      1   tape0 [1/4/0]
```

SCSI アドレス 1/4/0 にあるテープ・ドライブ (新しく追加したドライブ) の名前が tape0 に変わっています。

- b. デバイス特殊ファイルの番号を低い番号または 0 にする。たとえば、dump コマンドのデフォルト・デバイスは tape0_d0 (デフォルトの密度のテープ 0) です。複数のテープ・ドライブを持つテープ・チェンジャ (ジュークボックス) を使っている場合には、ローカルのスクリプトが期待どおりに動作するようにデバイスの番号を付け替えなければならないことがあります。

それには、次のように dsfmgr コマンドに -m (移動) オプションを付けて、DEVICE FILE を指定します。

```
# /sbin/dsfmgr -m tape16 tape1
```

```
tape16=>tape1      tape16_d0=>tape1_d0 tape16_d1=>tape1_d1
tape16_d2=>tape1_d2 tape16_d3=>tape1_d3 tape16_d4=>tape1_d4
tape16_d5=>tape1_d5 tape16_d6=>tape1_d6 tape16_d7=>tape1_d7
tape16c=>tape1c     tape16=>tape1      tape16_d0=>tape1_d0
tape16_d1=>tape1_d1 tape16_d2=>tape1_d2 tape16_d3=>tape1_d3
tape16_d4=>tape1_d4 tape16_d5=>tape1_d5 tape16_d6=>tape1_d6
tape16_d7=>tape1_d7 tape16c=>tape1c
```

この出力には、デバイス特殊ファイル名が移動されたことを示す一方向の矢印が含まれています。次のようにして変更を確認します。

```
# /sbin/hwmgrr show scsi
      SCSI
HWID: DEVICEID HOSTNAME  TYPE    SUBTYPE  OWNER  PATH  FILE  VALID PATH
-----
46:   4          f2394   tape    none     0      1   tape1 [0/5/0]
48:   5          f2394   tape    none     0      1   tape0 [0/6/0]
```

SCSI アドレス 0/5/0 にあるテープ・ドライブの名前が tape1 に変わっています。

この手順に含まれているチェックポイントによって、作業は問題なく終わるはずですが、しかしまれに、新しいテープ・ドライブの追加または既存テープ・ドライブの取り外しが失敗することがあります。その場合には、次のトラブルシューティング手順に従って問題を解決できます。

1. フィールド・サービス窓口に連絡する必要がある場合に備えて、どのオプションを試したか記録を残しておくようにしてください。

hwmgrr と dsfmgr コマンドをさまざまに組み合わせる試行錯誤による方法は避けてください。ほとんどの場合、問題の原因はデバイスに係るデータベースのレコードが中間的な状態になっていることです。コマンド・オプションをあれこれ試すと、かえって問題が複雑になる可能性があります。

2. Event Manager (EVM) ビューアを起動し、優先度 300 以上のイベントが表示されるように構成されていることを確認します。

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

イベント・ビューアを使って、手順の実行によって生成されるハードウェア・イベントを確認したり、エラーを発見したりできます。最新のイベントを見るには、イベント・ビューアの表示を更新する必要があります。イベント・ビューアには、テープ・ドライブが検出され、新しいベース名 (tape*) とハードウェア識別子 (HWID) が割り当てられたことを示すイベントが表示されます。システムがこれらのイベントを通知するまで少し時間がかかったり、イベントを見るためにイベント・ビューアを更新する必要があったりするかもしれません。

3. SCSI バスをスキャンし、検出されたすべての新しいデバイスに識別子を割り当てるには、次のコマンドを使います。

```
# /sbin/hwmgrr scan scsi
hwmgrr: Scan request successfully initiated
```

ハードウェア・スキャンは非同期で実行されるので、システム・プロンプトにすぐに戻りますが、それはスキャンが終わったことを示すわけではありません。SCSI デバイスの数が多いシステムでは、スキャンに数分かかることがあります。スキャンが完了するまで待ってください。通常、スキャンはシステム起動時に自動的に行われるため、処理が完了してもメッセージが表示されません。

4. スキャンが完了したら、次のコマンドを実行してテープ・ドライブが検出されたことを確認します。

```
# /sbin/hwmgrr show scsi
```

	SCSI		DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICEID	HOSTNAME	TYPE	SUBTYPE	OWNER	PATH	FILE	VALID PATH
46:	4	f2394	tape	none	0	1	tape1	[0/5/0]
48:	5	f2394	tape	none	0	1		[1/4/0]

5. 手順 4 の hwmgrr の出力に、含まれているべきデバイス特殊ファイルが含まれていなければ、新しく追加したデバイスのデバイス特殊ファイルが作成されているかどうかを確認します。必要に応じて、/dev/tape ディレクトリまたは /dev/ntape ディレクトリの内容を確認します。

デバイス特殊ファイルが見つからなければ、dsfmgr コマンド (dsfmgr(8) を参照) に -K オプションを指定して、次のようにデバイス特殊ファイルを作成します。

```
# /sbin/dsfmgr -K
+tape16 +tape16_d0 +tape16_d1...
```

6. 以上の手順で問題が解決しなければ、デバイス・データベースが中間的な状態になっている可能性があります。テープ・ドライブの HWID またはデバイス特殊ファイル名を指定して、`/etc/dfsc.dat` ファイルの中でテープ・ドライブのエントリを探します。たとえば、テープ・ドライブがリスト出力に `tape0` と表示されている場合には、次のコマンドを入力します。

```
# grep "tape0" /etc/dfsc.dat
```

次のような出力が得られます。

```
A: 0 130003e 48 9 6 c "" /dev/tape tape 0
```

この出力の第 4 欄にハードウェア識別子 (HWID) があります。ここでは 48 です。

7. 前の手順でわかった HWID 48 を指定し、次のコマンドを使ってデバイスの状態を確認します。

```
# /sbin/hwmgr show component -id 48
```

コマンドから何か出力された場合には、技術サポート窓口までご連絡ください。さもなければ、次の手順に進みます。

8. 依然としてデバイスにアクセスできなければ、システムを再起動してデバイス・データベースをリセットします。システムを再起動することもできなければ、技術サポート窓口までご連絡ください。

1.5.3.6 デバイス特殊ファイルの番号の付け替え

システムのコンポーネントを再構成すると (たとえば、デバイスを削除するなど)、デバイス特殊ファイルに割り当てられているインスタンス番号がインクリメントされます。時間がたつにつれて、デバイス特殊ファイルが無作為に番号付けされているように見えてきます。また、クローニング、コピー、削除の手順を用いてシステムをバックアップした場合、バックアップのたびにインスタンス値が大きくなり、デバイス名が長くなって管理しにくくなってきます。クローニング・バックアップ・プログラムも、バックアップのたびに新しいデバイス・インスタンスを知る必要があるため、デバイス特殊ファイル名の変更によって、バックアップに要する時間が長くなります。

`dsfmgr -vI` コマンド・オプションを使うことによって、各デバイスのインスタンス番号を可能限り最小の値にリセットできます。システムの構成が

バックアップ・プロシージャを除いて固定の場合、このオプションを使うことで、クローニングされるバックアップ・デバイスは、セットが改まるたびに同じ新しい名前を持つようになり、バックアップ・プロシージャが簡単になります。次の手順は、dsfmgr -vI コマンドを使ってすべてのデバイスのインスタンス値を可能な限り最小の値にリセットする方法を示します。この手順は、dsfmgr -vI コマンドの仕組みを示します。

1. 次の (一部省略された) 出力は、新しくインストールされたシステムでテープ・ドライブのデバイス特殊ファイルがどのように番号付けされているかを示します。

```
# /sbin/hwmgrr view device
61: /dev/ntape/tape0    COMPAQ   SDT-10000
bus-5-targ-0-lun-0
```

2. システムの構成に多数の変更が加えられた状況を再現するために、次のコマンドでテープ・ドライブのデバイス特殊ファイルの番号を 40 にします。

```
# /sbin/dsfmrr -m tape0 tape40
tape0=>tape40  tape0_d0=>tape40_d0  tape0_d1=>tape40_d1
tape0_d2=>tape40_d2  tape0_d3=>tape40_d3
tape0_d4=>tape40_d4.....
```

3. 次の (一部省略された) 出力は、テープ・ドライブのデバイス特殊ファイルの番号が大きくなって tape40 に変わったことを示します。

```
# /sbin/hwmgrr view device
.61: /dev/ntape/tape40    COMPAQ   SDT-10000
bus-5-targ-0-lun-0
```

4. システムからデバイスを削除した状況を再現するために、次のコマンドでテープ・ドライブのデバイス特殊ファイルの番号を 10 にします。

```
# /sbin/dsfmrr -m tape0 tape40
tape40>tape10
```

これは、11 から 39 の範囲で未使用のデバイス特殊ファイルのインスタンスがあることを意味します。しかし、システムは現時点ではこれらのインスタンスが空いていることを知らないので、新しくインストールされたデバイスをインスタンス番号 41 から番号付けをします。

5. システムが可能な限り小さいインスタンス番号を割り当てるようにするには、次のコマンドを実行します。

```
# /sbin/dsfmrr -vI
dsfmrr: verify all datum for system (version) at /
```

Default File Tree:

OK.

Device Class Directory Default Database:

OK.

Device Category to Class Directory Database:

OK.

Dev directory structure:

OK.

Device Status Files:

OK.

Dev Nodes:

OK.

Minimize next instance values:

tape 41 => 11

“Minimize next instance values:” のメッセージは、利用可能な最小のインスタンス値が 11 であることを示します。

6. 処理が正しく行われたことを確かめるために、次のコマンドを使ってテープ・デバイスの削除と再スキャンを行い、システムが新しいデバイス特殊ファイルにインスタンス番号 11 から番号を付けるようにします。

```
# /sbin/hwmgrr delete scsi -did 3
hwmgrr: The delete operation was successful
# /sbin/hwmgrr scan scsi
hwmgrr: Scan request successfully initiated
# /sbin/hwmgrr show scsi
84: 3 argol tape none 0 1
tape11 [5/0/0]
```

これでテープ・デバイスのデバイス特殊ファイルが 11 番になり、システムが利用可能な最小のインスタンス値を使ったことがわかります。

このオプションを自分のシステムで使うには、バックアップ・スクリプトを修正するか、適切なタイミングで `dsfmgr -vI` コマンド・オプションを実行する新しいスクリプトを作成します。あるいは、ハードウェアの構成を変えるたびにこのコマンドを実行するようにします。

SysMan Menu と SysMan Station の使用

本章では、システム・ハードウェアの管理に役立つ利用可能な SysMan Menu タスクについて説明します。ユーティリティは、シングル・システムでもクラスタ・システムでも使用できます。SysMan Menu は、別のオペレーティング環境からリモートで実行したり、Web アプリケーションとして起動したりできます。

本章では、SysMan Station を使ってハードウェアのステータスを監視し、ハードウェア管理ユーティリティを起動する方法を説明します。SysMan Station は、共通デスクトップ環境 (CDE) または他の X Window ユーザ環境で動作する X Window に準拠するグラフィカル・ユーザ・インタフェース (GUI) です。

この章では、次の項目について説明します。

- 次の SysMan Menu ハードウェア・タスクの使い方
 - ハードウェア階層の参照方法 (2.1.1 項)
 - クラスタの参照方法 (2.1.2 項)
 - デバイス情報の参照方法 (2.1.3 項)
 - CPU (Central Processing Unit) 情報の参照方法 (2.1.4 項)
- SysMan Station を使ってコンポーネントとデバイスを検出して監視する方法 (2.2 節)

この章の情報と例は、Tru64 UNIX のコマンド行または CDE から SysMan Menu と SysMan Station を起動して使う方法を示します。Web など、他のオペレーティング環境を使う方法については、『システム管理ガイド』を参照してください。

ほとんどのハードウェア管理の操作に root ユーザの特権が必要ですが、SysMan の DOP (Division of Privileges) 機能を用いて、その特権を root 以外のユーザに割り当てることもできます。詳細は、dop(8) を参照してください。

SysMan Menu は次のサービス・タスクも提供します。

- CPU の管理
- OLAR (Online Addition/Replacement) ポリシー情報

これらの機能の詳細は、`hwmgr_ops(8)` と『*Managing Online Addition and Removal*』を参照してください。

2.1 SysMan Menu ハードウェア・タスクの使用

SysMan Menu タスクには、`hwmgr` コマンドを実行した際にそのコマンド行から使用できる多くのハードウェア管理オプションがあります。`hwmgr` コマンドとそのオプションについては、第 3 章 にさらに詳細な説明があります。コマンド構文とオプションすべてのリストは、`hwmgr(8)` を参照してください。SysMan Menu にあるハードウェア・タスクでヘルプ・オプションを選択すると、対応するリファレンス・ページが表示されます。

『システム管理ガイド』で説明したように SysMan Menu を起動すると、メニューの [ハードウェア] ブランチでハードウェア管理オプションが使えるようになります。このブランチをダブルクリックして開くと、次のタスクが表示されます。

- ハードウェア階層の参照
- クラスタの参照
- デバイス情報の参照
- CPU (Central Processing Unit) 情報の参照

これらのタスクを選択すると、以降の項で説明する基本的なハードウェア管理タスクが起動されます。OLAR (Online Addition/Replacement) についての説明は、『*Managing Online Addition and Removal*』を参照してください。

次のオプション・ボタン (端末では選択肢) は、すべてのタスクで使用できます。

- 再実行 – コマンドを再度実行し、表示された情報を更新します。
- 停止 – コマンドを停止します。[再実行] オプションを使用すると情報が更新され、[了解] を選択すると終了します。
- 了解 – タスクを終了し、ウィンドウを閉じます。
- ヘルプ – リファレンス・ページを表示します。

2.1.1 ハードウェア階層の参照

[ハードウェア階層の参照] タスクは、`/sbin/hwmgr view hierarchy` コマンドを起動します。次の例は、クラスタの一部として構成されていない単一の CPU システムによる出力を示したものです。

ハードウェア階層の参照

```
HWID:  hardware component hierarchy
-----
1:  platform AlphaServer 800 5/500
2:    cpu CPU0
4:    bus pci0
5:      connection pci0slot5
13:        scsi_adapter isp0
14:          scsi_bus scsi0
30:            disk bus-0-targ-0-LUN-0 dsk0
31:            disk bus-0-targ-4-LUN-0 cdrom0
7:      connection pci0slot6
15:        graphics_controller trio0
9:      connection pci0slot7
16:        bus eisa0
17:          connection eisa0slot9
18:            serial_port tty00
19:          connection eisa0slot10
(以下、省略)
```

このタスクを使用して、システムやクラスタ全体のハードウェア階層を表示します。この階層には、CPU から、ディスクやテープなどの個々の周辺機器までの、各バス、コントローラ、およびシステム上のその他のコンポーネントが示されます。デバイスの多いシステムやクラスタでは、出力が長くなり、スクロールしなければ出力の先頭部分にあるコンポーネントが見えないこともあります。

この出力には、コンポーネントのさらに詳細な情報の調査やデバイスの追加または削除など、ハードウェア管理操作を実行するために `hwmgr` コマンド・オプションに指定できるコンポーネント情報が含まれているので役に立ちます。出力される階層に示される次のような項目を、コマンドへの入力として使用できます。

- HWID – ハードウェア識別子 (または `id`)。階層内の個々のエントリに振られる一意の整数値。
- コンポーネント名。PCI (Peripheral Component Interconnect) バスを表す `pci` など。

- コンポーネントのベース名。後ろに、コンポーネントを識別する整数が続きます。たとえば、`cdrom0` はそのコンポーネントのデバイス特殊ファイル (`/dev/disk/cdrom0`) に対応します。デバイス特殊ファイル名についての詳細は、1.5 節に記載されています。
- 物理位置の属性。デバイスのアドレスまたはバスを表します。たとえば、`bus-0-targ-0-LUN-0` です。0/0/0 と出力されることもありますが、次のような情報を表しています。
 - `scsi-0` はバスを表しており、対象コンポーネントが取り付けられているバスの番号を示す。
 - `targ-0` は、バス上での対象コンポーネントのターゲット番号を示す。この場合は、バス 0 の最初のターゲットであることを表しています。
 - `LUN-0` は、論理ユニット番号 (LUN) を示す。この場合は、バス 0 におけるターゲット 0 の最初の論理ユニット番号であることを表しています。
- デバイスのハードウェア・カテゴリ。`bus` や `ide_controller` など。
- スロットへの接続。`pci0slot5` や `eisa0slot9` のように、デバイスのスロット番号を表します。
- バス、コントローラ、およびコンポーネントの関係。たとえば、表示例にある以下の行は、バス `scsi_bus scsi0` 上のコントローラ `scsi_adapter isp0` に接続された 2 台のディスク・デバイスを表しています。

```
13:          scsi_adapter isp0
14:          scsi_bus scsi0
30:          disk bus-0-targ-0-LUN-0 dsk0
31:          disk bus-0-targ-4-LUN-0 cdrom0
```

同じコンポーネントが (共用バスなどで) 共用されている場合、階層に 2 回以上現れることがあります。その場合、それぞれが異なる ID で表示されます。共用デバイスの例を、3.4.7 項に示します。

特定のハードウェア・コンポーネントに対する操作に焦点を絞りたい場合は、`view hierarchy` コマンドの出力情報を、`hwmgr` コマンドで次のように使用します。次のコマンドは、HWID (`id`) が 30 であるコンポーネントについて、コンポーネント属性 `device_starvation_time` の値を取得する例

です。コンポーネント 30 は、例に示した階層内で bus 0 , target 0 , LUN 0 の位置にある SCSI ディスクです。

```
# /sbin/hwmmgr get attribute -id 30 \  
-a device_starvation_time  
30:  
    device_starvation_time = 25 (settable)
```

この出力は、device_starvation_time 属性の値が 25 であることを示しています。ラベル (settable) は、これが構成可能な属性であり、次のようなコマンド・オプションで設定できることを示します。

```
# /sbin/hwmmgr set attribute -id 35 \  
-a device_starvation_time=30
```

コンポーネント属性の値を変更する前に、変更による影響を理解しておく必要があります。デバイスに付属のドキュメントを参照してください。

2.1.2 クラスタの参照

[クラスタの参照] タスクを選択すると、/sbin/hwmmgr view cluster コマンドが起動され、次のような出力が SysMan Menu のウィンドウ (端末の場合は画面) に表示されます。

クラスタの参照

/sbin/hwmmgr -view cluster を開始しています...

/sbin/hwmmgr view cluster を実行しています: 金 5月 21 13時42分37秒 JST 1999

Member ID	State	Member HostName
-----	----	-----
1	UP	rene (localhost)
31	UP	witt
10	UP	rogr

このコマンドをクラスタのメンバになっていないシステムで実行しようとすると、次のようなメッセージが表示されます。

hwmmgr: This system is not a member of a cluster.

クラスタの特定メンバに対する操作に焦点を絞りたい場合は、次の例のように、hwmmgr コマンドで Member ID と HostName を指定できます。

```
# /sbin/hwmmgr scan scsi -member witt
```

2.1.3 デバイス情報の参照

[デバイス情報の参照] タスクを選択すると `/sbin/hwmgrr view devices` コマンドが起動され、出力が SysMan Menu のウィンドウ (端末の場合は画面) に表示されます。

このオプションを使って、システムまたはクラスタ全体のコンポーネント情報を表示させます。出力には、システムに接続されたすべてのコンポーネントおよび擬似デバイス (`/dev/kevm` など) が表示されます。次に示すのは、クラスタを構成していない、小型の単一 CPU システムの出力例です。

デバイス情報の参照

`/sbin/hwmgrr -view devices` を開始しています...

`/sbin/hwmgrr view devices` を実行しています :
金 5月21日 14時20分08秒 JST 1999

HWID:	Device	Special File	Mfg Model	Location
	Name			
3:	/dev/kevm			
28:	/dev/disk/floppy0c		3.5in floppy	fdi0-unit-0
30:	/dev/disk/dsk0c	DEC	RZ1DF-CB (C) DEC	bus-0-targ-0-LUN-0
31:	/dev/disk/cdrom0c	DEC	RRD47 (C) DEC	bus-0-targ-4-LUN-0

このコマンドが対象とするコンポーネントは、属性 `dev_base_name` を持つとともにデバイス特殊ファイル (DSF) に対応している、階層内のエンティティすべてです。このコマンドからは、次の情報が出力されます。この情報を `hwmgrr` コマンドで使用して、そのデバイスに対するハードウェア管理操作を実行します。

- ハードウェア識別子 (HWID または `id`)。階層内の個々のエントリに振られる一意の整数値。
- DSF 名。 `/dev/disk/cdrom0c` など。ディスク・デバイスの場合には、ディスク領域全体を表す `c` パーティションのデバイス特殊ファイル名になります。テープ・デバイスの場合には、デバイスの省略時密度に対応するデバイス特殊ファイル名を表します。これらの名前については、1.5 節を参照してください。
- 製造元の型番を示す型式、または `3.5in floppy` のような一般的な名前。

- デバイスの物理位置。たとえば、SCSI bus-0-targ-0-LUN-0 です。0/0/0 と出力されることもありますが、次のような内容を表しています。
 - bus-0 - 対象コンポーネントが接続されているバスの番号。ここでは 0 です。
 - targ-0 - バスにおけるこのコンポーネントのターゲット番号。この場合は、バス 0 の最初のターゲットを表しています。
 - LUN-0 - 論理ユニット番号。この場合は、バス 0 の最初の論理ユニット番号を表しています。

前に示した出力例では、フロッピー・ディスク・インタフェース fdi に接続されているフロッピー・ディスクも、デバイス 0、ユニット 0 として示されています。

これらの情報を hwmgr コマンドに指定することにより、デバイスに対するハードウェア管理操作を実行します。次の例は、あるディスクに対して、それに対応するデバイス特殊ファイルを指定し、そのディスクのランプ (LED) を 30 秒間点滅させるコマンドの例です。

```
# /sbin/hwmgr locate -id 60 -time 90
hwmgr: Locate request successfully initiated
```

前記のコマンドは、特定の SCSI デバイスでのみ動作し、HSV110 のようなマネージド・アレイの一部を構成するディスクに対しては動作しないことがあります。ただし、ストレージ・アレイは通常、故障したディスクを検出すると、故障を知らせるために黄色または赤色のディスク・エラー・ライトを点滅させます。

2.1.4 CPU 情報の参照

[CPU (Central Processing Unit) 情報の参照] タスクを選択すると、`/usr/sbin/psrinfo -v` コマンドが起動され、出力が SysMan Menu のウィンドウ (端末の場合は画面) に表示されます。このオプションを使って、CPU の状態情報を表示させます。シングル・プロセッサ・システムの場合、出力は次のようになります。

このタスクの出力は、プロセッサと、その状態を示します。

```
                                /usr/sbin/psrinfo
/usr/sbin/psrinfo -v を開始しています...

/usr/sbin/psrinfo -v を実行しています: 金 5月21日 14:22:05 EDT 1999
```

```
Status of processor 0 as of: 05/21/99 14:22:05
Processor has been on-line since 05/15/1999 14:42:28
The alpha EV5.6 (21164A) processor operates at 500 MHz,
and has an alpha internal floating point processor.
```

2.2 SysMan Station の使用

SysMan Station は、さまざまなウィンドウ環境や Web ブラウザで動作するグラフィカル・ユーザ・インタフェースです。SysMan Station の起動と使用の方法については、『システム管理ガイド』とオンライン・ヘルプを参照してください。ハードウェア管理に役立つ SysMan Station の機能には次のものがあります。

システムとデバイスのモニタリング

SysMan Station は、システムとコンポーネントの現在の状態を表示します。表示をカスタマイズして、システムとクラスタの中で最も関心のある部分に集中することができます。システムにハードウェアの障害が発生すると、GUI で表示されるアイコンの色が変わります。システムは階層表示され、CPU からテープ・デバイスなどの個々のコンポーネントまで、システム全体の位置関係が示されます。バス、コントローラ、およびアダプタのレイアウトは論理アドレス付きで表示されます。それぞれのバスまたはコントローラに接続されているコンポーネントと、そのスロット番号も表示されます。こうした情報は、コマンド・プロンプトから hwmgr コマンドを実行する場合に便利です。

デバイス・プロパティ (または属性) の表示

コンポーネントを選択して、そのデバイスの詳細な属性を表示させることができます。たとえば、SCSI ディスクを選択してマウスの右ボタンを押すと、オプションのメニューが表示されます。このメニューには、選択したディスクのコンポーネント・プロパティを表示させるオプションがあります。このオプションを選択すると、そのコンポーネント・プロパティの詳細なテーブルが表示されます。この操作は、次の出力例 (一部省略) からわかるように、hwmgr コマンドを使用したのと同じことです。

```
# hwmgr get attr -id 30
30:
  name = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
  category = disk
  sub_category = generic
```

```
architecture = SCSI
phys_location = bus-0-targ-0-LUN-0
dev_base_name = dsk0
access = 7
capacity = 17773524
block_size = 512
open_part_mask = 59
disk_part_minor_mask = 4294967232
disk_arch_minor_mask = 4290774015
display truncated
```

ハードウェア管理タスクの起動

デバイスを選択するとともに、必要なコマンドを起動することによって、デバイスの構成や日常的な管理操作を行うことができます。たとえば、ネットワーク・アダプタを選択して、その設定を構成したり、ドメイン・ネーム・サーバ (DNS) の構成といった関連作業を実行します。イベント・ビューアを起動すれば、選択したコンポーネントに関連するシステム・イベント (エラーなど) がポストされているかどうかを調べることができます。

リモート管理オプションについての詳細は、『システム管理ガイド』を参照してください。



hwmgr を用いたハードウェア管理

ハードウェアの管理に使用する主なコマンドは、`hwmgr` コマンド行インタフェース (CLI) です。SysMan タスクなどのその他インタフェースには、`hwmgr` 機能の部分的な機能だけを提供します。たとえば、`hwmgr` であれば、クラスタを構成する全メンバのすべての SCSI アダプタに接続されている同じタイプ (SCSI ディスクなど) の全コンポーネントの属性を設定できます。

ほとんどのハードウェア管理はシステムによって自動的に実行されるので、操作が必要になるのは特定の状況に限られます。たとえば、故障したコンポーネントを交換して、代わりに組み込んだ正常なコンポーネントを故障したコンポーネントと同じ ID で動くようにする場合です。この章では、次の項目について説明します。

- ハードウェア管理モデル (3.1 節)
- `hwmgr` コマンドで利用できる主なユーザ・オプション (3.2 節)
- `hwmgr` コマンド環境の構成 (3.3 節)
- `hwmgr` コマンドを使用した次の管理作業の実施 (3.4 節)
 - SCSI デバイスの場所の確認 (3.4.1 項)
 - ハードウェア階層またはシステム・トポロジの表示 (3.4.2 項)
 - コンポーネントのカテゴリの表示 (3.4.3 項)
 - コンポーネントの属性の表示 (3.4.4 項)
 - 変更可能なコンポーネント属性の設定 (3.4.5 項)
 - クラスタ・ハードウェアの状態の表示 (3.4.6 項)
 - コンポーネントの状態の表示 (3.4.7 項)
 - ハードウェア管理トランザクションの表示 (3.4.8 項)
 - カスタムのワールドワイド識別子 (WWID) の定義 (3.4.9 項)
 - SCSI デバイスの削除 (3.4.10 項)

- RAID アレイのディスクの構成設定 (3.4.11 項)
- 故障した SCSI ディスクの交換 (3.4.12 項)
- 故障した SCSI テープ・ドライブの交換。これは、`redirect` オプションを使って交換することのできないディスクのための代替手順でもあります (3.4.13 項)。
- 故障したクラスタ・ブート・ディスクの交換 (3.4.14 項)
- `name` サブシステムのデータベースの表示 (3.4.15 項)
- `name` 永続データベースからのコンポーネントの抹消と登録解除 (3.4.16 項)
- `refresh` オプションを使った無効なパスの削除とデータベースの最適化 (3.4.17 項)
- コンポーネントの名前の付け替え (3.4.18 項)
- システム内でのコンポーネントの移動 (3.4.19 項)

3.1 ハードウェア管理モデル

オペレーティング・システム・カーネルの内部では、ハードウェアに関するデータは、KSM (Kernel Set Manager) が管理するハードウェア・セットとしてまとめられています。アプリケーションからのリクエストは、ライブラリ・ルーチンによってカーネル・コードまたはリモート・コードに渡されます。後者は、他のシステムとの間でやり取りされるリクエストを処理します。ハードウェア・コンポーネント・モジュール (HWC) はカーネルにあり、この中に、ハードウェア・セット内のハードウェア・コンポーネントを作成したり保守したりするための登録ルーチンがすべて含まれています。また、`dsfmgr` コマンドを用いて実行するデバイス特殊ファイルの管理に必要なデバイス・ノードも含まれています。

ハードウェア・セットは、システムのハードウェア・コンポーネントをすべて記述するデータ構造で構成されています。ハードウェア・コンポーネントは、そのドライバによって登録されると、ハードウェア・セットに組み込まれます。多くのコンポーネントが、機能や内容を表す属性や、操作を制御する属性をサポートしています。属性には、それぞれ値が割り当てられています。これらの属性値は、`hwmgr` コマンドを用いて読み取り、ときには操作することができます。

システム・ハードウェアは、`hwmgr` コマンドでサブシステムとして識別される 3 つの部分にまとめられています。サブシステムは、`component`、`SCSI`、および `name` として識別されます。サブシステムとシステム・ハードウェアの対応は、次のとおりです。

- `component` サブシステムには、(バイナリ形式の) `/etc/dec_hwc_ldb` および `/etc/dec_hwc_cdb` データベースで指定されるすべてのハードウェア・コンポーネントが対応する。このサブシステムには、システムのほとんどのコンポーネントが入ります。
- `name` サブシステムには、バイナリ形式の `/etc/dec_hw_db` データベースにあるハードウェア・コンポーネントすべてが対応する。これは、ハードウェア・トポロジと呼ばれることがよくあります。データベースには、カーネル・ドライバ・フレームワークが維持するハードウェア永続情報や、バス、コントローラ、コンポーネントのデータが含まれます。
- `SCSI` サブシステムには、バイナリ形式の `/etc/dec_scsi_db` データベースにある SCSI デバイスすべてが対応する。SCSI データベースには、SCSI/CAM アーキテクチャが管理するデバイスすべてに対して、それぞれエントリがあります。

`hwmgr` の一般的な特性と機能は、次のとおりです。

- 1 つのコマンドで、広範なハードウェア管理機能を提供する。
- 今はシステムに接続されていないが、直前のブートで認識されたハードウェア・コンポーネントを (簡単な範囲で) 管理できる。
- クラスタの複数のシステムに接続されているハードウェア・コンポーネントを管理できる。
- 管理リクエストを、クラスタの複数のメンバに送信できる。

3.2 `hwmgr` コマンド・オプション

`hwmgr` コマンドは、カーネル内のハードウェア管理モジュールを使って、ハードウェア・コンポーネントの管理を実行します。ハードウェア・コンポーネントの例としては、ディスクやテープといったストレージ・デバイスや、CPU やバスといったシステム・コンポーネントなどがあります。`hwmgr` コマンドを使用すれば、シングル・システムとクラスタのどちらについても、そのハードウェア・コンポーネントを管理できます。

操作コマンドは、コマンド名の後のサブシステム識別子で区別されます。サブシステムは、次のとおりです。 `component` , `name` , `scsi`

`hwmgr` 操作コマンドによっては、2 つ以上のサブシステムに対して使用することができます。 `view` および `show` コマンド・オプションを使って入手したパラメータ情報に従って、実行したい操作の種類に最も関連性の高いサブシステムを使用する必要があります。

コマンドによっては、サブシステム名を指定する必要があります。ただし、ハードウェア・コンポーネントの識別情報を指定した場合には、サブシステム名を指定する必要はありません。 `hwmgr` コマンドは、コンポーネント識別子に基づいて、操作対象の正しいサブシステムを識別することができます。

コマンド・オプションは、タスク・アプリケーション別に編成されています。コマンド・オプション、操作対象のサブシステム、および操作の種類を次の表に示します。

オプション	サブシステム	操作
<code>add</code>	<code>name</code>	データベース管理
<code>delete</code>	<code>component</code> , <code>name</code> , <code>scsi</code>	データベース管理
<code>edit</code>	<code>name</code> , <code>scsi</code>	データベース管理
<code>locate</code>	<code>component</code>	ハードウェア構成
<code>offline</code>	<code>component</code> , <code>name</code>	オンラインでの追加および削除
<code>online</code>	<code>component</code> , <code>name</code>	オンラインでの追加および削除
<code>power</code>	<code>component</code> , <code>name</code>	オンラインでの追加および削除
<code>redirect</code>	<code>scsi</code>	ハードウェア構成
<code>refresh</code>	<code>component</code> , <code>scsi</code>	データベース管理
<code>reload</code>	<code>name</code>	ドライバ構成
<code>remove</code>	<code>name</code>	データベース管理
<code>scan</code>	<code>component</code> , <code>name</code> , <code>scsi</code>	ハードウェア構成
<code>status</code>	<code>component</code>	ハードウェア構成
<code>unconfigure</code>	<code>component</code> , <code>name</code>	ハードウェア構成
<code>unindict</code>	<code>component</code>	オンラインでの追加および削除
<code>unload</code>	<code>name</code>	ドライバ構成

3.3 hwmgr 環境の構成

hwmgr コマンドには、表示される情報量を制御するための環境設定があります。次のコマンドで、省略時の環境設定を表示できます。

```
# /sbin/hwmgr view env

HWMGR_DATA_FILE = "/etc/hwmgr/hwmgr.dat"
HWMGR_DEBUG = FALSE
HWMGR_HEXINTS = FALSE
HWMGR_NOWRAP = FALSE
HWMGR_VERBOSE = FALSE
```

これらの環境変数の値は、ログイン・スクリプトやコマンド行で次の例のようにして設定できます。

```
# HWMGR_VERBOSE=TRUE
# export HWMGR_VERBOSE
```

通常、定義の必要なものは、次の HWMGR_HEXINTS、HWMGR_NOWRAP、および HWMGR_VERBOSE 環境変数の値に限られます。

- HWMGR_HEXINTS 環境変数を TRUE に定義すると、hwmgr コマンドの数値の出力はすべて 16 進数で表示されます。
- HWMGR_NOWRAP 環境変数を TRUE に定義すると、hwmgr コマンドの出力は 80 文字で切り捨てられます。hwmgr コマンド・オプションによる出力は、折り返されて読みにくくなる場合があります。HWMGR_NOWRAP 環境変数の値を TRUE に設定すると、コンソール出力が読みやすくなります。切り捨てられた行には、水平方向の省略記号が付けられます。
- HWMGR_VERBOSE 環境変数を TRUE に定義すると、hwmgr コマンドの出力にはさらに詳細な情報が含まれるようになります。hwmgr コマンドの省略時の設定では、重大でないエラーは表示されません。詳細な情報を表示させる場合は、hwmgr のコマンド・オプションに verbose を追加するという方法もあります。

たとえば、一部のハードウェア・コンポーネントにしか存在しない属性を照会すると、特に指定しない限り、hwmgr コマンドはその属性をサポートするハードウェア・コンポーネントからの出力のみを表示します。次にその例を示します。

```
# /sbin/hwmgr get attribute -a type
6:
  type = local
7:
```

```
type = local
9:
type = MOUSE
```

システムのハードウェア・コンポーネントがすべて `type` 属性をサポートしているとは限りません。`HWMGR_VERBOSE` 環境変数が `TRUE` に定義されていなければ、このコマンドが生成するエラーの中には、表示されないものもあります。それらのエラーを表示させるには、次のように、コマンド行で `-verbose` スイッチを指定します。

```
# /sbin/hwmgrr get attribute -a type -verbose
1: Attribute "type" not defined.
2: Attribute "type" not defined.
4: Attribute "type" not defined.
5: Attribute "type" not defined.
6:
  current type = local
7:
  current type = local
8: Attribute "type" not defined.
9:
  current type = MOUSE
10: Attribute "type" not defined.
11: Attribute "type" not defined.
.
.
(以下、省略)
```

`verbose` スイッチはすべての `hwmgrr` コマンドで使用できますが、追加出力が必ず得られるとは限りません。

3.4 `hwmgrr` を用いたハードウェア管理

この節では、`hwmgrr` コマンドを用いて実行しなければならない作業の例を示します。例の中には、CPU に接続されているコンポーネントが少ない小規模なサーバの管理には向いていないものがあるかもしれません。しかし、ネットワークに接続された多数のシステムで構成される大規模な環境や、コンポーネントが何百個もあるようなクラスタを管理する場合には、非常に役立ちます。`hwmgrr` コマンドを使用すれば、詳細が不明なシステムに接続し、そのコンポーネント階層の情報を入手して、管理作業を行うことができます。システムの構成を事前に調べておく必要も、システムのログやファイルを調べてコンポーネントを探す必要もありません。

3.4.1 SCSI ハードウェアの位置の特定

locate オプションは、現在一部の SCSI デバイスでしか動作しませんが、デバイスの識別に使用できます。このコマンドは、SCSI ディスクの物理的な位置を探す場合に使用できます。次のコマンドを実行すると、SCSI ディスクのランプが 1 分間点滅します。

```
# /sbin/hwmgrr locate -id 42 -time 60
```

このようにして、ディスク・ベイでランプが点滅しているコンポーネントを探せば、目的のディスクの場所を知ることができます。CD-ROM リーダやアレイの一部になっているディスク (たとえば、HSV110) などの SCSI デバイスを見つけるのに、このオプションを使用することはできません。ただし、アレイの一部になっているディスクは故障した場合には検出され、Event Manager のログにイベントがポストされます。アレイ・コントローラは、読み取りランプまたは黄色のエラー表示ランプを点滅させることで故障したディスクを示します。故障したデバイスを手作業で探す必要はありません。

3.4.2 システム階層の表示

view コマンドを使用すると、システム内部のハードウェア階層を表示できます。このコマンドでは、どのアダプタがデバイスを制御しており、バス上のどこにアダプタがインストールされているかを調べることができます。次の例は、クラスタに属さない独立の小規模システムを対象にこのコマンドを実行した場合の、典型的な出力を示します。

```
#!/sbin/hwmgrr view hierarchy
HWID: hardware hierarchy
-----
1: platform AlphaServer 800 5/500
2:   cpu CPU0
6:   bus pci0
7:     connection pci0slot5
15:      scsi_adapter isp0
16:      scsi_bus scsi0
32:        disk bus-0-targ-0-lun-0 dsk0
33:        disk bus-0-targ-4-lun-0 cdrom0
34:        disk bus-0-targ-5-lun-0 dsk1
35:        disk bus-0-targ-6-lun-0 dsk2
36:        disk bus-0-targ-8-lun-0 dsk3
9:     connection pci0slot6
17:      graphics_controller s3trio0
以下、省略
```

コンポーネントの中には、階層内で複数のエントリに現れるものがあります。たとえば、2つのアダプタの間で共用される SCSI バスにディスクがある場合、階層内に同じデバイスに対する2つのエントリが表示されます。SysMan Station GUI を使っても、同じようにシステム・ハードウェアの階層が表示できます。

3.4.3 コンポーネント・カテゴリの表示

同じカテゴリのすべてのコンポーネントを対象にハードウェア管理オプションを実行したり、カテゴリ内の特定のコンポーネントを選択したりする場合には、利用できるコンポーネント・カテゴリを知っている必要があります。ハードウェア・マネージャの `get category` コマンドを使用することにより、どのようなハードウェア・カテゴリがあるかを知ることができます。

このコマンドは `get attributes` と `set attributes` オプションとともに使用すると便利です。これらのオプションは、特定のコンポーネントの属性(またはプロパティ)を表示したり構成したりします。ハードウェア・カテゴリがわかれば、次のように、特定のタイプのハードウェアに限定して属性を照会できるようになります。

```
# /sbin/hwmgr get category
Hardware Categories
-----
category = undefined
category = platform
category = cpu
category = pseudo
category = bus
category = connection
category = serial_port
category = keyboard
category = pointer
category = scsi_adapter
category = scsi_bus
category = network
category = graphics_controller
category = disk
category = tape
```

カテゴリがわかれば、次のようにカテゴリを指定し、的を絞って属性を照会できます。

```
# /sbin/hwmgr get attribute -category platform
1:
  name = AlphaServer 800 5/500
  category = platform
```

この出力から、システム・プラットフォームに ID が 1 のハードウェアがあり、そのプラットフォーム名が AlphaServer 800 5/500 であることがわかります。get attribute コマンド・オプションと set attribute コマンド・オプションも参照してください。

3.4.4 コンポーネント属性の取得

属性はコンポーネントの特性です。コンポーネントのモデル番号のような読み取り専用属性や、動作速度のように、コンポーネントの何らかの動作を制御する値を設定できる属性があります。get attribute コマンド・オプションは、コンポーネントの属性を表示します。ハードウェア・マネージャのコマンドは、ハードウェア管理専用であり、ハードウェア・セットから属性だけを取得します。ハードウェア・コンポーネントはすべて、一意なハードウェア識別子 (ハードウェア ID または HWID ともいう) を使って識別されます。

次のコマンドは、ローカル・システムにある全ハードウェア・コンポーネントの属性をすべて取得し、ファイルに出力して、情報を検索できるようにします。

```
# /sbin/hwmgrr get attribute > sysattr.txt
```

しかし、3.4.3 項で示したように、照会するコンポーネント・カテゴリがわかっている場合は、特定のカテゴリに的を絞って照会することもできます。

ハードウェア・コンポーネントのカテゴリを指定して属性を照会すると、役に立つ情報が得られます。たとえば、何らかの理由でネットワークが正常に動作しているかどうかははっきりしない場合があったとします。また、システムにインストールされているネットワーク・アダプタの種類やその構成もわからないかもしれません。次の例のように get attribute オプションを使用すれば、ネットワーク・アダプタの状態を判断できます。

```
# /sbin/hwmgrr get attribute -category network
203:
  name = ln0
  category = network
  sub_category = Ethernet
  model = DE422
  hardware_rev =
  firmware_rev =
  MAC_address = 08-00-2B-3E-08-09
  MTU_size = 1500
  media_speed = 10
  media_selection = Selected by Jumpers/Switches
```

```
media_type =
loopback_mode = 0
promiscuous_mode = 0
full_duplex = 0
multicast_address_list = CF-00-00-00-00-00 \
    01-00-5E-00-00-01
interface_number = 1
```

この出力には、次のような情報が含まれています。

- 数値 203 は、このイーサネット・アダプタの HWID です。
- HWID の下に続くフィールドと値は、それぞれ属性名とその現在の値です。ドライバが値を初期化していない場合、値の部分が空白になっていることもあります。この情報から、システムには DE422 というモデル番号のイーサネット・アダプタがあり、そのコンポーネント名は `ln0` であることがわかります。
- この情報を基に次のように `ifconfig` コマンドを用いて、このネットワーク・アダプタの状態を調べることができます。

```
# ifconfig ln0
ln0:flags=c62 inet XX.XXX.XXX.XX netmask ffffffff00 \
broadcast XX.XXX.XX.XXX ipmtu 1500
```

コンポーネント属性の値を変更して、コンポーネント情報やシステムにおける動作を変更できる場合もあります。属性の設定については、3.4.5 項で説明しています。どの属性が設定できるかを調べるには、次の例のように `get` オプションを使ってすべての属性を取得し、`grep` コマンドで `(settable)` というキーワードを探します。

```
# /sbin/hwmgrr get attribute | grep settable

device_starvation_time = 25 (settable)
device_starvation_time = 0 (settable)
device_starvation_time = 25 (settable)
device_starvation_time = 25 (settable)
device_starvation_time = 25 (settable)
device_starvation_time = 25 (settable)
```

この出力は、このシステムに設定可能な属性が 1 つあり、それが `device_starvation_time` であることを示しています。この名前がわかれば、次のようにして、その属性をサポートするコンポーネントのリストを取得できます。

```
# /sbin/hwmgrr get attribute -a device_starvation_time
23:
device_starvation_time = 25 (settable)
```



```

24:
    device_starvation_time = 0 (settable)
25:
    device_starvation_time = 25 (settable)
31:
    device_starvation_time = 25 (settable)
34:
    device_starvation_time = 25 (settable)
35:
    device_starvation_time = 25 (settable)

```

このコマンドの出力は、`device_starvation_time` 属性をサポートしているコンポーネントの HWID を示しています。階層の出力で HWID を読むことで、この属性が SCSI ディスクでサポートされているのかがわかります。

ファイバ・チャネルのリンク速度を調べるには、次のようにしてリンク速度を照会できます。

```

# /sbin/hwmmgr get attribute -a link_speed
656:
    link_speed = 1Ghz

```

アダプタが複数ある場合、このコマンドはすべてのアダプタの属性値を表示します。`link_speed` 属性は、次の値をサポートします。

- 速度を示す値。1GHz または 2GHz。
- 不適用を示す文字列 N/A。アダプタのファームウェアが `link_speed` 属性をサポートしていないことを示します。

`set attribute` オプションと `get category` オプションも参照してください。

3.4.5 コンポーネント属性の設定

`set attribute` コマンド・オプションを使用すると、設定可能な属性の値を設定 (または構成) できます。ただし、すべてのコンポーネント属性を設定できるとは限りません。`get attribute` コマンド・オプションを使用したとき、設定が可能な属性については、属性値の隣に (settable) というラベルが表示されます。そのような属性を見つける方法を、3.4.4 項で説明しています。

3.4.4 項で述べたように、`device_starvation_time` は SCSI ディスクがサポートする設定可能な属性の例です。この属性は、SCSI バスの障害 (データが転送されない) によってコンポーネントへ到達できないという判断をディスク・ドライバが下すまでの経過時間を制御します。

device_starvation_time に設定されている時間が経過してもドライバがコンポーネントの存在を認識できないと、ドライバはバイナリ・エラー・ログにエラー・イベントを記録します。

次のコマンドを使用することにより、HWID が 24 であるコンポーネントの device_starvation_time 属性の値を変更するとともに、その新しい値を確認できます。

```
#/sbin/hwmmgr set attribute -id 24 \  
-a device_starvation_time=60  
# /sbin/hwmmgr get attribute -id 24 \  
-a device_starvation_time  
24:  
device_starvation_time = 60 (settable)
```

ただし、この操作では、属性の saved 値は変更できません。属性にはすべて、current 値、saved 値、および default 値という 3 種類の値があります。default 値は定数値であり、変更することはできません。属性の値を設定していなければ、default (省略) 値が適用されます。saved 値を設定すると、ブートにまたがって値が維持されます。この値は、default 値を永続的に指定変更した値であると考えることができます。

current 値として設定した値は、ブートにまたがって維持されません。この値は、属性の一時的な値であると考えることができます。システムをリブートすると、属性の値は saved 値に戻ります (saved 値がある場合)。saved 値がなければ、属性の値は default 値に戻ります。属性の値を設定するたびに、属性の current 値が変更されます。次の例に、属性の saved 値を調べる方法と設定する方法を示します。

```
# /sbin/hwmmgr get attribute saved -id 24 \  
-a device_starvation_time  
24:  
saved device_starvation_time = 0 (settable)  
  
# /sbin/hwmmgr get attribute saved -id 24 \  
-a device_starvation_time=60  
saved device_starvation_time = 60 (settable)  
# /sbin/hwmmgr get attribute saved -id 24 \  
-a device_starvation_time  
24:  
saved device_starvation_time = 60 (settable)
```

get attribute および get category コマンド・オプションを参照してください。

3.4.6 クラスタの表示

クラスタで作業している場合、クラスタに属する特定のホストに的を絞ってハードウェア管理コマンドを実行しなければならないことがよくあります。view cluster コマンド・オプションを使えば、クラスタに属するホストの詳細情報を取得できます。次の例に、一般的なクラスタを示します。

```
# /sbin/hwmmgr view cluster
Member ID      State      Member HostName
-----
1              UP        ernie.zok.paq.com (localhost)
2              UP        bert.zok.paq.com
3              DOWN      bigbird.zok.paq.com
```

また、このオプションを使って、hwmmgr コマンドがクラスタのすべてのメンバとそれらの現在の状態を認識していることを確認できます。

上記の例は、クラスタにメンバが3つあり、その1つ (bigbird) が現在ダウンしていることを示しています。(localhost) のマークは、hwmmgr が現在クラスタ・メンバ ernie で動作していることを示しています。-cluster オプションを指定して実行した hwmmgr コマンドはすべて、メンバ bert と ernie に送られますが、bigbird には、システムが使用できないため、送られません。また、-member bigbird オプションを指定して実行した hwmmgr コマンドもすべて失敗します。そのホストのクラスタ・メンバ状態が DOWN になっているためです。

view cluster コマンド・オプションは、システムがクラスタのメンバである場合にのみ有効です。シングル・システムで実行すると、エラー・メッセージが表示されます。clu_get_info コマンドを参照してください。クラスタ・システムについては、TruCluster Server のマニュアルを参照してください。

3.4.7 デバイスの表示

hwmmgr コマンドに view devices オプションを指定して使用すると、デバイス特殊ファイル名 (/dev/disk/dsk34 など) を持つコンポーネントをすべて表示できます。ハードウェア・マネージャは、dev_base_name 属性を持つハードウェア・コンポーネントをアクセス可能なデバイスと想定します (デバイス属性の取得方法については 3.4.4 項を参照)。

view devices オプションにより、システムのハードウェア管理に現在登録されているコンポーネントを調べることができます。また、このオプション

で得た情報を基に、デバイス特殊ファイル名を通してコンポーネントにアクセスできます。たとえば、CD-ROM をリーダに挿入したとき、CD-ROM リーダを `/dev/disk/cdrom0` としてマウントすべかどうか、この出力によってわかります。登録されているデバイスの HWID を調べる場合にも、`view devices` オプションは便利です。デバイスの HWID がわかっていれば、他の `hwmgr` コマンドのオプションを用いてそのデバイスの属性を照会したり、そのデバイスに対してその他の操作を行うことができます。

このコマンドの一般的な出力を、次の例に示します。

```
# /sbin/hwmgr view dev
```

HWID:	DSF Name	Mfg	Model	Location
3:	/dev/kevm			
22:	/dev/disk/dsk0c	DEC	RZ26	bus-0-targ-3-LUN-0
23:	/dev/disk/cdrom0c	DEC	RRD42	bus-0-targ-4-LUN-0
24:	/dev/disk/dsk1c	DEC	RZ26L	bus-1-targ-2-LUN-0
25:	/dev/disk/dsk2c	DEC	RZ26L	bus-1-targ-4-LUN-0
29:	/dev/ntape/tape0	DEC	TLZ06	bus-1-targ-6-LUN-0
35:	/dev/disk/dsk8c	COMPAQ	RZ1CF-CF	bus-2-targ-12-LUN-0

出力には、ローカル・システム上の `dev_base_name` 属性を持つハードウェア・コンポーネントがすべて表示されています。ハードウェア・マネージャは、`dev_base_name` を展開して、デバイス特殊ファイルまでのフル・パス (`/dev/ntape/tape0` など) を生成しようとします。必ずパーティション `c` のデバイス特殊ファイルまでのパスが使用されます。パーティション `c` は、テープの場合を除き、デバイスの総容量を表します。デバイス特殊ファイル名とその機能については、第 1 章を参照してください。

クラスタで作業している場合は、次のように `-cluster` オプションを使って、クラスタ全体にわたって、ハードウェア管理に登録されているすべてのコンポーネントが表示できます。

```
# /sbin/hwmgr view devices -cluster
```

HWID:	DSF Name	Model	Location	Hostname
20:	/dev/disk/floppy0c	3.5in	fdi0-unit-0	tril7e
34:	/dev/disk/cdrom0c	RRD46	bus-0-targ-5-LUN-0	tril7e
35:	/dev/disk/dsk0c	HSG80	bus-4-targ-1-LUN-1	tril7d
35:	/dev/disk/dsk0c	HSG80	bus-6-targ-1-LUN-1	tril7e
36:	/dev/disk/dsk1c	RZ26N	bus-1-targ-0-LUN-0	tril7e
37:	/dev/disk/dsk2c	RZ26N	bus-1-targ-1-LUN-0	tril7e
38:	/dev/disk/dsk3c	RZ26N	bus-1-targ-2-LUN-0	tril7e
39:	/dev/disk/dsk4c	RZ26N	bus-1-targ-3-LUN-0	tril7e
40:	/dev/disk/dsk5c	RZ26N	bus-1-targ-4-LUN-0	tril7e

3-14 hwmgr を用いたハードウェア管理

```

41:      /dev/disk/dsk6c  RZ26N  bus-1-targ-5-LUN-0  tr17e
42:      /dev/disk/dsk7c  RZ26N  bus-1-targ-6-LUN-0  tr17e
43:      /dev/disk/dsk8c  HSZ40  bus-3-targ-2-LUN-0  tr17d
43:      /dev/disk/dsk8c  HSZ40  bus-3-targ-2-LUN-0  tr17e
44:      /dev/disk/dsk9c  HSZ40  bus-3-targ-2-LUN-1  tr17d
44:      /dev/disk/dsk9c  HSZ40  bus-3-targ-2-LUN-1  tr17e
45:      /dev/disk/dsk10c HSZ40  bus-3-targ-2-LUN-2  tr17d
45:      /dev/disk/dsk10c HSZ40  bus-3-targ-2-LUN-2  tr17e

```

HWID が 45: であるディスクのように、複数回現れるデバイスがあります。これらは、2 つのクラスタ・メンバが共用しているバス上にあるコンポーネントです。ハードウェア・マネージャは、各クラスタ・メンバでコンポーネント・エントリを見つけるたびにそれを表示します。

次の hwmgr コマンド・オプションを参照してください。show scsi , show components , get attributes

3.4.8 トランザクションの表示

ハードウェア管理操作は、クラスタ内で同期をとる必要があるトランザクションです。view transaction コマンド・オプションは、システムをブートしてから行われたハードウェア管理トランザクションの状態をすべて表示します。このオプションを使って、失敗したハードウェア管理トランザクションを見つけることができます。

-cluster や -member オプションを指定していなければ、このコマンドは、ローカル・ホスト (コマンドを入力したシステム) で処理または開始されたトランザクションの状態を表示します。view transaction コマンド・オプションは、主にクラスタ内のハードウェア管理に関する問題のデバッグに使用するものなので、このコマンドを使うようなことはあまりありません。通常、このコマンドの出力は次のようになります。

```

# /sbin/hwmgr view transactions
hardware management transaction status
-----
there is no active transaction on this system
the last transaction initiated from this system was:
transaction = modify cluster database
proposal    = 3834
sequence    = 0
status      = 0
the last transaction processed by this system was:
transaction = modify cluster database
proposal    = 3834
sequence    = 0
status      = 0

```

proposal		last status	success	fail
	Modify CDB/ 3838	0	3	0
	Read CDB/ 3834	0	3	0
	No operation/ 3835	0	1	0
	Change name/ 3836	0	0	0
	Change name/ 3837	0	0	0
	Locate HW/ 3832	0	0	0
	Scan HW/ 3801	0	0	0
	Unconfig HW - confirm/ 3933	0	0	0
	Unconfig HW - commit/ 3934	0	0	0
	Delete HW - confirm/ 3925	0	0	0
	Delete HW - commit/ 3926	0	0	0
	Redirect HW - confirm/ 3928	0	0	0
	Redirect HW - commit1/ 3929	0	0	0
	Redirect HW - commit2/ 3930	0	0	0
	Refresh - lock/ 3937	0	0	0

この出力は、最後のトランザクションがクラスタ・データベースの変更であったことを示します。

3.4.9 ユーザ定義 SCSI デバイス名の作成

ほとんどのコンポーネントには、デバイスに対して一意に定まる ID 属性があります。これは SCSI デバイスの `serial_number` または名前の属性として読み取ることができます。次の `hwmgr` コマンドは、HWID が 30 のコンポーネント (SCSI ディスク) についてこれら 2 つの属性を表示させる例です。

```
# /sbin/hwmgr get attribute -id 30 -a serial_number -a name
30:
serial_number = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
name = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
```

この文字列は、システム上の各コンポーネントに対して一意であるため、ワールドワイド識別子 (WWID) といいます。

コンポーネントには、一意の ID を持たないものがあります。オペレーティング・システムは、そのようなコンポーネントに対して ID となる番号を作成します。その際、デバイスの物理的な位置を表す有効なパス・データ `bus/target/LUN` を使用します。デバイスを複数のシステムで共用する場合、コンポーネントにアクセスするシステムはそれぞれが異なるパスでそのデバイスを認識するため、システムごとに異なる一意の WWID を作成します。そのような共用デバイスに対しては、入出力アクセスが並列に発生し、データが破壊される可能性があります。共用デバイスを見つけるには、次のコマンドを使います。

```
# /sbin/hwmmgr show component -cshared
```

```
HWID:  HOSTNAME  FLAGS SERVICE COMPONENT NAME
-----
 40:  joey      -cd-- iomap   SCSI-WWID:04100026:"DEC \
RZ28M    (C) DEC00S846590H7CCX"
 41:  joey      -cd-- iomap   SCSI-WWID:04100026:"DEC \
RZ28L-AS (C) DECJEE019480P2VSN"
 42:  joey      -cd-- iomap   SCSI-WWID:0410003a:"DEC \
RZ28     (C) DECPCB=ZG34142470 ; HDA=000034579643"
 44:  joey      rcd-- iomap   SCSI-WWID:04100026:"DEC \
RZ28M    (C) DEC00S735340H6VSR"
.
.
.
```

TL895 モデルのメディア・チェンジャなど一部のデバイスは、INQUIRY ページ 0x80 または 0x83 をサポートしておらず、一意の WWID をシステムに提供することができません。パス・フェイルオーバまたは共用バス上でのクラスタへのインストールなどの機能をサポートするには、そのようなデバイスをシステムに手作業で追加する必要があります。これは、メディア・チェンジャだけを共用バスに追加するための推奨されている方法です。ディスク、CD-ROM リーダ、テープ・ドライブ、RAID コントローラなど、他の種類のデバイスはシリアル番号など一意の文字列を持っているため、システムが一意の WWID を生成できます。このようなコンポーネントは、WWID が常に同じであり、オペレーティング・システムが常に同じデバイスとして認識するため、共用バス上で使用できます。

hwmmgr コマンドを使って、ユーザ定義による一意の名前を作成し、デバイスを共用する全システムが認識する WWID を作成することができます。コンポーネントに共通の WWID があるため、デバイス特殊ファイル名が何通りにもならず、入出力が同時に発生することを防止できます。

ユーザ定義名を作成する手順は次のとおりです。

- 割り当てる名前を選択します。この名前をデバイスにアクセスする全システムの間で一意に定まるものにします。上の例に示した WWID ほど長く複雑でなくてもかまいませんが、名前を変更したコンポーネントを認識でき、他のコンポーネントとも区別できるだけの情報が入る長さにしてください。
- どのコンポーネントにこの名前を与えるかを決定します。名前を変更すると、そのコンポーネントはどのシステムでも同じコンポーネント

として認識されます。もちろん、システムを更新してコンポーネントが認識されるようにしなければなりません。

- コンポーネントを共用する各システムは、その文字列を用いて新しい WWID を作成し、その後はシステムへの登録にこの新しい WWID を用います。内部的には、コンポーネントは依然として、省略時の WWID (存在していれば) を用いて追跡されます。しかし、外部的な表示はユーザ定義名に基づく新しい WWID で行われます。クラスタの場合には、デバイスにアクセスするすべてのクラスタ・メンバで `edit scsi` コマンド・オプションを実行しなければなりません。

注意

デバイスにアクセス可能なすべてのクラスタ・システムを更新しなければなりません。

次の例に、ユーザ定義名を割り当てる方法を示します。 `edit scsi` コマンド・オプションは、一意の WWID を持たないデバイスの場合にだけお勧めしますが、例ではわかりやすいようにディスクを対象にしています。

```
# /sbin/hwmmgr show scsi
```

SCSI			DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICEID	HOST	TYPE	SUBTYPE	OWNER	PATH	FILE	VALID
	ID	NAME						PATH
22:	0	ftwod	disk	none	0	1	dsk0	[0/3/0]
23:	1	ftwod	cdrom	none	0	1	cdrom0	[0/4/0]
24:	2	ftwod	disk	none	0	1	dsk1	[1/2/0]
25:	3	ftwod	disk	none	2	1	dsk2	[2/4/0]

このコマンドは、システム上にある SCSI デバイスを表示します。この表示からはわかりませんが、このシステムには共用バスがあり、ハードウェア・コンポーネントの 24 と 25 が実際には同じデバイスであるとしています。このコンポーネントの WWID は、bus/target/LUN のアドレス情報を用いて作成されています。bus/target/LUN アドレスが異なっているので、このコンポーネントは別々のデバイスとして表示されます。オペレーティング・システムが 2 つの異なるデバイス特殊ファイル (`/dev/disk/dsk1` と `/dev/disk/dsk2`) をこのディスクへのアクセスに使用できるため、データが破壊されるおそれがあります。

次のコマンド例に、デバイスの名前を変更する方法と、名前を変更した後どのように表示されるかを示します。

```
# /sbin/hwmmgr edit scsi -did 2 -uwwid "this is a test"
hwmmgr:Operation completed successfully.

# /sbin/hwmmgr show scsi -did 2 -full

      SCSI
HWID: DEVICEID HOSTNAME  TYPE    DEVICE  DEVICE  DRIVER NUM  DEVICE FIRST
-----
24:  2              ftwod    disk    none    0        1    dsk1    [1/2/0]

WWID:0910003c:"DEC      (C) DECZG41400123ZG41800340:d01t00002100000"
WWID:ff10000e:"this is a test"

      BUS  TARGET  LUN  PATH STATE
-----
      1      2      0    valid
```

この操作をもう一方のコンポーネント、つまりアドレス 2/4/0 のコンポーネントにも実行して、同じ名前を割り当てます。この操作を行うと、ハードウェア管理機能はユーザ定義名を用いてコンポーネントを調べ、それが同じデバイスへの代替パス名であると認識します。

```
# /sbin/hwmmgr edit scsi -did 3 -uwwid "this is a test"
hwmmgr:Operation completed successfully.

# /sbin/hwmmgr show scsi -did 3 -full

      SCSI
HWID: DEVICEID HOSTNAME  TYPE    DEVICE  DEVICE  DRIVER NUM  DEVICE FIRST
-----
25:  3              ftwod    disk    none    0        1    dsk1    [2/4/0]

WWID:0910003c:"DEC      (C) DECZG41400123ZG41800340:d02t00004100000"
WWID:ff10000e:"this is a test"

      BUS  TARGET  LUN  PATH STATE
-----
      2      4      0    valid
```

これで、両方のデバイスが同じデバイス特殊ファイル名 (/dev/disk/dsk1) を使うようになります。2 つの異なるデバイス特殊ファイルを通じて同じディスクにアクセスすることで生じるデータ破壊の危険がこれではなくなります。

3.4.10 SCSI デバイスの削除

SCSI デバイスをシステムから取り外したい場合があります。エラーが続いてログに記録されたために交換が必要になった場合などがそうです。SCSI コンポーネントをクラスタにあるすべてのハードウェア管理データベースから削除するには、`delete scsi` コマンド・オプションを使用します。このオプションは、カーネルから対象コンポーネントの登録を解除し、そのデバイ

スに対する永続データベース・エントリをすべて削除した後、該当するデバイス特殊ファイルをすべて削除します。SCSI コンポーネントを削除すると、そこにはアクセスできなくなり、対応するデバイス特殊ファイルが /dev サブディレクトリから削除されます。現在オープンされている SCSI コンポーネントは、削除できません。デバイスへの I/O 接続 (マウントなど) はすべて外す必要があります。

SCSI コンポーネントをシステムから取り外した後で、そのコンポーネントに関する情報をシステムに残したくなければ、その SCSI コンポーネントを削除する必要があります。また、ハードウェアではなくオペレーティング・システムに問題があるために、SCSI コンポーネントを削除したいことがあります。たとえば、コンポーネントが正常に動作しているのにもかかわらず、何らかの理由でデバイス特殊ファイルによるアクセスが行えないということがあります。このような場合、そのコンポーネントをいったん削除した後、scan scsi コマンド・オプションを用いてそのコンポーネントを検索して登録できます。

SCSI コンポーネントを交換する (または元のコンポーネントに戻す) には、scan scsi コマンド・オプションを用いてそのコンポーネントを再度検索します。ただし、コンポーネントを削除し、scan 操作を実行してからそのコンポーネントをオンラインに戻しても、常に同じデバイス特殊ファイルを持つわけではありません。コンポーネントを交換して、オリジナルとまったく同じ複製として組み込む場合は、3.4.12 項で述べる操作も実行する必要があります。コンポーネントがスキャンの際にアクティブに応答しなければ、scan 操作を行ってもコンポーネントを検出できないことがあります。

このオプションは、SCSI デバイス ID の -did を受け付けます。この ID は、HWID と同じものではありません。次の例に、SCSI データベースを調べて SCSI デバイスを削除する方法を示します。

```
# /sbin/hwmmgr show scsi
```

	SCSI		DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICEID	HOST-	TYPE	SUBTYPE	OWNER	PATH	FILE	VALID
		NAME						PATH
23:	0	bert	disk	none	2	1	dsk0	[0/3/0]
24:	1	bert	cdrom	none	0	1	cdrom0	[0/4/0]
25:	2	bert	disk	none	0	1	dsk1	[1/2/0]
30:	4	bert	tape	none	0	1	tape2	[1/6/0]
31:	3	bert	disk	none	0	1	dsk4	[1/4/0]
34:	5	bert	disk	none	0	1	dsk7	[2/5/0]
35:	6	bert	disk	none	0	1	dsk8	

この例では、ID が 23 であるコンポーネントの DRIVER OWNER フィールドはゼロではありません。これは、デバイスがドライバによってオープンされていることを示します。DRIVER OWNER フィールドがゼロ以外の値の場合、コンポーネントがドライバによってオープンされ、使用されていることを意味します。したがって、SCSI コンポーネントの 23 は、現在使用されているので削除できません。

しかし、ID が 35 のコンポーネントはドライバによってオープンされておらず、FIRST VALID PATH フィールドに有効なパスが表示されていません。コンポーネントは現在アクセス不可能であり、安全に削除できます。デバイスを削除するとき、デバイス特殊ファイルの /dev/disk/dsk8* と /dev/rdisk/dsk8* も削除できます。

SCSI デバイスを削除する場合は、delete オプションに SCSI DEVICEID という値を指定してから、次のようにして SCSI データベースを調べます。

```
# /sbin/hwmgrr delete scsi -did 6
hwmgrr: The delete operation was successful.
# /sbin/hwmgrr show scsi
```

	SCSI		DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICE	HOSTNAME	TYPE	SUBTYPE	OWNER	PATH	FILE	VALID
	ID							PATH

23:	0	bert	disk	none	2	1	dsk0	[0/3/0]
24:	1	bert	cdrom	none	0	1	cdrom0	[0/4/0]
25:	2	bert	disk	none	0	1	dsk1	[1/2/0]
30:	4	bert	tape	none	0	1	tape2	[1/6/0]
31:	3	bert	disk	none	0	1	dsk4	[1/4/0]
34:	5	bert	disk	none	0	1	dsk7	[2/5/0]

コンポーネント /dev/disk/dsk8 は正常に削除されています。

3.4.11 RAID アレイのディスクの構成設定

RAID アレイを再構成するとき、新しいブロック・ゼロは、以前のブロック・ゼロと同じブロックである可能性があります。そのため、ディスクラベルがディスクの範囲を越えていても有効なラベルと見なしてしまうアプリケーションによって問題が生じることがあります。スキャンを実行すると、システムは新しいユニットを dskNN として認識します。ディスクを使い始める前に、次のコマンドを実行して不適切なラベルをゼロに初期化します。

```
# /sbin/disklabel -z dskNN
```

RAID アレイに新しいユニットを作成した場合や、RAID アレイのユニットを構成する 1 つ以上のディスクを移動してホスト・バス・アダプタに直結したときは、このコマンドを実行します。

続いて、新しい省略時のラベルを作成するか、プロト・ファイルに基づいて事前構成されたラベルを適用するには、次のように `disklabel` コマンドを実行します。

```
# /sbin/disklabel -rwn dskNN
# /sbin/disklabel -Rr dskNN PROTOFILE
```

3.4.12 故障した SCSI ディスクの交換

SCSI ディスクが故障した場合、そのデバイスを交換して、交換した新しいディスクに故障デバイスのハードウェア特性 (そのデバイス特殊ファイルの所有者など) を継承させることができます。 `redirect` コマンド・オプションを使用すれば、そのような特性を割り当てることができます。たとえば、HSZ (RAID) キャビネットのディスクが故障したときは、その故障ディスクをホットスワップによって交換し、次に `redirect` コマンド・オプションを用いて新しいディスクを故障ディスクの代わりとしてオンラインにします。

故障したディスクを AdvFS または LSM などのアプリケーションが管理している場合は、この手順を単独では利用しないでください。管理されたディスクを交換する前に、ディスク管理アプリケーションを適切な状態にするか、管理アプリケーションからディスクを削除します。『*Logical Storage Manager*』および『*AdvFS 管理ガイド*』などの、適切なマニュアルを参照してください。

注意

`redirect` 操作を正常に実行させるには、交換するディスクが元のディスクと同じタイプのものでなければなりません。

次の例は、`redirect` オプションの使い方を示します。

```
# /sbin/hwmgr show scsi
```

	SCSI	DEVICE	DEVICE	DRIVER	NUM	DEVICE	FIRST
HWID:	DEVICE-	HOST-	TYPE	SUB-	OWNER	PATH	VALID
	ID	NAME	TYPE	TYPE		FILE	PATH
23:	0	fwod	disk	none	2	1	dsk0 [0/3/0]
24:	1	fwod	cdrom	none	0	1	cdrom0 [0/4/0]

```

25:  2      fwod  disk  none  0      1      dsk1    [1/2/0]
30:  4      fwod  tape  none  0      1      tape2   [1/6/0]
31:  3      fwod  disk  none  0      1      dsk4     [1/6/0]
37:  5      fwod  disk  none  0      1      dsk10    [2/5/0]

```

この出力から、HWID 31 の SCSI ディスクが故障していることがわかります。このコンポーネントに有効なパスがありません。この故障ディスクを、デバイス特殊ファイル名が `/dev/disk/dsk4` で `dev_t` 情報が同じである新しいディスクと交換するには、次の手順を使います。

1. ハードウェア・マニュアルの説明に従ってコンポーネントをインストールします。
2. 次のコマンドを用いて新しいデバイスを検索します。

```
# /sbin/hwmgrr scan scsi
```

このコマンドは、SCSI サブシステムを調べて、新しく見つかったデバイスを登録します。`show scsi` コマンドをもう一度実行して、交換した新しいディスクの SCSI デバイス ID (`did`) を調べます。

3. 次のコマンドを用いて、故障ディスクのコンポーネント特性を交換ディスクに割り当て直します。この例では、新しいディスクに割り当てられた SCSI デバイスの ID (`did`) が 36 であるとしています。

```
# /sbin/hwmgrr redirect scsi -src 3 -dest 36
```

注意

ディスクに対する `redirect` オプションの実行が失敗した場合には、3.4.13 項で説明している代替手順を使います。

3.4.13 故障した SCSI テープ・ドライブ (またはハード・ディスク) の交換

SCSI テープが故障した場合、そのデバイスを交換して、交換した新しいドライブに故障デバイスのハードウェア特性 (そのデバイス特殊ファイルの所有者など) を継承させることができます。SCSI テープ・ドライブまたはディスクによっては、3.4.12 項で説明されている `redirect` コマンド・オプションが正しく動作しないことがあります。次の手順に従って、操作が正しく完了するようにします。この手順では、次の前提が適用されます。

- 交換用のテープ・ドライブは、元のドライブと同じモデルでなければならない。
- SCSI テープ・ドライブが動作中に故障した場合、デバイス・ドライバがデバイスをクローズできないことがある。手順を実行する前に、デバイスをリセットするためにシステムをリブートします。
- デバイスを追加または交換するためにシステムをシャットダウンする必要がある場合 (内蔵テープ・ドライブなどの場合) には、代替の手順を参照すること。

1. 次のコマンドを用いて、故障したコンポーネントを確認します。

```
# /sbin/hwmgrr show scsi
      SCSI          DEVICE DEVICE DRIVER NUM  DEVICE  FIRST
HWID: DEVICE- HOST-  TYPE   SUB-   OWNER  PATH FILE   VALID
      ID      NAME      TYPE
-----
31:   5          rocym  tape  none    2     1   tape0  [    ]
```

この出力から、HWID 31 の SCSI テープ・ドライブが故障していることがわかります。このコンポーネントに有効なパスがありません。この故障テープ・ドライブを、デバイス特殊ファイル名が `/dev/tape/tape2` で `dev_t` 情報が同じである新しいテープ・ドライブと交換するには、次の手順を使います。

2. ハードウェア・マニュアルの説明に従ってコンポーネントをインストールします。
3. 次のコマンドを使って、新しいデバイスをスキャンします。

```
# /sbin/hwmgrr scan scsi | grep tape
```

4. 次のコマンドを使って、故障したデバイスと交換デバイスの SCSI データベース・エントリを検索します。

```
# /sbin/hwmgrr show scsi | grep tape
```

次の出力例のように、交換デバイスを示す新しいエントリと、元の (故障した) デバイスを示す不完全なエントリが表示されます。

```
      SCSI          DEVICE DEVICE DRIVER NUM  DEVICE  FIRST
HWID: DEVICEID HOSTNAME TYPE   SUBTYPE OWNER  PATH FILE   VALID
      PATH
-----
31:   5          rocym  tape  none    2     1   tape0  [    ]
35:   7          rocym  tape  none    0     1   tape5  [0/7/0]
```

5. `hwmgr` コマンドに故障したデバイスのハードウェア識別子 (HWID) を指定して、そのデータベース・エントリを削除します。

```
# /sbin/hwmgr delete component -id 31
```

6. 次のコマンドを使って、交換デバイスの名前を付け替え、故障したデバイスのデバイス特殊ファイルを移します。

```
# /sbin/dsfmgr -m tape5 tape0
tape5=>tape0  tape5_d0=>tape0_d0  tape5_d1=>tape0_d1
tape5_d2=>tape0_d2  tape5_d3=>tape0_d3  tape5_d4=>tape0_d4
tape5_d5=>tape0c
```

ホットスワップできないデバイスを交換する場合には、システムをシャットダウンする必要があります。その場合、この手順の最初の 2 つの手順を次のように変更します。

1. 交換デバイスを挿入し、システムをブートします。ブート時に SCSI のスキャンが自動的に実行され、コンソールに交換デバイスが検出され登録されたことを示すメッセージが表示されます。
2. システムがシングル・ユーザ・モードになったら、次のコマンドを実行して交換デバイスが検出されたことを確認します。

```
# /sbin/hwmgr show scsi | grep device_type
```

`device_type` には、`disk` や `tape` など、デバイスのタイプを指定します。

元の手順の手順 3 に進みます。

3.4.14 `hwmgr` によるクラスタ・メンバのブート・ディスクの交換

シングル・システムでは、故障したディスクを交換する手順の中で、`hwmgr` コマンドに `redirect` オプションを指定して使用できます。故障したディスクを交換するときは、`redirect` オプションを使って、故障したコンポーネントから新しい交換デバイスに入出力をリダイレクトします。このオプションは、デバイス特殊ファイル名、クラスタの `dev_t` の値、ローカルの `dev_t` の値、論理 ID、および HWID をリダイレクトします。

`redirect` オプションに指定できるのは一意のデバイス識別子 (`did`) だけです。クラスタの場合、デバイス識別子が一意であるとは限らず、次に示す例のように、コマンドがエラーになる可能性があります。

```
# /sbin/hwmgr redirect scsi -src source_did -dest target_did
# "Error (95) Cannot start operation."
```

リダイレクト操作を正常に行うには、クラスタ上の各メンバにハードウェア識別子の両方が存在するか、どちらも存在しない必要があります。次の手順に従って、`redirect` 操作が正しく行われるようにします。

1. 交換元と交換先のコンポーネントがともに存在することを確認します。クラスタの各メンバにおいて、次のコマンドを使います。

```
# /sbin/hwmgrr show scsi -did device_identifier
      SCSI          DEVICE DEVICE DRIVER NUM  DEVICE FIRST
HWID: DEVICEID HOST  TYPE   SUBTYPE OWNER  PATH FILE   VALID PATH
32:   DID          rymoc disk   none    2    1   dsk1   [0/1/0]
```

2. 交換元コンポーネントが別のクラスタ・メンバ上に存在するが、交換先がそこに存在しない場合にだけ、この手順に従います。

交換先コンポーネントをこれらのクラスタ・メンバ上で次のように構成します。

```
# /sbin/hwmgrr scan scsi
```

注意

バスのスキャンは非同期で実行されます。システム・プロンプトにすぐに戻りますが、これはスキャンが完了したことを示すわけではありません。多数のデバイスがあるシステムでは、スキャンの完了まで数分かかることがあります。

3. 交換先コンポーネントがシステムの別のメンバ上に存在するが、交換元コンポーネントがそこに存在しない場合にだけ、この手順を実行します。

交換先コンポーネントをこれらのクラスタ・メンバから次のように削除します。

```
# /sbin/hwmgrr delete scsi did
```

4. これで、`redirect` オプションを使って、入出力を交換先のドライブに向けることができます。

3.4.15 name サブシステムの永続データベースの表示

`name` 永続データベースには、システムのハードウェア・トポロジに関する情報が保存されています。このデータはカーネルによって維持され、コントローラやバスに関するデータが含まれます。`show name` コマンド・オプションを使って永続データを表示します。永続データは、他の `hwmgrr` コマンドを用いて操作できます。

次の例は、小規模なシステムで `show name` コマンド・オプションを実行した場合の典型的な出力を示します。

```
# /sbin/hwmgr show name -member ychain
```

HWID:	NAME	HOSTNAME	PERSIST TYPE	PERSIST AT
13:	isp0	ychain	BUS	pci0 slot 5
4:	pci0	ychain	BUS	nexus
14:	scsi0	ychain	CONTROLLER	isp0 slot 0
29:	tu0	ychain	CONTROLLER	pci0 slot 11

出力には、次の情報が含まれています。

- **HWID:** – コンポーネントに対して一意に定まるハードウェア ID。この情報は、`view hierarchy` コマンド・オプションで調べることもできます。
- **NAME** – コンポーネント名とインスタンス番号。PCI (Personal Computer Interconnect) バス 0 に対する `pci0` など。PCI バスごとにインスタンス番号が異なります。
- **HOSTNAME** – コマンドを実行したホスト。クラスタで作業している場合は、コマンドを実行するクラスタ名を指定できます。
- **PERSIST TYPE** – ハードウェア・コンポーネントの種類。通常はバスかコントローラです。
- **PERSIST AT** – コンポーネントの論理アドレス。ハードウェア内の物理位置に対応しています。たとえば、SCSI コントローラ `scsi0` は、バス `isp0` の `slot 0` にあります。

3.4.16 name 永続データベースからのコンポーネントの抹消と登録解除

`name` サブシステムを操作して、永続データベースからコンポーネントを削除することができます。`hwmgr` コマンドには、登録解除の方法が 2 通りあります。

- **remove** – 永続データベースからエントリを登録解除する。動作中のシステムには影響しませんが、リブートするとそのコンポーネントは認識されなくなります。
- **delete** – 永続データベースからエントリを登録解除するとともに、動作中のシステムから抹消する。このコマンドはコンポーネントの登録と構成を解除し、すべてのハードウェア管理データベースから抹消します。

次の例は、小規模なシステムで `show name` コマンド・オプションを実行した場合の典型的な出力を示します。`name` には、3.4.15 項で説明した `show name` コマンド・オプションの出力で表示されるコンポーネント名を指定します。

```
# /sbin/hwmmgr show name
HWID:  NAME      HOSTNAME  PERSIST TYPE      PERSIST AT
-----
33:  aha0        fegin    BUS          eisa0 slot 7
31:  ln0         fegin    CONTROLLER   eisa0 slot 5
8:   pci0        fegin    BUS          ibus0 slot 0
34:  scsi1       fegin    CONTROLLER   aha0 slot 0
17:  scsi0       fegin    CONTROLLER   psiop0 slot 0
15:  tu0         fegin    CONTROLLER   pci0 slot 0
```

この例の出力では、2つの SCSI アダプタが表示されています。`scsi0` を対象にして `remove` 操作を行っても、`scsi1` は `scsi0` になりません。アダプタの位置に関する情報は `aha0s lot 0` のままで、`scsi1` の名前はブートしても変わりません。

`scsi0` を削除して `scsi1` の名前を変更するには、次のコマンドを使います。

```
# /sbin/hwmmgr remove name -entry scsi0
# /sbin/hwmmgr edit name -entry scsi1 -parent_num 0
```

3.4.17 ハードウェア・データベースの最適化

無効なパスの数がシステムのブート時間に影響する場合があります。データベースの読み取りに余分の時間がかかるからではなく、SCSI サブシステムが無効かどうかにかかわらず各パスをスキャンするからです。このような無効なパスは、ストレージを別のアダプタに移動したり、アダプタを取り外したり交換するなどして、システムの構成に多数の変更を加えたときに発生します。ただし、予想外に多くの無効なパスがシステムにある場合には、構成に問題があるかもしれません。その場合には、`refresh` オプションを使用する前に、技術サポート担当までご相談ください。

無効なパスを削除するには、次のオプションを使用します。

```
# /sbin/hwmmgr refresh scsi
```

このコマンドは SCSI デバイスへの無効なパスを削除しますが、システムがデバイスへの第 1 パスと見なす無効パスは除きます。

`refresh component` オプションは必ずしも使用する必要はなく、一般的にはブートに何の影響もしませんが、コマンドの出力がわかりやすくなります。ハードウェアの構成を大幅に変えたとき、特にコンポーネントを取り外したり交換したりしたときは、コマンド出力には多数の不要なエントリが表示されるようになります。これら未使用エントリは、次のコマンドを使ってコンポーネント・データベースを表示したときに目に見えます。

```
# /sbin/hwmgrr show component
```

次のコマンドを使って、システムに戻されることのないコンポーネントに対応するデータベース・エントリを削除します。

```
# /sbin/hwmgrr refresh component
```

3.4.18 コンポーネント名の変更

コンポーネント名は、使用されているドライバ・インタフェースと、デバイスのインスタンスに基づいています。たとえば、`tu0` という名前のコンポーネントは、Tulip Ethernet インタフェースをサポートする NIC です (`tu(7)` を参照)。Intel (82558/82559) PCI 10/100 Ethernet NIC は、`name` データベースの中で `eeN` という名前のコンポーネントとして登場します (`ee(7)` を参照)。

コンポーネントは、この物理位置に基づいて永続タイプと永続位置を割り当てられます。この情報は、`name` データベースに格納されます。永続タイプとしては、ネットワーク・コントローラや、一連のスロットを提供する PCI (Peripheral Component Interconnect) バスなどがあります。コンポーネントの永続位置は、PCI バス 1 のスロット 2 や、主バス位置 (`nexus`) など、バス上の特定のスロットの論理アドレスがあります。

次の手順は、使用したい名前をコンポーネントに割り当てる方法を示します。これによって、複数の異なるシステムの間でコンポーネント名に一貫性を持たせて、特定のコンポーネントを指定するスクリプトの保守を簡単にできます。

注意

HP では、可能な限りシステムにコンポーネント名を動的に割り当てさせることを推奨します。環境を更新するときにシステムのカスタム設定を保存することを可能にする代替手順もあります。たとえば、システムのクローニングを行うためのサポートされている方法は、インストレーション・クローニング・ユーティリ

ティです。使用しているオペレーティング・システムのバージョンに対応する『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。コンポーネント名は動的なので、ローカルのスクリプトやプログラムがコンポーネント名に依存しないように作ることを考えてください。そのために、hwmgr および dsfmgr コマンドを使ってコンポーネント名とコンポーネントを利用できるかどうかを簡単に調べることができます。静的な名前を指定しなければ、スクリプトやプログラムの信頼性や移植性ははるかに高くなります。

3.4.18.1 コンポーネントの識別

次の hwmgr コマンドを使って name データベースの内容を表示できます。

```
# /sbin/hwmgr show name
```

HWID:	NAME	HOSTNAME	PERSIST TYPE	PERSIST AT
41:	ata1	rocym	BUS	pci0 slot 205
39:	ata0	rocym	BUS	pci0 slot 105
56:	ee2	rocym	CONTROLLER	pci3 slot 5
19:	ee1	rocym	CONTROLLER	pci2 slot 5
18:	ee0	rocym	CONTROLLER	pci2 slot 4
54:	itpsa1	rocym	BUS	pci3 slot 4

一部が省略されているこの出力例は、システムに3つのNICがあり、ee0、ee1、および ee2 という名前であることを示します。この出力例は、コンポーネント ee1 の物理位置がPCI バス 2、スロット 5 であることなど、ネットワーク・コンポーネントの物理位置も示します。

コンポーネントの名前を ee3、ee4 および ee5 に変えるには、3.4.18.2 項の手順に従います。

3.4.18.2 コンポーネント名の変更

コンポーネント名を変更するには次の手順に従います。

1. 次のコマンドを使って、name データベースからハードウェア永続エントリを削除します。

```
# /sbin/hwmgr remove name -entry ee0
# /sbin/hwmgr remove name -entry ee1
# /sbin/hwmgr remove name -entry ee2
```

名前を変更したいすべてのコンポーネントについて、このコマンドを繰り返します。この操作は、削除する名前を使用しているハードウェア・コンポーネントには影響しません。リブートにまたがった名前の維持にだけ影響します。

注意

`delete` オプションの代わりに `remove` オプションを使うと、コンポーネントの属性設定がすべて維持されます。たとえば、コンポーネント `ee0` について、その `user_name` 属性の値を指定することで、独自にユーザ名を定義できます。ユーザ定義の名前（および他のカスタマイズされた設定）は、`delete` オプションを使うと維持されません。

2. 次のコマンドを使って、システムをシャットダウンしリブートします。

```
# shutdown -r now
```

システムのリブート時に、永続名が再作成されます。新しい名前がハードウェアが検出された順番で割り当てられます。たとえば、`ee0` はハードウェアのスキャン時に最初に見つかった NIC に割り当てられ、`ee1` はその次に見つかったものに、そしてすべてのコンポーネントが検出されるまで以降同様となります。

3. 次のコマンドを使って、各コンポーネントの以前の名前を `name` データベースから削除します。

```
# /sbin/hwmgrr delete name -entry ee0
# /sbin/hwmgrr delete name -entry ee1
# /sbin/hwmgrr delete name -entry ee2
```

3.4.18.3 名前変更の検証

名前の変更が正しく行われたことを次のようにして確認します。

1. 次のコマンドを使って、どのコンポーネントが特定の PCI スロットに対応するかを確認します。ここでは、コマンドを使って `ee*` という条件に当てはまるコンポーネントを検索しています。

```
# /sbin/hwmgrr show name | grep ee
.
.
.
56: ee5      rocy      CONTROLLER      pci3 slot 5
```

```
19: ee4      rocym      CONTROLLER      pci2 slot 5
18: ee3      rocym      CONTROLLER      pci2 slot 4
.
.
.
```

2. 次のコマンドを使って、不正な名前がすべて name データベースから恒久的に削除されたことを確認します。

```
# /sbin/hwmgrr show components | grep ee
.
.
.
18: rocym      r---- none      ee3
19: rocym      r---- none      ee4
56: rocym      r---- none      ee5
.
.
.
```

名前の変更が正しくできていないようであれば、次の表に示すトラブルシューティングの方法を試してみます。

問題	考えられる解決策
hwmgrr コマンドを使用したときにコンポーネントが表示されない。	システムをシャットダウンし、コンソール・コマンドを使って、コンポーネントが構成の一部として表示されることを確認する。コンポーネントが表示されない場合、コンポーネントのテストと検証の手順についてハードウェアのドキュメントを参照する。
コンポーネントは表示されるが、エラーが発生しているように見える。	コンポーネントがコンソールに表示された場合には、システムをリブートし、ブート・メッセージにコンポーネント固有のエラーが表示されないことを確認する。このようなメッセージは、 <code>/var/adm</code> ディレクトリのログ・ファイルに記録されていることもある。

問題	考えられる解決策
システムが正常にリブートするが、ブート後にコンポーネント・エラーが発生する。	<p>Event Viewer (EVM) を使用してバイナリ・イベント (binlogd) を表示するには、次のコマンドを使う。</p> <pre># sysman event_viewer</pre> <p>イベント・ビューアと関連する診断ツールの詳細については、使用するオペレーティング・システムのバージョンに対応する『システム管理ガイド』を参照すること。</p>
正常なリブートと名前変更が行われたにもかかわらず、1 つ以上のコンポーネントの名前が依然として正しくない。	コンポーネントがシステムに接続されており、Hardware Manager から認識できることを確認したら、名前変更の手順を再度実行する。
コンポーネントの名前を変更できない。	まれに、ハードウェア・データベースの内容が壊れることがある。そのような状態のデータベースを編集またはリフレッシュしてはならない。技術サポート担当まで連絡して支援を受けるか、完全インストールを実施してカスタマイズを再度行う。

3.4.19 コンポーネントの再配置 (移動)

Tru64 UNIX では、コンポーネント名は一意になるように作られています。一部のハードウェア・コンポーネントは、その物理位置に基づいて一意の名前が作られます。特に、ほとんどのバスとコントローラは、システム全体で一意の名前を、その物理位置に基づいて作られます。そのようなコンポーネントのシステム内での物理位置を、ある位置から別の位置に変えてシステムをリブートすると、コンポーネントが新しい名前を持つ新しいデバイスとして表示されることがあります。

特定のコンポーネント (ネットワーク・カードなど) を指定するスクリプトやプログラムは、コンポーネントの名前が変わると、それを見つけられなくなることがあります。システム内のコンポーネントを移動するときに、元のコンポーネント名を維持することができます。この節で説明する手順は、コンポーネント名を維持する方法を示します。ここでは移動コンポーネントの例として、メモリ・チャンネル・カードを使います。

3.4.19.1 コンポーネント名の探索

多くのコンポーネント名は、デバイスによって使われるドライバのインタフェースと、デバイスのインスタンスに基づいています。たとえば、`mchan0` という名前のコンポーネントは、インスタンス 0 のメモリ・チャンネル・カードです。コンポーネントは、この物理位置に基づいて永続タイプと永続位置を割り当てられます。この情報は、`name` データベースに格納されます。

次の `hwmgr` コマンドを使って `name` データベースの内容を表示できます。

```
# /sbin/hwmgr show component
HWID:  NAME      HOSTNAME  PERSIST TYPE      PERSIST AT
      62:  ata0      host12    BUS              pci0 slot 15
      45:  emx2      host12    BUS              pci0 slot 1
      228: mchan0     host12    CONTROLLER      pci1 slot 9
      .
      .
      .
```

コンポーネントは、複数のスロットを提供する PCI (Peripheral Component Interconnect) のようなバスにインストールされている場合があります。前記の出力では、`PERSIST AT` フィールドは以下で構成されています。

- `parent_nam` – 親コンポーネントの論理名。PCI バスを示す `pci0` など。
- `parent_num` – 親の場所の物理アドレス。PCI バス 1 の 15 番スロットなど。

特定のコンポーネントの下での 2 次コンポーネントのトポロジを表示するは、`hwmgr view hierarchy` コマンドを次のように使って、コンポーネントのハードウェア識別子を指定します。

```
# /sbin/hwmgr view hierarchy -id 610
HWID:  hardware hierarchy
-----
610:    bus pci2
611:      connection pci2slot0
617:        scsi_adapter itpsa0
618:          scsi_bus scsi0
674:            disk bus-0-targ-5-lun-0 cdrom51
613:      connection pci2slot1
619:        scsi_adapter itpsa1
620:          scsi_bus scsi1
208:            disk bus-1-targ-0-lun-0 dsk1188
209:            disk bus-1-targ-1-lun-0 dsk1189
210:            disk bus-1-targ-2-lun-0 dsk1190
615:      connection pci2slot2
621:        network tu0
```


3.4.19.2 コンポーネントの再配置

コンポーネントは次のようにして再配置します。

1. 次のコマンドを使って、再配置するコンポーネントの現在のバスとスロット位置を調べます。この例では、メモリ・チャンネル・カード (mchan) を指定します。

```
# /sbin/hwmgrr show name | mchan
228: mchan0 host12 CONTROLLER pci1 slot 9
```

2. 次のようにコンポーネントの新しい位置を指定して、ハードウェア name データベースを更新します。

```
# /sbin/hwmgrr edit name -entry mchan0 parent_num 5 slot 8
```

このコマンドは、コンポーネント mchan0 を PCI バス 1 (pci1)、スロット 9 の現在の位置から、PCI バス 5 (pci5)、スロット 8 の新しい位置に再配置します。

注意

使用したい名前と位置のペアが取得できない場合には、hwmgrr remove name コマンドまたは hwmgrr delete entry コマンドを使って、データベース内の既存の名前を削除する必要があるかもしれません。ただし、既存の (移動しない) コンポーネントの必要なエントリを削除しないよう注意してください。

3. 次のようにして、システムをシャットダウンします。

```
# /usr/sbin/shutdown now
```

4. コンポーネントを新しいスロットに物理的に再配置します。
5. システムをリブートし、次のようにしてシングルユーザ・モードにします。

```
>>> boot -flags s
```

6. コンポーネントが正しい位置で検出されたことを確認します。

```
# /sbin/hwmgrr show name | mchan
228: mchan0 host12 CONTROLLER pci5 slot 8
```

7. 次のようにして、システムをマルチユーザ・モードにブートします。

```
# Ctrl/D
```

```
228: mchan0 host12
```

```
CONTROLLER
```

```
pci5 slot 8
```

問題	考えられる解決策
hwmgr コマンドを使用したときにコンポーネントが表示されない。 ほとんどの場合、この問題は新しい位置にコンポーネントを正しくインストールできなかったときに生じる。	システムをシャットダウンし、コンソール (>>>) のコマンドを使って、コンポーネントが構成の一部として表示されることを確認する。表示されたら、再配置の手順を再度実行する。 コンポーネントが表示されない場合、コンポーネントのテストと検証の手順についてハードウェアのドキュメントを参照する。
コンポーネントは表示されるが、エラーが発生しているように見える。	コンポーネントがコンソールに表示された場合には、システムをリブートし、ブート・メッセージにコンポーネント固有のエラーが表示されないことを確認する。このようなメッセージは、 <code>/var/adm</code> ディレクトリのログ・ファイルに記録されていることもある。
システムが正常にリブートするが、ブート後にコンポーネントエラーが発生する。	Event Viewer (EVM) を使用してバイナリ・イベント (binlogd) を表示するには、次のコマンドを使う。 # sysman event_viewer イベント・ビューアと関連する診断ツールの詳細については、使用するオペレーティング・システムのバージョンに対応する『システム管理ガイド』を参照すること。
正常なリブートと再配置が行われたにもかかわらず、1 つ以上のコンポーネントの名前が依然として正しくない。	コンポーネントを新しい位置に正しく挿入したことを確認する。 コンポーネントがシステムに接続されており、Hardware Manager から認識できることを確認したら、再配置の手順を再度実行する。

DDR (動的デバイス認識)

ほとんどの場合、デバイス管理は自動で行われます。システムに追加したデバイスは、第3章で説明したように認識され、マップされ、追加されます。しかし、自動的に検出してシステムに追加できないデバイスを手動で追加しなければならないこともあります。このようなデバイスには型の古いものや、新しいプロトタイプがあります。また、サポートしている標準 (SCSI など) に厳密に準拠していないものもあります。このような場合、この章で説明する `ddr_config` コマンドを用いて、デバイスとそのドライバをカーネル内に手作業で構成しなければなりません。

この章では、次の項目について説明します。

- カーネル再構成の手順の概要 (4.1 節)
- 動的な方法を用いたカーネルの再構成 (4.2 節)
- DDR データベースの手作業による変更 (4.3 節)
- カスタマイズされた CAM (共通アクセスメソッド) データの変換 (4.4 節)
- リモート・ログインを可能にする端末擬似デバイスである擬似端末 (ptys) の作成方法 (4.5 節)

4.1 カーネルの再構成

カーネルの再構成と再構築を行う方法には、次の2種類があります。

- 動的な方法では、`ddr` コマンドを使用してカーネルを再構築することによって、ディスク構成の変更を確定します。この方法では、オペレーティング・システムをシャットダウンする必要はありません。
- 静的な方法では、`MAKEDEV` および `config` コマンドを使用する必要があります。カーネルを再構築してその変更を確定するためには、システムをシャットダウンして再起動する必要もあります。

`dsfmgr` コマンドの代わりに `MAKEDEV` コマンドまたは `mknod` コマンドを使って、デバイス特殊ファイルを作成します。他社製のカーネル・

レイヤード・プロダクトについては、`kmknod` コマンドを使って、デバイス特殊ファイルを作成します。詳細は、リファレンス・ページの `MAKEDEV(8)`、`mknod(8)`、および `kmknod(8)` を参照してください。

ロード可能ドライバについては、`sysconfig` コマンドを使って、デバイス特殊ファイルを作成します。その際、データベース・ファイル `/etc/sysconfigtab` のスタンザ・エントリで指定された情報が用いられます。

4.2 動的メソッドの使用

以降の項では、`ddr_config` コマンドを使用してシステムの DDR (動的デバイス認識) データベースを管理する方法を説明します。これらの項では、DDR の概要を説明した後、`ddr_config` コマンドを用いて行う次のタスクについて説明します。

- DDR データベースへの SCSI デバイスの追加
- カスタマイズした `cam_data.c` ファイルの変換

4.2.1 DDR (動的デバイス認識)

DDR は、SCSI デバイスの特性および操作パラメータを SCSI CAM I/O サブシステムに対して記述するためのフレームワークです。DDR を使用することによって、オペレーティング・システムをリブートせずに、新規の SCSI デバイスや変更した SCSI デバイスを、システムに追加できます。これにより、静的なデバイス認識による方法では避けられないユーザ・サービスやプロセスの中断を防ぐことができます。

DDR は SCSI デバイス認識の静的な方法に取って代わるものです。現在の静的な方法では、`/sys/data/cam_data.c` データ・ファイルを編集して SCSI デバイス情報をカスタマイズし、カーネルを再構成したのち、オペレーティング・システムをシャットダウンしてリブートします。

注意

SCSI デバイス認識の静的な方法は、将来のリリースではサポートされなくなる予定です。

同じシステムで両方の方法を使用することができますが、各方法で記述されているデバイスは、その方法に限定されます (二重に定義することはできません)。

DDR が SCSI デバイスに関して提供する情報は、SCSI ドライバが必要とするものです。ユーザは、新しい SCSI デバイスをシステムに追加するときに、DDR を使用してこの情報を提供することができます。そうでない場合は、`/sys/data/cam_data.c` データ・ファイルと静的構成方法を使用することができます。DDR が提供する情報は、`cam_data.c` ファイルが提供する情報と同じ目的を持っています。SCSI デバイスに関する情報を静的な方法で提供する場合と比較すると、DDR では、デバイス・ドライバまたはサブシステムがオペレーティング・システムに対して提供する情報量を最小にして、デバイス自体が提供する情報または DDR データベースで指定されている省略時の設定で提供される情報量を最大にします。

4.2.1.1 標準に対する準拠

システムに追加するデバイスは、少なくとも *SCSI-2, Small Computer System Interface-2 (X3.131-1994)* に規定されている SCSI-2 標準に準拠していることが推奨されます。デバイスが標準に準拠していない場合、または標準に対して例外が必要な場合には、これらの相違に関する情報を DDR データベースに格納しておきます。デバイスが SCSI-2 標準に準拠している場合には、データベースを変更する必要はありません。このようなデバイスは、システムによって自動的に認識され、`hwmgrr` コマンドを使ってこれらのデバイスを構成することができます。

4.2.1.2 DDR メッセージ

次のリストに、最も一般的な DDR メッセージのカテゴリと、メッセージに対する対応策 (ある場合) を示します。

- ブート処理の間、コンソール・メッセージが表示される。これらのメッセージは、ほとんどの場合、カーネルが DDR データベースを読み取れないことを示しています。このエラーは、システムのファームウェアのリビジョン・レベルが適切でない場合に起こります。適切なファームウェア・リビジョン・レベルにアップグレードしてください。
- データベース内の壊れたエントリに関して、コンソール・メッセージで警告が表示される。データベースを再コンパイルして再作成してください。

- 実行時メッセージは、通常、`ddr_config` コンパイラによって生成される構文エラーを示している。`ddr_config` コマンドに対し `-c` オプションを指定してコンパイラを実行すると、構文エラーがすべて修正されるまで、出力データベースが作成されません。

`ddr_config` コマンドに `-h` オプションを指定すると、コマンド・オプションに関するヘルプが表示されます。

4.2.2 データベース・エントリの例

この節では、サード・パーティ製の製品をサポートする `/etc/ddr.dbase` ファイルにおける 2 つの典型的なエントリの例を示します。この情報は、サポートされているデバイスのベンダによって提供されます。インストールされている HP Tru64 UNIX のバージョンに対応する正しいパラメータの設定に関する重要な情報については、ベンダのドキュメントを参照してください。

警告

この節で示すエントリ例は使用しないでください。デバイスのベンダに問い合わせて、最新の構成設定を入手してください。HP では、システムの省略時の `/etc/ddr.dbase` ファイルにリストされているものを除き、サード・パーティ製デバイスの構成情報は提供しません。

4.2.2.1 DDR エントリの例、StorageTek 9840

次の DDR エントリ例は、StorageTek 9840 磁気テープ・デバイスをサポートします。パラメータ、モードの選択、密度、属性の各フィールドの詳細については、`ddr.dbase(4)` を参照してください。デバイスに付属のハードウェア・ドキュメントに、属性の設定についての情報が記載されています。

```
# Please note that this entry is not supported by HP
# It is supplied by StorageTek as a means of enabling
# their tape drive to work in HP Tru64 UNIX.
# If you experience problems when using this drive
# contact StorageTek Technical Support.

SCSIDEVICE
#
# STK 9840
#
Type = tape
Name = "STK" "9840"
#
PARAMETERS:
```

```

TypeSubClass    = 3480
MaxTransferSize = 0x40000    #256k
SyncTransfers   = enabled
WideTransfers   = enabled
Disconnects     = enabled
CmdReordering   = disabled
TaggedQueuing   = disabled
TagQueueDepth   = 0
WCE_Capable     = false
PwrMgmt_Capable = false
LongTimeoutRetry = disabled
ReadyTimeSeconds = 240
DisperseQueue   = false
CMD_PreventAllow = supported
CMD_ExtReserveRelease = supported

```

DENSITY:

```

#
# /dev/tape/tapeX_d0, _d4
#
DensityNumber = 0,4
DensityCode = density_code_42
CompressionCode = 0
Buffered = 0x1

```

DENSITY:

```

#
# /dev/tape/tapeX_d1, _d5
#
DensityNumber = 1,5
DensityCode = density_code_42
CompressionCode = 1
Buffered = 0x1

```

DENSITY:

```

#
# /dev/tape/tapeX_d2, _d6
#
DensityNumber = 2,6
DensityCode = density_code_43
CompressionCode = 0
Buffered = 0x1

```

DENSITY:

```

#
# /dev/tape/tapeX_d3, _d7
#
DensityNumber = 3,7
DensityCode = density_code_43
CompressionCode = 1
Buffered = 0x1

```

Buffered = 0x1

DENSITY:

```

#
DensityNumber = 1
DensityCode = default
CompressionCode = 0x1
Buffered = 0x1

```

4.2.2.2 DDR エントリ例, EMC Symmetrix

次の DDR エントリ例は, EMC Symmetrix ディスク・ストレージ・デバイスとファイバ・チャネル・デバイスをサポートします。パラメータ, モードの選択, 密度, 属性の各フィールドの詳細については, `ddr.dbase(4)` を参照してください。

サードパーティ製デバイスに付属のハードウェア・ドキュメントに属性の設定についての情報が記載されています。この例では, `ATTRIBUTE` 設定には `ubyte[0]` 属性に対して設定し得る値を示すコメントが含まれています。これらの設定については, デバイスのハードウェア・ドキュメントを参照するか, ベンダに問い合わせてください。

```
SCSIDEVICE
#
# SAMPLE Entry for Symmetrix SCSI devices (DO NOT USE, CONSULT EMC)
#
Type = disk
Name = "EMC" "SYMMETRIX"
PARAMETERS:
TypeSubClass = hard_disk, raid
BlockSize = 512
BadBlockRecovery = disabled
DynamicGeometry = true
LongTimeoutRetry = enabled
DisperseQueue = false
TagQueueDepth = 20
ReadyTimeSeconds = 45
InquiryLength = 160
RequestSenseLength = 160
PwrMgmt_Capable = false
ATTRIBUTE:
# ubyte[0] = 8 Disable AWRE/ARRE only, PR enabled
# ubyte[0] = 25 Disable PR & AWRE/ARRE, Enable I/O Barrier Patch resets
AttributeName = "DSBLflags"
Length = 4
ubyte[0] = 8

SCSIDEVICE
#
# Entry for Symmetrix Fibre Channel devices
#
Type = disk
Stype = 2
Name = "EMC" "SYMMETRIX"
PARAMETERS:
TypeSubClass = hard_disk, raid
BlockSize = 512
BadBlockRecovery = disabled
DynamicGeometry = true
LongTimeoutRetry = enabled
DisperseQueue = false
TagQueueDepth = 20
ReadyTimeSeconds = 45
InquiryLength = 160
RequestSenseLength = 160
PwrMgmt_Capable = false
ATTRIBUTE:
```



```
AttributeName = "DSBLFlags"  
Length = 4  
ubyte[0] = 8
```

4.3 DDR データベースの手作業による変更

SCSI デバイスの操作パラメータまたは特性を変更した場合には、
/etc/ddr.dbase ファイルにその変更を記述する必要があります。
ddr_config -c コマンドを使用して、その変更をコンパイルします。

変更を行う一般的な理由は次の 2 つです。

- 使用しているデバイスが SCSI 標準からはずれているか、または SCSI 標準とは異なると報告された。
- デバイスの省略時の設定、ほとんどの場合、TagQueueDepth パラメータ (そのデバイスがサポートするアクティブなタグ付き要求の最大数を指定する) を最適化する。

ddr_config -c コマンドを使用して /etc/ddr.dbase ファイルをコンパイルし、バイナリのデータベース・ファイル /etc/ddr.db を作成します。カーネルは、ファイル状態の変更を認識すると、新しい /etc/ddr.dbase ファイルをロードします。このように、SCSI CAM I/O サブシステムは、/etc/ddr.dbase ファイルに対して行われた変更に基づいて動的に更新され、ディスク上のデータベースの内容が、メモリ内のデータベースの内容と同期化されます。

次の手順に従って、/etc/ddr.dbase データベースをコンパイルします。

1. root としてログインするか、またはスーパーユーザになります。
2. ddr_config -c コマンドを入力します。たとえば、次のように入力します。

```
# /sbin/ddr_config -c
```

正常に完了したことを確認するメッセージは表示されません。プロンプトが表示されると、コンパイルが終了します。構文エラーがあるとエラーが標準出力に出力され、コンパイル済みの出力ファイルは生成されません。

4.4 カスタマイズ済みの cam_data.c 情報の変換

次の手順を使用して、SCSI デバイスに関するカスタマイズ済み情報を、
/sys/data/cam_data.c ファイルから /etc/ddr.dbase テキスト・デー

データベースに転送します。この例では、*MACHINE* は使用しているマシンのシステム構成ファイル名です。

1. `root` としてログインするか、またはスーパーユーザになります。
2. `/etc/DDR.dbase` ファイルに対して行われた追加および変更の要約を作成するために、`DDR_config -x` コマンドを実行します。たとえば次のようにします。

```
# /sbin/DDR_config -x MACHINE > output.file
```

このコマンドは、システム構成ファイルを入力として使用します (構成ファイルは、現在実行中のカーネルを構築したときに指定したファイルです)。この手順はマルチユーザ・モードで実行され、開始後に入力を行う必要はありません。出力をファイルにリダイレクトして、要約情報を保存しておくようにしてください。コンパイル・エラーは標準エラーに報告され、エラーが報告されるとコマンドは直ちに終了します。警告メッセージは標準エラーに出力され、コマンドは終了しません。
3. 出力ファイルにリストされた特性を、`/etc/DDR.dbase` ファイルの構文に従って、そのファイル内に編集します。`/etc/DDR.dbase` データベースの編集方法については、`DDR.dbase(4)` を参照してください。
4. `DDR_config -c` コマンドを入力して、変更をコンパイルします。

詳細は、4.3 節を参照してください。

以降の項で説明する手順に従うと (DDR を使用せずに) 擬似端末、ディスク、およびテープ・デバイスを静的に追加することができます。

4.5 DDR を使用せずに擬似端末とデバイスを追加する

特に指定しなくても System V Release 4 (SVR4) の擬似端末 (ptys) はインプリメントされており、次のように定義されています。

`/dev/pts/N`

変数 *N* は、0 ~ 9999 の数値です。

この機能により、BSD の pty (tty[a-zA-Z][0-9a-zA-Z]) より高いスケーラビリティが得られます。ベース・システムのコマンドおよびユーティリティは、SVR4 と BSD の pty をサポートしています。元の省略時の設定に戻るには、`MAKEDEV` コマンドを用いて BSD ptys を作成します。`SYSV_PTY(8)`、`pty(7)`、`MAKEDEV(8)` も参照してください。

4.5.1 擬似端末の追加

擬似端末を使用することによって、ネットワーク経由でシステムにアクセスすることができます。擬似端末は、システムへのハードウェア端末の接続をエミュレートする 1 組のキャラクタ・デバイスです。ハードウェアの代わりに、マスタ・デバイスとスレーブ・デバイスが存在します。擬似端末は、通常の端末と違い、対応する物理的な端末ポートがシステム上に存在しません。リモート・ログイン・セッション、ウィンドウ・ソフトウェア、およびシェルは、擬似端末を使用してシステムにアクセスします。特に指定しなければ、SRV4 のデバイス特殊ファイル (`/dev/pts/n` など) が作成されます。`/dev/ttyp/n` のような BSD の擬似端末を作成するには、`/dev/MAKEDEV` を使用しなければなりません。利用できる擬似端末の種類は 2 つあります。BSD STREAMS と BSD `clist` の 2 つです。

環境によっては、`pty` デバイスの省略時の数で十分な場合があります。しかし、環境が拡大し、その中で各ユーザが 1 台以上のタイムシェアリング・マシンで複数のセッションを実行しようとするすると、マシンの `pty` の数が足りなくなること考えられます。現在の値を調べるには、次のコマンドを実行します。

```
# sysconfig -q pts
pts:
nptys = 255
```

次のように、`sysconfig` コマンドでこの値を動的に変更することもできますが、リブートするとその変更は無効になります。

```
# sysconfig -r pts nptys=400
```

変更した値をリブート後も維持するには、次の手順に従ってください。

1. スーパユーザ (root) としてログインします。
2. システム構成ファイル `/etc/sysconfigtab` で擬似デバイスを追加または編集します。省略時の設定では、カーネルは 255 個の擬似端末をサポートします。さらに多くの擬似端末をシステムに追加する場合は、システム構成ファイルを編集して、追加したい擬似端末数まで増加させます。次の例では擬似端末を 400 個増加する方法を示しています。

```
pts:
nptys=400
```

`clist` ベースの擬似端末に対する擬似端末エントリは、次のようになります。

`pseudo-device pty 655`

構成ファイルとその擬似デバイス・キーワードの詳細については、『システム管理ガイド』を参照してください。

3. `clist` ベースの擬似端末では、さらに、新しいカーネルを再構築してブートします。新しいカーネルの再構築とブートについては、『システム管理ガイド』を参照してください。

システムを最初にインストールした際には、構成ファイルの擬似デバイスエントリに省略時の設定として 255 の擬似端末数が設定されます。この値が削除されている場合、システムの省略値の擬似端末数は最小の 80 になります。最大値は 131072 です。

BSD 端末を作成する場合は、次のように `/dev/MAKEDEV` コマンドを使用します。

1. `root` としてログインし、`/dev` ディレクトリに移動します。
2. 次のように `MAKEDEV` コマンドを使用して、デバイス特殊ファイルを作成します。

```
./MAKEDEV pty_#
```

(番号記号)には、作成したい擬似端末のセット数 (0 ~ 101) を指定します。最初の 51 セット (0 ~ 50) は、各セットでそれぞれ 16 の擬似端末を作成します。残りの 51 セット (51 ~ 101) は、各セットに対して 46 の擬似端末を作成します。大量の擬似端末を作成する手順については、`MAKEDEV(8)` を参照してください (サポートされている擬似端末の数の上限については、Tru64 UNIX 『*QuickSpecs*』を参照してください)。

注意

省略時の設定では、インストール・ソフトウェアは、擬似端末の最初の 2 セット、`pty0` および `pty1` のデバイス特殊ファイルを作成します。`pty0` 擬似端末には、`/dev/ttyp0` から `/dev/ttypf` までの名前が付いた対応するデバイス特殊ファイルがあります。`pty1` 擬似端末には、`/dev/ttyq0` から `/dev/ttyqf` までの名前が付いた対応するデバイス特殊ファイルがあります。

インストレーション・ソフトウェアは `pty0` と `pty1` を設定するので、システムに擬似端末を追加する場合には、`pty#` 変数は `pty1` より大きくなければなりません。たとえば、擬似端末の 3 番目のセットのためのデバイス特殊ファイルを作成するには、次のコマンドを入力します。

```
# ./MAKEDEV pty2
```

MAKEDEV コマンドは、作成したデバイス特殊ファイルのリストを表示します。たとえば次のように表示されます。

```
MAKEDEV: special file(s) for pty2:
ptyr0 ttyr0 ptyr1 ttyr1 ptyr2 ttyr2 ptyr3 ttyr3 ptyr4 ttyr4
ptyr5 ttyr5 ptyr6 ttyr6 ptyr7 ttyr7 ptyr8 ttyr8 ptyr9 ttyr9
ptyra ttyra ptyrb ttyrb ptyrc ttyrc ptyrd ttyrd ptyre ttyre
ptyrf ttyrf
```

3. BSD の `pty` を削除するには、`/dev/SYSV_PTY` コマンドを使用します。
4. すべての擬似端末で `root` ログインを許可する場合は、`/etc/securettys` ファイルに `ptys` エントリを記述します。擬似端末での `root` ログインを許可しない場合は、`/etc/securettys` ファイルの `ptys` エントリを削除します。たとえば、新しい `tty` ラインのエントリを追加し、すべての擬似端末で `root` ログインを許可する場合は、`/etc/securettys` ファイルに次の行を追加します。

```
/dev/tty08      # direct tty
/dev/tty09      # direct tty
/dev/tty10      # direct tty
/dev/tty11      # direct tty
ptys
```

詳細については、`securettys(4)` を参照してください。

4.5.2 他のデバイスの追加

新しい SCSI デバイスをシステムに追加すると、システムが自動的にそれらを検出して構成します。適切な `hwmgrr` コマンドと `dsfmgr` コマンドを実行して、デバイスを登録し、識別子を割り当て、デバイス特殊ファイルを作成します。しかし、他のデバイスについては、MAKEDEV コマンドを用いて手作業でデバイス名を作成する必要があるかもしれません。間違っシステムから削除されたデバイス特殊ファイルを再度作成しなければならないこともあります。

新しいデバイスの場合には、そのデバイスを物理的に接続して、システムにデバイスを認識させなければなりません。この方法には、静的ドライバ用と

ロード可能ドライバ用の 2 つの方法があります。デバイスを追加する前に、システム・プロセッサに添付されているオーナーズ・マニュアルと、デバイス自体に添付されているドキュメントを必ず読んでください。また、ドライバ・ソフトウェアの入ったディスクが必要になることもあります。

第 8 章 には、システムに PCMCIA モデムを追加する例の概要と、デバイス特殊ファイルの作成方法が示されています。

古いデバイス特殊ファイル `rz` または `tz` を `/dev` に作成する (`/dev/rz5` など) だけであれば、`MAKEDEV` コマンドは使う必要はありません。`dsfmgr` コマンドを使えば、これらのデバイス名を作成できます。ロード可能ドライバにデバイスを追加する方法については、デバイス・ドライバのマニュアルを参照してください。

静的ドライバ用にデバイスを追加するには、4.5.1 項を参照してください。

続いて、次の手順を実行して、そのデバイス用にデバイス特殊ファイルを作成します。

1. `/dev` ディレクトリに移動します。
2. 次のように `MAKEDEV` コマンドを使用して、デバイス特殊ファイルを作成します。

```
# ./MAKEDEV deviceN
```

変数 `device` は、追加しようとしているデバイスを示すデバイス・ニーモニックです。変数 `N` はデバイス番号を示します。たとえば、2 つの PCMCIA モデム・カードのデバイス特殊ファイルを作成するには、次のコマンドを入力します。

```
# ./MAKEDEV ace2 ace3
```

```
MAKEDEV: special file(s) for ace2:
```

```
tty02
```

```
MAKEDEV: special file(s) for ace3:
```

```
tty03
```

作成されるデバイス特殊ファイルは次のようになります。

```
crw-rw-rw-  1 root      system   35,  2 Oct 27 14:02 tty02
crw-rw-rw-  1 root      system   35,  3 Oct 27 14:02 tty03
```

3. `shutdown` コマンドを使用してシステムを停止し、システムの電源を切ります。詳細については、『システム管理ガイド』と `shutdown(8)` を参照してください。

4. システムの電源を入れます。すべてのデバイスをシステムに認識させるためには、周辺装置の電源を先に入れてから、システム本体の電源を入れてください。
5. 新しいカーネルでシステムをブートします。ブートについての詳細は、『システム管理ガイド』を参照してください。



デバイス・コマンドとデバイス・ユーティリティ

これまでの章では、すべてのデバイスを対象にさまざまな管理を行うための汎用ハードウェア管理ツール(第3章で説明した `hwmgrr` コマンドなど)について説明しました。以降の節では、特定の種類のデバイスを対象に特定の作業を行うためのハードウェア管理ツールについて説明します。

- デバイス・ユーティリティに関する情報(5.1 節)
- SCSI ユーティリティの使用(5.2 節)
- `diskconfig` GUI を使用したディスクのパーティショニング(5.3 節)
- `disklabel` コマンドを使用したディスクのパーティショニング(5.4 節)
- ディスクのコピー(5.5 節)
- ディスクのモニタリング(5.6 節)

5.1 デバイス・ユーティリティに関する情報

デバイス・ユーティリティの多くについては、このマニュアルだけでなく、他のドキュメントでも説明しています。たとえば、ネットワーク・デバイスを構成するユーティリティについては、『ネットワーク管理ガイド：サービス編』で詳細に説明しています。表 5-1 に、この章で説明するものも含めて、本書で説明しているユーティリティを示します。その他のユーティリティについては、リファレンス・ページを参照してください。表 5-2 に、リファレンス・ページで説明されているユーティリティのほかに、インタフェースを説明するリファレンス・ページの第7節など、リファレンス・データの参照先も示します。

表 5-1: 本書で説明されているデバイス・ユーティリティ

デバイス	タスク	場所
プロセッサ	起動または停止	『システム管理ガイド』
	リソースの共用	『システムの構成とチューニング』
	モニタリング	『システム管理ガイド』
	電力管理	『システム管理ガイド』 , dxpowers
	メモリのテスト	『システム管理ガイド』
	エラーおよびイベント処理	『システム管理ガイド』
SCSI バス	高度な構成と管理	5.2.1 項 , scu
ディスク	パーティションの作成	diskconfig , disklabel
	コピー	5.5 節 , dd
	利用状況のモニタリング	5.6 節 , df と du
	電力管理	『システム管理ガイド』
	ファイル・システムの状態	『システム管理ガイド』
	テストと動作確認	『システム管理ガイド』
テープ(およびディスク)	アーカイブ	『システム管理ガイド』
	テストと動作確認	『システム管理ガイド』
クロック	設定	『システム管理ガイド』
モデム	構成	『システム管理ガイド』

表 5-2: リファレンス・ページで説明されているデバイス・ユーティリティ

デバイス	タスク	場所
デバイス (汎用)	構成	hwmmgr(8) , devswmmgr(8) , dsfmgr(8)
	デバイス特殊 ファイル	kmknod(8) , mknod(8) , MAKEDEV(8) , dsfmgr(8)
	インタフェース	atapi_ide(7) , devio(7) , emx(7)

表 5-2: リファレンス・ページで説明されているデバイス・ユーティリティ (続き)

デバイス	タスク	場所
プロセッサ	起動と停止	halt(8), psradm(8), reboot(2)
	CPU リソースの割り当て	class_scheduling(4), processor_sets(4), runon(1)
	モニタリング	dxsysinfo(8), psrinfo(1)
SCSI バス	管理	sys_attrs_cam(5), ddr.dbase(4), ddr_config(8)
ディスク	パーティションの作成	diskconfig(8), disklabel(4), disklabel(8), disktab(4)
	モニタリング	dxsysinfo(8), diskusg(8), acctdisk(8), df(1), du(1), quota(1)
	テストと保守	diskx(8), zeero(8)
	インタフェース	ra(7), radisk(8), ri(7), rz(7)
	スワップ領域	swapon(8)
テープ (およびディスク)	アーカイブ	bttape(8), dxarchiver(8), rmt(8)
	テストと保守	tapex(8)
	インタフェース	tz(7), mtio(7), tms(7)
フロッピー	ツール	dxmtools(1), mtools(1)
	テストと保守	fddisk(8)
	インタフェース	fd(7)
端末, ポート	インタフェース	ports(7)
モデム	構成	chat(8)
	インタフェース	modem(7)
キーボード, マウス	インタフェース	dc(7), scc(7)

SysMan グラフィカル・ユーザ・インタフェースが提供するユーティリティのリストについては、『システム管理ガイド』を参照してください。

5.2 SCSI デバイスおよびデバイス・ドライバのユーティリティ

この項では、SCSI デバイスとデバイス・ドライバの管理に使用するユーティリティについて説明します。

5.2.1 SCSI 構成ユーティリティ **scu** の使用

SCSI/CAM ユーティリティ・プログラム **scu** には、SCSI 周辺機器と CAM I/O サブシステムに対して行う、高度な保守と診断に必要なコマンドがあります。日常的操作のほとんどには、**hwmgr** コマンドを使用します。また、**scu** プログラムには、利用できるオプションとその規則を詳細に説明するヘルプ機能もあります。このコマンドの使い方についての詳細は、**scu(8)** を参照してください。

scu は、次の作業に使用できます。

- ディスクのフォーマット
- 不良ディスク・ブロックの再割り当て
- デバイスの予約と解放
- デバイスおよびプログラム・パラメータの表示と設定
- デバイスを使用可能/不可能にする

注意

DSA (Digital Storage Architecture) ディスクには、**radisk** プログラムを使用してください。詳細は、**radisk(8)** を参照してください。

scu の使用例を次に示します。

```
# scu
scu> set nexus bus 0 target 0 LUN 0
Device:RZ1CB-CA, Bus:0, Target:0, LUN:0, Type:Direct Access
scu> show capacity

Disk Capacity Information:

Maximum Capacity: 8380080 (4091.836 megabytes)
Block Length: 512
scu> show scsi status 0
```

```
SCSI Status = 0 = SCSI_STAT_GOOD = Command successfully completed
```

5.2.2 デバイス・スイッチ・マネージャ devswmgr の使用

devswmgr コマンドを使って、デバイス・スイッチ・テーブル内のデバイス・ドライバについての情報を表示し、このテーブルを管理することができます。また、このコマンドでデバイス・スイッチ・テーブルからエントリを解放することもできます。一般に、ドライバのエントリを解放するのは、ドライバをアンロードした後、再ロードする予定がない場合です。エントリを解放すると、そのエントリは他のデバイス・ドライバに使用できるようになります。

devswmgr を使ってデバイス・データを取得する例を次に示します。

```
# devswmgr -display
device switch database read from primary file
  device switch table has 200 entries
# devswmgr -getnum

Device switch reservation list
                                (*=entry in use)
  driver name                   instance  major
  -----
                                pfm        1      71*
                                fdi        2      58*
                                xcr        2       57
                                kevm       1      56*
                                cam_disk   2      55*
                                emx        1       54
                                TMSCP      2       53
                                MSCP      2       52
                                xcr        1       44
                                LSM        4       43
                                LSM        3       42
                                LSM        2      41*
                                LSM        1      40*
                                ace        1      35*
                                parallel_port 1      34*
                                cam_uagt   1       30
                                MSCP      1       28
                                TMSCP      1       27
                                scc        1       24
                                presto     1       22
                                cluster    2      21*
                                cluster    1      19*
                                fdi        1      14*
                                cam_tape   1        9
                                cam_disk   1        8*
```

pty	2	7
pty	1	6
tty	1	1
console	1	0

5.3 diskconfig を使ったディスクのパーティショニング

Disk Configuration グラフィカル・インタフェース (diskconfig) で、次の作業を行うことができます。

- 既存の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。グラフィカル・インタフェースの使い方は、オンライン・ヘルプで見ることができます。コマンド・オプションについての詳細は、disklabel(8) を参照してください。

Disk Configuration GUI のグラフィカル・インタフェースを使って行うディスク保守作業のいくつかは、次のコマンドを用いて手作業で行うこともできます。

- `disklabel` – ディスク・ドライブやディスク・バックのラベルをインストール、検査、および変更するコマンド。ディスク・ラベルには、ディスクの種類、物理パラメータ、パーティション情報など、ディスクに関する情報が入っています。disklabel(4) で説明している `/etc/disktab` ファイルも参照してください。
- `newfs` – 指定したデバイスに新しく UFS ファイル・システムを作成するコマンド。newfs コマンドは Advanced File System (AdvFS) の作成には使用できません。AdvFS の作成には、mkfdomn(8) で説明している mkfdomn コマンドを使用してください。
- `mkfdomn` と `mkfset` – AdvFS (Advanced File System) ドメインとファイルセットを作成するコマンド。

Disk Configuration インタフェースは、次のどちらかの手順で起動します。

- システム・プロンプトに対して `diskconfig` と入力する。
- CDE フロント・パネルの SysMan アプリケーション・ポップアップ・メニューから、[Configuration] を選択する。続いて、SysMan [Configuration] フォルダから [ディスク] アイコンを選択する。

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。必要なデータを上書きしてしまわないように、変更内容を前もって計画しておきます。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 : *hostname*] というタイトルのウィンドウが表示されます。DiskConfig ユーティリティ用のメイン・ウィンドウに、各ディスクについて次の情報がリストされます。

- 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、1.5 節を参照してください。
- デバイスのモデル。たとえば `RZ1CB-CA`。
- バス、ターゲット、および LUN (論理ユニット番号) によって表される物理位置。デバイスの位置についての詳細は、第 3 章を参照してください。

リスト項目をダブルクリック (またはディスクが強調表示されている時に [設定] ボタンをクリック) すると、2 つのウィンドウが同時に表示されます。ウィンドウは重なって表示されます。適切なウィンドウにカーソルを移動して、次のようにしてパーティションを構成するか、ディスクのパーティション・テーブルを編集します。

Disk Configuration:Configure Partitions:デバイス名 デバイス・タイプ

このウィンドウには、次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、区切りの中央にあるハンドル。
- 区切りの右側にあるパーティションの開始位置を右に移動する場合は、上のフラグ。
- 区切りの左側にあるパーティションの終了位置を左に移動する場合は、下のフラグ。

カーソルの位置でMB1を押してマウスをドラッグすることで、ハンドルを移動します。

- ブルダウンメニュー。ユーティリティにディスク・サイズ情報を表示させる方法を選択できます。メガバイト単位 (Megabytes)、バイト単位 (Bytes)、ブロック単位 (Blocks) から選択できます。
- 統計情報ボックス。デバイス名、ディスクの総容量、使用量などディスク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 (エイリアス) 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。
- [ディスクの属性...] 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。[Disk Configuration: Partition Table] というタイトルのウィンドウがポップアップします。

Disk Configuration:Partition Table:デバイス名 デバイス・タイプ

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。次から選択できます。

- 現在表示されているパーティション。

- ディスク・デバイスの標準のテーブル。
- このセッションが開始されたときの元の設定 (ディスク上のパーティション・テーブル) にパーティションを戻す。

手作業によるパーティション変更で間違えたときは、このウィンドウを用いてパーティション・テーブルを元に戻せます。

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. SysMan Menu を起動します。
2. [ストレージ] オプションを展開して、[ファイルシステム管理ユーティリティ] - [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] を選択します。
3. 「マウント操作」ウィンドウで、特定のファイル・システムをマウントするオプションを選択し、[次へ] ボタンをクリックします。
4. 「ファイル・システム名とマウント・ポイント」ウィンドウで、以下を行います。
 - a. `/usr/newusers` などのマウント・ポイントを入力します。
 - b. `/dev/disk/dsk0g` のようなパーティション名、または `newusr_domain#usr` のようなドメイン名を入力します。
5. [了解] ボタンをクリックしてパーティションの調整を適用します。

これで新しいファイル・システムへのアクセスが可能になります。

5.4 手作業によるディスク・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・サイズの変更に必要な情報を示します。一般に、ディスク領域は、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ときに割り当てます。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。

ディスク・ラベルには、ディスクのジオメトリおよびディスクを分割するパーティションに関する詳細な情報が入っています。ラベルは `disklabel` コマンドで変更することができます。 `disklabel` コマンドを使用する場合は、`root` ユーザにならなければなりません。

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、ディスク・ラベルがメモリにコピーされます。メモリ内のラベルではなく、ディスク上のラベルに直接アクセスするには、`disklabel -r` コマンドを使用します。

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ファイル記述子がオープンされているファイル・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、ディスクの先頭 (セクタ 0) から始まるパーティション `a` がマウントされていない場合に限り、`a` の raw デバイスを指定する。パーティション `a` がすでにマウントされている場合には、パーティション `c` を使用してラベルを変更する。パーティション `c` もセクタ 0 から始まっている必要がある。

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。またそうした場合に、ラベルを書き込めなかったというエラー・メッセージも表示されません。

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。disklabel コマンドを使用してパーティション・サイズを表示できます。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は、次の disklabel コマンドを実行します。

```
/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 1498312  unused  1024  8192      # (Cyl. 1877*- 2569*)
f: 819200 393216  unused  1024  8192      # (Cyl. 492*- 1519*)
g: 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` オプションを指定して、省略時のパーティション・ラベルを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。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領域の情報を表示させます。
2. `/etc/fstab` ファイルを表示して、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ディスクのラベルを調べます (省略時のディスク・パーティションについては `rz(7)`、`ra(7)`、そして `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. ラベルを変更する対象とな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の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. 次のように `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、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、ラベルはディスクに直接書き込まれ、可能なら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。

5.4.1 パーティションの重なりの検索

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チェックします。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`、`mkfdmn`、`rmfdmn`、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使用方法を設定します。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、処理を継続するか中断する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。

5.5 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ないので、ディスクまたはパーティションをコピーするときは、データ・ディスクとして使用しているディスク、または、ファイル・システムを含まないディスクをコピー先にします。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、『システム管理ガイド』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。

UNIX では、有効なディスク・ラベルを持つディスクの先頭のブロックが保護されます。そこにディスク・ラベルが保管されているからです。したがって、有効なディスク・ラベルが存在するターゲット・ディスクにパーティションをコピーするときは、そのターゲット・ディスク上のディスク・ラベルを維持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維持する場合には、`dd` コマンドに `skip` または `seek` オプションを指定して実行し、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、ドライバ番号、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、そのラベルが維持したくない場合は、`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維持する場合は、`dd` コマンドを使用します。このコマンドに、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します。

```
# dd if=/dev/rdisk/dsk0c of=/dev/rdisk/dsk1c \
skip=16 seek=16 bs=512k
```

5.6 ディスク使用量の監視

適切な量のディスクの空き領域を確保するには、次のいずれかの方法で、構成済みのファイル・システムのディスク使用量を定期的に監視します。

- `df` コマンドを使用して利用可能な空き領域を確認する (5.6.1 項)

- `du` コマンドおよび `quot` コマンドを使用してディスクの使用量を確認する (5.6.2 項)
- ディスク・クォータを設定している場合は、`quota` コマンドを使用してディスク・クォータを確認する (5.6.3 項)

`quota` コマンドは、`root` ユーザのみが使用できます。

5.6.1 利用可能な空き領域の確認

ファイル・システムに十分な領域を確保するには、`df` コマンドを定期的に使用して、マウントしたファイル・システム全体のディスクの空き領域の容量を確認します。`df` コマンドは、指定されたファイル・システムの空き領域、または指定されたファイルを含むファイル・システムの空き領域についての統計値を表示します。

`df` コマンドの引数やオプションを省略すると、次が表示されます。

- 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領域。
- 各ファイル・システムの構成サイズ。512 バイト・ブロック単位。
-k オプションを指定すると、サイズ情報は K バイト・ブロック単位になります。
- 領域の総容量、使用中の容量、利用可能な (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。
- AdvFS ファイル・ドメインについては、`df` コマンドは各ファイルセットのディスク使用量を表示する。

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・パス名を指定すれば、そのファイルを含んでいるファイル・システムで利用可能なディスク容量を表示させることができます。詳細については、`df(1)` を参照してください。

注意

`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、アンマウントされているファイル・システムの空き容量を知ることはできません。この場合には、`df` コマンドを使用してください。

次の例では、マウントされているすべてのファイル・システムのディスク領域に関する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/dsk2g   101372    5376   85858      5%  /var
/dev/disk/dsk3c   394796     12  355304      0%  /usr/users
/usr/share/mn@tsts 557614  449234   52620     89%  /usr/share/mn
domain#usr       838432  680320  158112     81%  /usr
```

注意

newfs コマンドは、割り当ておよびブロック・レイアウトのために一定の割合のファイル・システム・ディスク領域を確保します。これにより、ファイル・システムがその容量の 100 パーセントを超えて使用していることを df コマンドが報告するという現象が発生します。tunefs コマンドに `-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。

5.6.2 ディスク使用量の確認

ファイル・システムの容量が不足していると判断した場合は、du コマンドまたは quot コマンドを使用して、誰が領域を使用しているかを知ることができます。

du コマンドは、ディレクトリごとに、ディスク領域の割り当てを表示します。この情報によって、ディスク領域を最も多く使用しているユーザやディスク領域を解放すべきユーザを判断することができます。

du コマンドは、指定されたディレクトリやファイル名、または(それらが指定されていない場合は)現在の作業ディレクトリ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)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・カウントには、各ファイルの間接ブロックが、1K バイト単位で含まれます。

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が生成されます。コマンド・オプションについての詳細は、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が使用す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/morse
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領域を使用していることがわかります。

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が使用する容量を表示しています。

```
# /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
```

注意

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。使用量を取得するために 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。

5.6.3 ディスク・クォータの確認

quot コマンドを使用して、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。

次のコマンドは、/dev/disk/dsk0h ファイル・システムの中で各ユーザが所有しているブロックの数とファイルの数を表示します。

```
# /usr/sbin/quot -f /dev/disk/dsk0h
```

注意

デバイスがマウントされているとブロック型特殊ファイルはビジーになるため、情報を表示するためには文字型特殊ファイルを指定しなければなりません。

詳細については、`quot(8)` を参照してください。



プロセッサ固有の情報

本章では、AlphaServer プロセッサの特定のモデルとクラスをサポートするためのオペレーティング・システム機能に関する情報を提供します。また、特定のプロセッサ・モデルに対して恒久的で限定的な構成上の制限事項についても説明します。次の情報が含まれます。

- 古いプラットフォーム向けのプロセッサ固有の情報 (6.1 節)
- AlphaServer TS202c 向けのプロセッサ固有の情報 (6.2 節)
- AlphaServer GS140 での論理パーティションの構成 (6.3 節)
- AlphaServer 1000 および 1000A の構成情報 (6.4 節)
- AlphaServer GS シリーズの構成情報 (6.5 節)
- Personal Workstation 433au , 500au , および 600au systems (6.6 節)

6.1 旧式のプロセッサ

旧いシステムに関しては以下の注意事項があります。

Alpha VME シングルボード・コンピュータ

Alpha VME シングルボード・コンピュータ (SBC) および PCI/ISA EBMnn モジュール SBC 向けにオペレーティング・システムを構成する方法の詳細については、『*System Configuration Supplement: OEM Platforms*』を参照してください (PCI/ISA モジュール・システムおよびそのコンポーネントの製品ファミリは、以前は DIGITAL Modular Computing Components DMCC と呼ばれていました)。

VME バスに対するサポートは、オペレーティング・システムの将来のリリースでは廃止される予定です。それに伴い、このバス技術を使用するシステムやオプションも廃止されます。

EISA 設定ユーティリティ

Tru64 UNIX およびそのソフトウェア補助機能でサポートされている ECU (EISA Configuration Utility) はバージョン 1.10 以上です。使用し

ているシステムで EISA バスが構成されている場合には、サポートされているこのバージョンまで ECU をアップデートしてください。

6.2 AlphaServer TS202c

AlphaServer TS202c システムは、デュアルプロセッサのシステムで、16 GB のメモリを搭載するように構成できます。このシステムには、2 つの cPCI スロットがありますが、ディスク・ストレージ・デバイスなしで構成されています。システムは、ネットワーク・ブート可能なスタンドアロン・システム (SAS) カーネルを実行するようになっています。

本節には、Tru64 UNIX バージョン 5.1B について、AlphaServer TS202c システム固有の情報が含まれています。この情報は、2001 年 5 月に『*Release Notes and Installation Instructions for AlphaServer TS202c Systems*』として発行されました。本リリースでは、インストール情報は Tru64 UNIX『インストール・ガイド』に組み入れられました。

次の情報を提供します。

- AlphaServer TS202c で構成できる固有のオペレーティング・システム機能 (6.2.1 項)
- 特定のオペレーティング・システム機能の使用関する最新の制限事項 (6.2.2 項)
- mksas コマンドを使用したディスクレス・システム用ネットワーク・ブート可能なカーネルの作成 (6.2.3 項)
- 除外メモリ領域の構成 (6.2.4 項)
- AlphaServer TS202c の構成 (6.2.5 項)

6.2.1 オペレーティング・システム機能

次の機能は、AlphaServer TS202c のリリースをサポートするために Tru64 UNIX バージョン 5.1 のベース・オペレーティング・システムで初めて追加された機能です。

- ディスクレス環境の AlphaServer TS202c システムで稼動するネットワーク・ブート可能なスタンドアロン・カーネル。6.2.3 項および mksas(8) を参照してください。

- 除外メモリ領域。6.2.4 項, および `dump_exmem*` 属性について記述されている `sys_attrs_generic(5)` を参照してください。
- システムのコア・ダンプ・ファイルのメモリへの保存とリモート・ホストへの書き出し。 `savecore(8)` と `vmzcore(7)` を参照してください。
- 共用オブジェクトがランタイム・ローダによってメモリにロードされるときに割り当てられるメモリ・セグメントのサイズと範囲の指定。 `loader(5)` を参照してください。
- 新しいオペレーティング・システム機能と新しいハードウェア機能をサポートする新しいハードウェア・エラー・コード。これらのエラー・コードの表示方法と解釈の詳細については, `binlogd(8)` を参照してください。新しいコードは次のとおりです。

- 113 - Double error halt.

システムがメモリ・エラーを検出したことを示します。システムが致命的なエラーの処理中に第 2 の致命的エラーを検出しました。メモリ巡視機能によって検出されたエラーは、回復可能なエラー、回復不能なエラー、またはダブルビット・エラーです。

- 115 - (Un)correctable environmental error.

環境モニタリング・デーモンがエラーを検出しました。

- 120 - Reporting of correctable errors is disabled because the single-bit error reporting threshold has been reached.

6.2.2 オペレーティング・システム機能の制限事項

以下の使用上の制限事項は、Tru64 UNIX バージョン 5.1B にのみ当てはまります。

ファームウェア要件

AlphaServer TS202c でネットワーク・ブート可能カーネルを実行するには、ファームウェアのリビジョン・レベルがバージョン X6.0-0 以上でなければなりません。

RIS からのインストール

本リリースのオペレーティング・システムでは、RIS サーバからのインストールをサポートしていません。本ソフトウェアは、インストール用の CD-ROM からのみインストールできます。

誤ったフォーマットの rc.config ファイル

EVM (Event Manager) は、rc.config ファイルに空行が含まれていると異常処理をすることがあります。発生しうる処理異常としては、EVM デーモンによるコア・ダンプまたはブート処理中のシステムによる次のエラーの表示があります。

```
evmwatch: Failed to create EVM listening connection
evmwatch: Error: Connection error
Failed to create EVM listening connection
evmwatch: Error: Connection error
S97evm: Communication with syslogd is not functioning
evmpost: Failed to create EVM posting connection
evmpost: Error: Connection error
evmpost: Failed to create EVM posting connection
evmpost: Error: Connection error
evmpost: Failed to create EVM posting connection
evmpost: Error: Connection error
SysMan authentication server started
SysMan Station Server (smsd) started
The system is ready.
```

環境モニタによる不正なエラー・メッセージの表示

/usr/sbin/envconfig コマンド行ユーティリティを使って ENVMON_HIGH_THRESH 属性の値を変更すると、次のエラー・メッセージが表示されます。

```
env_high_temp_thresh: attribute does not allow this operation
```

このメッセージにもかかわらず、属性値は変更され、デーモンは再スタートされます。

正しい属性値を表示するには、次のコマンドを使います。

```
# /usr/sbin/envconfig -q
```

バイナリ・エラー・ログの不正なエントリ

システムは Correctable Environmental Machine Check エラー (Error 686) を Uncorrectable Environmental Machine Check エラー (Error

682) として、バイナリ・エラー・ログに誤って記録します。これは、AlphaServer TS202c システム固有の問題で、オペレーティング・システムの将来のリリースでは修正されます。

バイナリ・エラー・ログ・イベントがダブル・エラー停止として不正に報告されることがある

タイプ 113 のバイナリ・エラー・ログ (binlog) イベントは、EVM (Event Manager) が報告するときは Double Error Halt イベントと報告され、分析ユーティリティが報告するときは Console Data Log イベントとして報告されます。EVM はこのイベントの発生を、root ユーザにメールするか、システム・コンソールに表示することによって知らせます。

イベントは実際には Console Data Log イベントです。このタイプのイベントは、ダブル・エラー停止、回復不能な環境エラー、プラットフォーム固有のシステム障害など、いくつかの種類のエラーが生じると報告されます。イベントの原因については、その変換データを参照してください。

hwautoconfig がサブシステムのデバイス特殊ファイルを作成しない

システムのブート時に、hwautoconfig ユーティリティによってサブシステムが構成されますが、サブシステムのデバイス特殊ファイルが作成されません。サブシステムのデバイス特殊ファイルを作成するには、sysconfig -u コマンドを使用してサブシステムの構成解除を行い、sysconfig -c コマンドを使用してサブシステムを再構成します。

または、autosysconfig ファイルの中で環境変数 SUBSYSTEM_LIST にサブシステムの名前を追加してから、hwautoconfig ユーティリティを実行することによって問題を回避することもできます。

kmem_debug_leak フラグがセットされるとアイドル状態のシステムでパニックが発生することがある

アイドル状態の AlphaServer TS202c システムで kmem_debug が有効になっていて kmem_debug_leak フラグがセットされると、パニックが発生することがあります。この問題を回避するには kmem_debug_leak フラグをセットしないようにします。

SEL ログに不正なコードが記録される

AlphaServer TS202c システムで 670 および OS Panic イベントが記録されるとき、SEL イベント情報が正しく記録されないことがあります。これらのイベントは、第 3 バイトが DF にセットされて記録されますが、本来は 660/670 のエントリについては C0、OS Panic については C1 がセットされるはずです。

この不正な記録によって、以下のイベントについて重複が生じます。

670 イベント・コード

- x40 PAL detected Bugcheck Error
- x43 EV68 detected Dcache Tag Parity Error
- x45 EV68 detected Duplicate Dcache Tag Parity Error
- x46 EV68 detected Bcache Tag Parity Error
- x4A EV68 detected Double Bit ECC Memory Fill Error
- x5C EV68 detected Second Dcache Store Data ECC Error

OS Panic コード

- x40 ku_recvfrom - not SO_NAME
- x45 m_copym offset
- x46 m_copym sanity
- x43 nfs3_bio: write count
- x4A nfs_dgreceive 3
- x5C rtinithead

これらのイベントはバイナリ・エラー・ログでは正しく記録されます。

6.2.3 mksas コマンドの使用

以下の使用上の注意事項は、AlphaServer TS202c システムで mksas コマンドを使用する場合にのみ当てはまります。

6.2.3.1 構成ファイルの要件

mksas ユーティリティは、ブート可能イメージとインメモリ・ファイル・システムを生成するために使用します。このイメージは、ネットワークを介してシステムをブートし、ディスクレスで動作する機能を提供します。mksas を使用するには、ターゲット・システムのための構成ファイルが必要です。たとえば、ターゲット・ホストが TS2ONE なら、ホスト・システ

ム TS2TWO 上でブート可能なイメージを生成するためには、TS2ONE のための構成ファイルが TS2TWO 上に必要です。

6.2.3.2 su コマンドの使用に関する制限事項

mksas ユーティリティを使用して作成したカーネルを実行しているときは、su コマンドを使用して root 以外のユーザ ID に切り替えることができません (この問題は、不正な umask が mksas ユーティリティで設定されることが原因です)。

6.2.3.3 省略時の構成ファイルに ftp が含まれていない

システムを構成してコア・ダンプ・ファイルをリモート・システムに書き出すようにするには、省略時の構成ファイルを編集して ftp コマンドをオプションとして含める必要があります。省略時の構成ファイルの編集については、6.2.3 項と mksas(8) を参照してください。

6.2.3.4 mksas コマンドが警告メッセージを発する

mksas コマンドに指定した構成ファイルに重複するエントリが含まれていると、次のようなエラー・メッセージが表示されます。

```
Entry no: 160 -> /usr/lib/sabt/etc/mksas.inittab /etc/inittab
WARNING Entry no: 160. Duplicated entry. The file
/etc/inittab in the second field
is already given earlier.
Entry no: 161 -> /etc/rc.config /etc/rc.config
WARNING Entry no: 161. Duplicated entry. The file
/etc/rc.config in the second field
is already given earlier.
Entry no: 162 -> /etc/rc.config.common /etc
WARNING Entry no: 162. Duplicated entry.
The file/etc/rc.config.common in the second field
is already given earlier.
Entry no: 163 -> /etc/hosts /etc/
WARNING Entry no: 163. Duplicated entry.
The file /etc/hosts in the second field
is already given earlier.
```

このメッセージは無視してください。カーネルはエラーなしで構築されます。

6.2.3.5 addlist ファイルのサンプルを提供

システムには、/usr/lib/sabt/etc/addlist_file のサンプルが含まれています。addlist_file を使用して、miniroot ファイル・システムに追加

ファイルを含めます。このサンプルは /usr/lib/sabt/etc ディレクトリにあり、mksas.inv という名前です。

6.2.3.6 処理の終了時に作業ディレクトリを削除する mksas の -t オプション

-t *directory_name* オプションを mksas コマンドで使用して、カーネル構築時に使用される一時作業領域を指定できます。mksas ユーティリティは処理を終了すると、一時作業領域とその作業領域を含んでいたディレクトリを削除します。-t オプションを使用するときは、削除したくないファイルが含まれている場所を *directory_name* に指定しないでください。

6.2.4 除外メモリの割り当て

除外メモリは、UNIX によって管理されるけれども、テストとコア・ダンプの対象とならないメモリです。メモリの除外領域は、contig_malloc() を使って割り当てることができます。これは、擬似デバイス・ドライバの postconfig_callback ルーチンの中で行えます。

除外メモリの割り当ての手順は、次の点を除き、通常の連続メモリ割り当て (malloc) に似ています。

- type フィールドは M_EXEMPT です。

メモリの利用状況は、vmstat -M コマンドを使って監視できます。表示出力の中で、次の例に示すような形式になっているセクションを探してください。

Memory usage by Type and Number of bytes being used

ADVFS	= 7088016	KERN	= 1317456	SONAME	= 11264
AIO	= 8192	KERNEL TBL	= 456912	STREAMS	= 1856
ANON	= 111200	KEVM	= 65536	STRHEAD	= 2560
.					
.					
.					

- addrlimit フィールドは、開始アドレスを指定できるようにオーバーロードされています。使用できないアドレスを指定すると、呼び出しは失敗します (この場合、alignment パラメータは何の効果もありません)。

呼び出しは、次のような形式になります。

```
addr = contig_malloc(size, alignment, start_addr, M_EXEMPT, flags);
```

			割り当て開始アドレス、または
			未指定の場合はゼロ。
			start_addr が指定されていない場合は割り当てのアラインメント。
			指定がなければベースのページ・サイズとなる。
			バイト単位の割り当てのサイズ。必須。

除外メモリは、`cma_dd` サブシステムを次のように `sysconfigtab` エントリを使って定義することによって構成することもできます。

```
CMA_option = size - 0x40000, alignment - 0, Addrlimit - 0, Type - 150, Flag - 1
```

引数は前記の呼び出しと同じです。150 という値は `malloc.h` の内容に対応します。

割り当て領域に不良ページがあると、呼び出しによって返される `addr` の下位ビットがセットされます。

6.2.5 AlphaServer TS202c の構成

本節では、AlphaServer TS202c について次の構成手順を示します。

- ネットワーク・ブート可能カーネルの作成方法 (6.2.5.1 項)
- サーバとクライアントの構成方法 (6.2.5.2 項)

6.2.5.1 ネットワーク・ブート可能なカーネルの作成

`mksas` ユーティリティを使ってネットワーク・ブート可能 (SAS) カーネルを作成します。特に指定しなければ、SAS カーネルはシステムをシングルユーザ・モードでブートするために必要な最小限の数のファイルとディレクトリの構成で作成され、メモリ・ファイル・システムに格納され、インメモリのファイル・システムとして動作するようになっています。テストとデバッグのために、ディスク・ブート可能な SAS カーネルを作成することもできます。`mksas` ユーティリティは、カーネルを実行するシステム上にあるカーネル構成ファイルを使ってカーネルを作成します。その後に、追加ファイルを追加します。

次の手順は、SAS カーネルを作成し、ネットワークを介してブートする方法の例を示します。デバッグの手順も説明します。

1. ホスト・システムを構成して、`mksas` ユーティリティをサポートするバージョンの Tru64 UNIX オペレーティング・システムを実行させます。このホスト・システムには、AlphaServer TS202c カーネル構成ファイルのコピーも含まれていなければなりません。
2. ホスト・システムの上で、ネットワーク・ブート可能な SAS カーネルと `miniroot` ファイル・システムを作成します。デバッグのために、この SAS カーネルと `miniroot` ファイル・システムのディスク・ブート可能なコピーも作成しておきます。

3. ホスト・システムを構成して BOOTP と `tftp` を実行するようにして、ネットワークを介して SAS カーネルをブートするクライアント・システムを決めます。
4. クライアント・システム上で、`boot` コマンドに、次のようにブート・デバイスとしてネットワーク接続を指定して実行します。

```
>>> boot eia0
```

追加機能と `miniroot` ファイル・システムのファイルを追加するには、`-a` オプションを使用して、`addfile_list` または追加ファイルのリストを `mksas` コマンドに指定します。このコマンドに指定できるオプションの一覧については、`mksas(8)` を参照してください。省略時の構成にファイルを追加するときには、依存関係のあるファイルをすべて含める必要があります。たとえば、`savecore` コマンドを使ってクラッシュ・ダンプ・ファイルをリモート・ホストに書き出す場合には、構成に `/usr/sbin/ftp` と `/usr/sbin/ftpd` を含める必要があります。

ファイルを追加するとき、`-C` オプションを `mksas` コマンドに指定してファイル・システムのサイズを調べることができます。このオプションでは、`miniroot` ファイル・システムのサイズはわかりますが、カーネルのサイズは含まれません。通常、カーネルは約 13 MB です。ファイル・システムの合計サイズを算出するには、このサイズを `-C` オプションによって返されるサイズに足します。AlphaServer システムは、92 MB までのサイズのイメージをブートできます。

6.2.5.2 サーバとクライアントの構成

ネットワーク・ブート可能な SAS カーネルを作成したら、SAS カーネルのブート元となるサーバ・システムを構成する必要があります。また、クライアント・システムを構成して SAS カーネルをブートして実行するようにします。サーバ・システムには、BOOTP に対するサポートと、RFC 1782 と 1783 に対応した `tftp` プログラムが必要です。

サーバを構成するには、次の手順に従います。

1. `/etc/inetd.conf` ファイルを編集して、`joind` と `tftp` がクライアントからのブート・リクエストに応えるようにして、SAS カーネル・イメージのブート元となるディレクトリを指定します。それには、

次の行のコメントを解除し、tftpd の行の末尾に正しいディレクトリを追加します。

```
#tftpd dgram udp wait root /usr/sbin/tftpd tftpd /tmp
#bootps dgram udp wait root /usr/sbin/joyind joyind
```

/tmp ディレクトリの指定を削除し、SAS カーネルを格納しておくディレクトリを代わりに指定します。

2. rcmgr コマンドを使って、joyind のランタイム構成変数を設定します。

```
# rcmgr set JOIND yes
```

```
# rcmgr set JOIND_FLAGS ""
```

3. /etc/bootptab ファイルを作成または編集します。このファイルは、サーバがリモート・クライアントをブートするのに必要な情報を含んでいるテキスト・ファイルです。xjoin GUI を使用して、/etc/bootptab ファイルを編集または作成します。詳細については、xjoin(8) および bootptab(4) を参照してください。

このファイルに次の行を追加します。

```
.ris.dec:hn:vm=rfc1048:sm=255.255.255.0:
.ris0.alpha:tc=.ris.dec:bf=/usr/mksas.kernel:
anchor:tc=.ris0.alpha:ht=ethernet:gw=16.142.160.1:ha=00508B6B32CC:
ip=16.142.160.55:
```

.ris.dec エントリは、すべてのクライアントに居る特性を定義します。それぞれのフィールドは、次のことを指定します。

- hn: – ブート・サーバに対して、クライアント・システムがブート・リクエストを発行したときに、そのクライアントにクライアントの名前を送信するよう指示する。
- vm: – ベンダ固有の情報を定義する。

.ris0.alpha エントリは、bootp サーバを使用するすべてのクライアントに共通する特性を定義します。それぞれのフィールドは、次のことを指定します。

- tc: ポインタをたどって共通エントリまで戻れるようにする。たとえば、.ris0.alpha の tc エントリは .ris.dec エントリをサポートします。.ris.dec エントリには、共通ハードウェア・タイプ(ht) とベンダ固有(vm) の情報が含まれています。.ris0.alpha エントリそのものには、ブート・ファイルの場所に関する共通の情報が含まれています。

- bf: ブート・ファイルの名前を指定する。

例に示したホスト名エントリは、特定のクライアントの特性を定義します。それぞれのフィールドは、次のことを指定します。

- tc: ブート・ファイルの情報を含む ris0.alpha を指定する。
ris0.alpha エントリは、関連するハードウェアのタイプとベンダ固有の情報を含む ris.dec を指定します。
- ht: クライアントハードウェアのタイプを ethernet, fddi, または ieee802 (Token Ring) のいずれかとして指定する。
- ha: クライアントのネットワーク・ハードウェアのアドレスを指定する。
- gw: ゲートウェイを定義する。

4. 次のコマンドを使って joinnd サーバ・プロセスを開始します。

```
/sbin/init.d/dhcp start
```

クライアントをブートするには、次のコマンドを使います。

```
>>> boot eia0
```

デバイス名 eia0 は、ネットワーク接続の名前と同じです。デバイス名は、AlphaServer システムによっては異なる場合があります。次のコンソール・コマンドを使ってデバイス名を調べることができます。

```
>>> show dev
```

64 MB を超える大きさの SAS カーネルをブートする場合、eia0_TFTP_blocksize コンソール変数の値を 1450 に設定する必要があります。

```
>>> set eia0_TFTP_blocksize 1450
```

クラッシュ・ダンプ・ファイルを除外メモリまたはリモート・ホストに書き出すことができます。除外メモリ・ダンプは、dump_to_memory システム属性を設定し、メモリ・コア・ダンプの発生時にカーネル・ダンプ・サブシステムが使用する除外メモリまたはメモリ全体を指定することによって、行われるようになります。システム属性 dump_exmem_addr は、ダンプ可能な領域の開始位置を (仮想アドレスまたは物理アドレスとして) 指定します。システム属性 dump_exmem_size は、利用可能なバイト数を指定します。特に指定しなければ、除外メモリの内容はダンプに含まれません。これを変更する

には、システム属性 `dump_exmem_include` を設定します。これらのシステム属性の詳細については、`sys_attrs_generic(5)` を参照してください。

クラッシュ・ダンプ・ファイルをリモート・ホストに書き出すには、`-r` オプションを `savecore` コマンドに指定して使います。このオプションを使用するときは、ホスト、ユーザ名、パスワード、リモート・ホストに書き出すファイルを指定します。構成ファイルを使用して `ftp` 情報と書き出すファイルの名前を指定できます。構成ファイルを使用すれば、ユーザ名とパスワードの秘密を保てます。詳細については、`savecore(8)` を参照してください。

6.3 AlphaServer GS140 論理パーティション

1 つの AlphaServer GS140 システムは、最高 3 つの論理パーティションに分割することができます。各パーティションは、専用のハードウェア・リソース群に割り当てられます。オペレーティング・システムやアプリケーション・ソフトウェアは、パーティションを 1 つの AlphaServer GS140 システムとして扱います。

論理パーティションは、非共用モデルを採用しています。すなわち、1 つのパーティションに割り当てられたすべてのハードウェア・リソース (プロセッサ、メモリ、および I/O) は、そのパーティションに隔離されます。パーティションで実行しているオペレーティング・システムのインスタンスだけが、そのパーティションのハードウェア・リソースにアクセスできます。

論理パーティションを使用すれば、必要なフロア・スペースや電力消費を削減し、(コンピュータ・ルームを冷却する必要性が減少するので) 熱の発散効率を向上させることができます。たとえば、1 つの企業内の 2 つの部門が、コンピュータ利用について異なるニーズを持っていることがあります。この場合は、異なるアプリケーションを実行しなければならないこともあるでしょうし、オペレーティング・システムの構成や調整が異なることもあるでしょう。論理パーティショニングにより、1 台の AlphaServer GS140 コンピュータで、両方の部門のコンピューティング・ニーズを満たすことができます。

6.3.1 ハードウェアの要件

パーティションのハードウェア要件は、次のとおりです。

- センタ・プレーン・スロットを少なくとも 6 つ備えた 1 台の AlphaServer GS140

AlphaServer GS140 6-525 でのみサポートされています。新たにサポートされるシステムについての詳細は、『*Systems and Options Catalog*』を参照してください。論理パーティション機能は、AlphaServer GS140 システムでサポートされます。AlphaServer 8400 (プロセッサ・モジュールを交換して GS140 にアップグレードしたもの) でもサポートされます。

- 1 台のコンソール・デバイス

このコンソール・デバイスは、キャラクタ・セル・ビデオ端末でも、他のシステムまたはターミナル・コンセントレータへのシリアル回線接続でもかまいません。サポートされているグラフィック・デバイスは、オペレーティング・システムのウィンドウ・ソフトウェアで使用できますが、コンソール・デバイスとしては使用できません。

ウィンドウ・ソフトウェアをコンソール・デバイスにできないという、グラフィック・デバイスの制限は、2 次パーティションにのみあてはまります。サポートされているグラフィック・デバイスは、1 次パーティション (パーティション 0) のコンソールにすることはできます。グラフィック・コンソールを使用するには、パーティションの初期化前に、`console` 環境変数の値に `BOTH` を設定します。たとえば次のようにします。

```
P00>>>set console BOTH
```

AlphaServer GS140 には、コンソール・シリアル・ポートが 1 つ用意されています。このポートは、最初のパーティション (パーティション 0) のコンソールとなります。パーティションを追加するごとに、KFE72 オプションのインストールが必要です。このオプションには、シリアル・ポートが 2 つ含まれています (ポート 0 はコンソール・ポート)。KFE72 オプションとインストールの方法についての詳細は、ハードウェアのマニュアルを参照してください。

- 1 つのデュアル・プロセッサ CPU モジュール
- 1 つの I/O ポート (IOP) モジュール

パーティションにおける最低要件は、IOP モジュールが 1 つあることです。パーティションには 2 つ目の IOP モジュールが含まれる場合もあります。システム全体 (全パーティションの合計) では、IOP モジュールは 3 つまで使用できます。

- XMI ハードウェアを論理パーティションで 사용할 수 있습니다. ただし, XMI 컨트롤러와デバイ스는, 파티션 0에 구성되어야 합니다. 이는, 콘솔·펌웨어의 제한입니다.
- 1개의 메모리·모듈
 파티션이 지원하는 최소 메모리·사이즈는, 512 MB입니다. 하지만, 파티션에서 실행되는 애플리케이션によっては, 하한의 512 MB를 초과하는 메모리를 필요로 하는場合があります.
- 소프트웨어를 로드하기 위한 소스·장치 (CD-ROM 드라이브 또는 네트워크·아답타)
- 버전 5.4-19 이상의 AlphaServer GS140 콘솔·펌웨어
 시스템의 논리 파티션을 설치하여 구성할 때는, 설치 대상 오퍼레이팅·시스템의 『릴리스·노트』를 참조하고, 필요에 따라 펌웨어를 업데이트합니다. 펌웨어의 업데이트에 대한 자세한 내용은, 『인스트レーション·가이드』를 참조하십시오.

이제, 이 절에서는 파티션을 구성하기 위해 실행하는 작업을 설명하고, 파티셔닝된 AlphaServer GS140 시스템을 관리하는 방법에 대한 정보를 제공합니다. 이 절에서는, 다음 작업에 대해 설명합니다.

- 파티셔닝된 시스템의 인스트レーション과 작업의 준비
- 시스템·하드웨어가 파티션에 대해 올바르게 구성되어 있는 것의 확인
- 시스템의 콘솔·펌웨어의 리ビジョン·레벨의 확인과, 필요에 따라 펌웨어의 업그레이드
- 논리 파티셔닝·콘솔·펌웨어의 환경변수 (EV)를 생성함으로써, 시스템의 파티션의 구성
- 파티션과 부트스트랩 2차 파티션의 콘솔·모드 (P##>>> 프롬프트)로의 초기화
- 각 파티션에 대한 UNIX 애플리케이션·소프트웨어의 인스트レーション
- 파티셔닝된 시스템의 운용과 관리

6.3.2 論理パーティションのインストールと操作の準備

システムに添付されているハードウェア・ドキュメントを読んで、システムの操作方法を覚えてください。パーティショニングに特に関係するのは、システムの OFF/SECURE/ENABLE/RESET スイッチおよびいくつかのコンソール・コマンド (たとえば、`boot`、`create`、`init`、`set`、および `show` など) です。

パーティションをセットアップする前に、システム・ハードウェアが完全にインストールされ、すべての自己診断テストが正常終了することを確認します。

注意

パーティションへのオペレーティング・システムのインストールを開始する前に、パーティショニング手順を一度通読してください。パーティショニングされたシステムには、システムを運用可能な状態にする前に、管理について確認しておくべきことがあります。あるパーティションでのコンソール操作が別のパーティションの運用を妨げないように、予防措置を講じる必要があります。

次の項では、本書で使用する論理パーティションの用語を説明します。これらの用語を確認したら、6.3.3 項に進んでください。

6.3.2.1 一般に使用される用語の定義

パーティションを構成する前に、次の用語について知っておきましょう。

論理パーティション

オペレーティング・システムの 1 つのインスタンスが専用を使用するための、シングル・システム内のハードウェア・リソース (CPU、I/O、MEMORY、およびコンソール) の論理的なグループ化。1 つの物理システムが複数の論理パーティションを備え、それぞれのパーティションでオペレーティング・システムの独立したインスタンスを実行することもあります。

1 次パーティション

パーティション番号 0。パーティションが無効になっている場合に (つまり、すべてのハードウェア・リソースが 1 つのパーティションにある)、アクティブなコンソール端末を持つパーティション。

2 次パーティション

番号が 0 より大きいパーティション。1 次パーティションのコンソールで `lpinit` コマンドが実行された後にコンソール・プロンプトを表示するパーティションの 1 つ。

1 次コンソール

1 次パーティションに接続されているコンソール端末。パーティションが無効な場合に唯一アクティブなコンソール端末です。

2 次コンソール

2 次パーティションに接続されているコンソール端末。パーティションが有効な場合のみアクティブになります。

電源 OFF/ENABLE スイッチ

AlphaServer GS140 コントロール・パネルにある 4 点スイッチ。スイッチの 4 点は次の機能を実行します。

- OFF – システム電源 (全部のパーティション) がオフ。
- SECURE – 電源がシステム (全部のパーティション) に供給されています。1 次コンソールの `ctrl/p halt` 機能が無効です。
- ENABLE – 電源がシステム (全部のパーティション) に供給されています。1 次コンソールの `ctrl/p halt` 機能が有効です。
- RESET – これは瞬間的なポジションです。スイッチを RESET の位置にしてから放すと、システムが完全に初期化されます。2 次パーティションはすべて即座に終了します。1 次パーティションは、通常のパワー・オン自己診断メッセージを表示し、コンソール・モードに入ります。

コンソール・プロンプト

パーティションのコンソール端末上に表示され、コンソール・ファームウェアがコマンドを受け付ける準備ができたことを示すプロンプト。

P##>>>

ここで ## は、コンソール・ファームウェアが現在実行されているプロセッサの番号です。これは、以降の例に示すように、通常、現在のパーティションの 1 次プロセッサです。

- CPU 0 のパーティション 0:

P00>>>

- CPU 4 のパーティション 1:

P14>>>

ctrl/p halt

コントロール・キーを押したまま、文字 p を押すと、パーティション 0 の 1 次プロセッサが停止してコンソール・モード (P00>>> プロンプト) に入ります。これは、1 次コンソール端末のみで可能です。halt 操作は、電源スイッチを SECURE の位置に合わせることで無効にできます。halt 操作は、2 次パーティションでは無視されます。

P##>>>stop N

コンソール・プロンプト (P##>>>) で stop N と入力すると、プロセッサ N が停止してコンソール・モードに入ります。このコマンドを 1 次コンソール端末で実行すると、任意のパーティションの任意のプロセッサを停止することができます。たとえば、パーティション 1 の 1 次プロセッサがプロセッサ 4 の場合、次のコマンドにより、プロセッサ 4 がコンソール・モードに入ります。

P00>>>stop 4

P##>>>continue N

プロセッサ N が ctrl/p halt または stop N コマンドによってコンソール・モードに入った場合、P##>>> プロンプトで continue N と入力すると、プロセッサがプログラムの実行を再開します。たとえば次のようにします。

P##>>>continue 4

停止したプロセッサが 1 つの場合は、プロセッサ番号 (N) を省略できます。

P##>>>init

任意のパーティションのコンソール (P##>>>) プロンプトで、init と入力すると、システム全体が完全に再初期化されます。アクティブなパーティションがすべて即座に終了し、システムがリセットされます (電源スイッチを瞬間的に RESET の位置に動かしたときと同様です)。パーティションが有効な場合、コンソールは次のプロンプトを表示して、init コマンドの確認を要求します。

Do you really want to reset ALL partitions?(Y/<N>)

Y を入力して init コマンドを実行するか、N を入力して取り消します。

6.3.3 論理パーティションの構成とインストレーション・タスク

以降の各項では、AlphaServer GS140 システムのパーティショニングを行うための特定のタスクを説明します。それぞれのタスクは、示された順に実行しますが、場合によって省略できるタスクもあります。

この項を以前に読んでおり、スタートアップ・コマンドの通常の実行順を知りたいだけならば、ここにその要約を示します。

```
P00>>> set lp_count n
          (論理パーティションの数に n を設定します)
```

```
P00>>>init
          (1 次パーティションを初期化します)
```

```
P00>>>lpinit
          (各 2 次パーティションを開始します)
```

```
P00>>>boot
          (1 次パーティションをブートします)
```

```
P##>>>boot
          (各 2 次パーティションをブートします)
```

lpinit コマンドを省略すると、不適切な操作が行われます。コンソール・ファームウェアでは、これを防ぐために、lp_count がゼロ以外のときに、1 次パーティションのコンソール端末で boot コマンドが実行された場合、自動的に lpinit コマンドを実行するようにしています。

各 2 次パーティションは、スタートアップ時に構成情報を表示します。このメッセージには、6.3.3.8 項で説明しているように、連続する文字 Y が続く可能性があります。これはエラーではないので無視してかまいません。

6.3.3.1 システムのハードウェア構成の確認

ハードウェアが論理パーティショニングのために正しく構成されていることを確認する必要があります。また、後で (パーティションを構成するときに) 使用するために、ハードウェア構成についていくつかの情報を記録しておく必要があります。次の手順に従って、ハードウェア構成を確認します。

1. 電源の OFF/ENABLE スイッチを ENABLE の位置に設定することでシステムの電源を投入します。

注意

新たに設置されたシステム (出荷時にソフトウェアがインストール済み)、または既存のシステムで、auto_action コンソール環境変数が BOOT か RESTART に設定されているシステムは、ハードウェアの自己診断が完了すると、自動的にオペレーティング・システムをブートします。この場合は、コンソール端末で ctrl/c を入力して、自動ブートを中断する必要があります。自動ブートが中断できない場合は、オペレーティング・システムが完全にブートするのを待ってから、そのオペレーティング・システムをシャットダウンします (ctrl/p を入力して自動ブートを停止してはなりません)。論理パーティションのセットアップを試みる前に、出荷時にインストール済みのソフトウェアについての詳細を『インストールレーション・ガイド』で確認してください。

出荷時にインストール済みのソフトウェアのディスクは、いずれかのパーティションのシステム・ディスクとして使用されている場合があります。オペレーティング・システムのインストールについては 6.3.6 項を参照してください。

2. 約 15 秒後に、次の例のような構成情報が 1 次コンソール画面に表示されます。

```
F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
      A   A   M   .   M   P   P   P   P   TYP
      O   O   +   .   +   ++   ++   ++   ++   ST1
      .   .   .   .   .   EE   EE   EE   EB   BPD
      O   O   +   .   +   ++   ++   ++   ++   ST2
      .   .   .   .   .   EE   EE   EE   EB   BPD
      +   +   +   .   +   ++   ++   ++   ++   ST3
      .   .   .   .   .   EE   EE   EE   EB   BPD

      .   +   +   +   .   +   +   +   C0 PCI +
      .   .   +   .   .   +   +   C1 XMI +

      .   .   .   .   .   .   .   .   .   C4
      .   .   .   +   .   +   .   .   .   C5 PCI +
      .   .   .   .   .   .   .   .   .   C6
      .   .   .   +   .   +   +   +   .   C7 PCI +
      .   .   .   +   .   .   .   .   .   EISA +

      .   .   A1   .   A0   .   .   .   ILV
      .   .   1GB   .   1GB   .   .   .   2GB

Compaq AlphaServer GS140 8-6/525, Console V5.4 15-MAR-99 10:07:33
SROM V1.1, OpenVMS PALcode V1.48-3, Tru64 UNIX PALcode V1.45-3
System Serial = , OS = UNIX, 12:58:49 March 15, 1999
Configuring I/O adapters...
isp0, slot 0, bus 0, hose0
```



```

isp1, slot 1, bus 0, hose0
tulip0, slot 2, bus 0, hose0
isp2, slot 4, bus 0, hose0
isp3, slot 5, bus 0, hose0
tulip1, slot 6, bus 0, hose0
demna0, slot 1, bus 0, xmi0
kzmsa0, slot 2, bus 0, xmi0
kzmsa2, slot 5, bus 0, xmi0
kzpsa0, slot 3, bus 0, hose5
tulip2, slot 8, bus 0, hose5
tulip3, slot 9, bus 0, hose5
pfi0, slot 11, bus 0, hose5
tulip4, slot 12, bus 0, hose7
floppy0, slot 0, bus 1, hose7
kzpsa1, slot 4, bus 0, hose7
tulip5, slot 4, bus 2, hose7
tulip6, slot 5, bus 2, hose7
tulip7, slot 6, bus 2, hose7
tulip8, slot 7, bus 2, hose7
pfi1, slot 6, bus 0, hose7
pfi2, slot 8, bus 0, hose7
kzpsa2, slot 9, bus 0, hose7
P00>>>

```

3. NODE # で終わる行は、スロット番号を示します (構成手順のところで説明します)。システムには最高 9 つのスロットがあり、それぞれにスロット番号が付いています。次の行 (TYP で終わる行) は、各スロットのモジュールのタイプを示します。各スロットのモジュールのタイプを記録してください。

```

P = CPU (dual processor CPU module)
M = MEM (memory module)
A = IOP (IO port module)

```

8	7	6	5	4	3	2	1	0

4. 各パーティションにスロット (したがってモジュール) を割り当てることにより、システムを論理パーティションに分割します。各パーティションには、少なくともデュアル CPU モジュールが 1 つと MEM モジュールが 1 つ、および IOP モジュールが 1 つ割り当てられます。AlphaServer GS140 には全部で 9 つのスロットがあるので、最大 3 つのパーティションを構成することができます。

注意

1 つの CPU モジュールに、両方が同じパーティションに割り当てられている必要がある 2 つのプロセッサがあります。

5. システムが最低限の要件を満たしていれば、次の項に進みます。そうでない場合は、要件を整えてから (ハードウェアを追加するなどして)、次の項に進んでください。

6.3.3.2 ファームウェアのリビジョン・レベルの確認

論理パーティションには、コンソール・ファームウェアのサポートが必要です。『リリース・ノート』を参照して、リビジョン・レベルの最低要件が変わっていないことを確認してください。システムのファームウェアが論理パーティションをサポートしていることを確認するには、1 次コンソールで次のコマンドを使用して、ファームウェアのリビジョン・レベルを表示します。

```
P00>>>show version
```

コンソールに次のようなメッセージが表示されます。

```
version V5.4, 15-MAR-1999 10:07:33
```

ファームウェアがバージョン 5.4 以上であることを確認します。システムのファームウェアをアップグレードする必要がある場合には、ハードウェアのマニュアルでファームウェアのアップグレード手順を参照してください。ファームウェア CD-ROM が、ソフトウェア・キットに付属していますが、Web ページから、または ftp でファームウェアをダウンロードすることもできます。ファームウェアの検索と更新についての詳細は、『インストール・ガイド』を参照してください。

6.3.3.3 論理パーティションの構成

一連のコンソール環境変数 (EV) を使用して、論理パーティションを構成して有効 (または無効) にします。2 つのコンソール環境変数は、16 進数の形式を取ります。これはビットマスクで、マスク内のビット位置がモジュールまたはプロセッサ番号に対応します。ハードウェア構成規則では、次の基準に従い、モジュール・タイプに応じてモジュールが特定のスロット番号に設置されている必要があります。

- IO ポート (IOP) モジュールは、スロット 8, 7, および 6 に降順に設置されます。最高 3 つの IOP モジュールが設置可能です。

- CPU (デュアル・プロセッサ) モジュールは、スロット 0 から N に昇順に設置されます (N は、設置されている CPU モジュールの数に依存します)。N の値は、IOP および MEM モジュールの数によって制限されます。
- MEM (メモリ) モジュールは、一番大きな番号が付いた CPU モジュールと一番小さな番号が付いた IOP モジュールの間の、利用可能な任意のスロットに設置することができます。

プロセッサ・マスク変数 (lp_cpu_mask) は、数字の 3 を CPU モジュールのスロット番号の 2 倍だけ左にシフトして設定します。各スロットについて考えられる CPU マスクは、次のとおりです。

```

プロセッサ 00 と 01 (スロット 0):3 << (2 * 0) = 003
プロセッサ 02 と 03 (スロット 1):3 << (2 * 1) = 00c
プロセッサ 04 と 05 (スロット 2):3 << (2 * 2) = 030
プロセッサ 06 と 07 (スロット 4):3 << (2 * 4) = 0c0
プロセッサ 08 と 09 (スロット 5):3 << (2 * 5) = 300
プロセッサ 10 と 11 (スロット 6):3 << (2 * 6) = c00

```

lp_cpu_mask 変数の値は、個々の CPU モジュール・スロットのマスクの論理和をとることにより生成されます。たとえば、スロット 0 と 1 の CPU モジュールにある 4 つのプロセッサをパーティション 0 に割り当てる場合、lp_cpu_mask0 変数に値 00f を代入します。

I/O ポート・マスク変数 (lp_io_mask) は、数字の 1 を IOP モジュールのスロット番号だけ左にシフトして設定されます。各スロットについて考えられる IOP マスクは、次のとおりです。

```

スロット 8 の IO ポート・モジュール:1 << 8 = 100
スロット 7 の IO ポート・モジュール:1 << 7 = 080
スロット 6 の IO ポート・モジュール:1 << 6 = 040

```

パーティションに IOP モジュールが 2 つある場合、lp_io_mask 変数の値は、個々の IOP モジュール・スロットのマスクの論理和をとることにより生成されます。たとえば、スロット 7 と 8 の IOP モジュールをパーティション 1 に割り当てる場合、lp_io_mask1 変数の値は 180 になります。

IOP モジュールを 2 次パーティションに割り当てる場合、そのパーティションに割り当てられている IOP のうちの 1 つは、KFE72 オプションがインストールされた DWLPB オプションに接続されていなければならないことを忘れないでください。KFE72 オプションは、2 次パーティション用のコンソール・シリアル・ポートを提供します。

6.3.3.4 環境変数の選択と設定

論理パーティションのためのコンソール環境変数を作成するには、まず、パーティションの数と、各パーティションに割り当てられているスロット (すなわち、CPU、MEM、および IOP モジュール) を特定します (先に記録したモジュール・タイプとスロット番号を使用します)。特定できたら、コンソール環境変数を作成することができます。

コンソール環境変数および値の要約を次に示します。

コンソール環境変数	値
lp_count	パーティションの数
lp_cpu_mask N	パーティション N に対する CPU 割り当てマスク
lp_io_mask N	パーティション N に対する IOP モジュール割り当てマスク
lp_mem_mode	メモリ分離モード

次の表に、6.3.3.3 項の構成情報に基づいて、2 つのパーティションを構成する例とそのモジュールを示します。

- 4 CPU モジュール (スロット 0 から 3)
- 2 MEM モジュール (スロット 4 と 6)
- 2 IOP モジュール (スロット 7 と 8)

パーティション	モジュール
Partition 0	スロット 0 および 1 の CPU モジュール (CPU 0 ~ 3、マスク = 00F) スロット 8 の IOP モジュール (I/O Port、マスク = 100) スロット 6 の MEM モジュール (2GB メモリ)
Partition 1	スロット 2 および 3 の CPU モジュール (CPU 4 ~ 7、マスク = 0F0) スロット 7 の IOP モジュール (I/O Port、マスク = 080) スロット 4 の MEM モジュール (1GB メモリ)

メモリ用のコンソール環境変数マスクは存在しません。コンソール・ファームウェアは、メモリ・モジュールをパーティションに割り当てます。ファームウェアは、メモリ・モジュールをパーティションに割り当てます。ファームウェアは、メモリ・モジュールをパーティションに割り当てます。

ムウェアは、各パーティションに割り当てられるメモリの量の均衡を図ろうとします。

環境変数の作成または変更を行うには、コンソール・プロンプトで次のコマンドを実行します。ここで使用している値は、この項の最初に説明した、2つのパーティションの例に基づくものです。実際に入力する値は、ハードウェアの構成とパーティションのレイアウトに依存します。

lp_count 環境変数の値は 0 です (これは後で変更します)。

コンソール環境変数が作成されている場合、次のコマンドを入力するとそれらが表示されます。コンソール環境変数が存在しない場合は、何も出力されません。

```
P00>>>show lp*
```

コンソール環境変数が存在しない (あらかじめ作成していない) 場合、次のコマンドを使用して環境変数を作成します。

それぞれのコマンドを入力してからの処理は 10 秒間かかります。環境変数が作成されると、コンソールに各環境変数の値がエコーされます。

```
P00>>>create -nv lp_count 0
P00>>>create -nv lp_cpu_mask0 f
P00>>>create -nv lp_cpu_mask1 f0
P00>>>create -nv lp_io_mask0 100
P00>>>create -nv lp_io_mask1 80
P00>>>create -nv lp_mem_mode isolate
```

コンソール環境変数がすでに存在する (あらかじめ作成されている) 場合は、次のコマンドを使用して、それらの値を設定します。

```
P00>>>set lp_count 0
P00>>>set lp_cpu_mask0 f
P00>>>set lp_cpu_mask1 f0
P00>>>set lp_io_mask0 100
P00>>>set lp_io_mask1 80
P00>>>set lp_mem_mode isolate
```

次の 2 つの項の情報を使用して、コンソール環境変数の設定を表示 (および必要に応じて訂正) します。

6.3.3.5 コンソール環境変数の表示

コンソール環境変数の値を `show` コマンドを使用して、任意のパーティションのコンソールに表示できます。たとえば、`lp_count` の値を表示するには、次のように入力します。

```
P00>>>show lp_count
```

パーティショニング環境変数をすべて表示するには、次のコマンドを入力します。

```
P00>>>show lp*
```

コンソール環境変数が正しければ、次の項を無視して、6.3.3.7 項に進んでください。正しくない場合は、6.3.3.6 項に進んで必要な訂正を行います。

6.3.3.6 コンソール環境変数の訂正

注意

環境変数名の前に `lp_` が付いたコンソール環境変数名は、1 次パーティション (パーティション 0) のコンソールだけを使用して設定しなければなりません。これらの値は、2 次パーティションで変更してはなりません。

`set` コマンドを使用して、任意のまたはすべてのコンソール環境変数を変更します。たとえば、すべての環境変数を変更するには、次のコマンドを実行します。

```
P00>>>set lp_count 0
P00>>>set lp_cpu_mask0 f
P00>>>set lp_cpu_mask1 f0
P00>>>set lp_io_mask0 100
P00>>>set lp_io_mask1 80
P00>>>set lp_mem_mode isolate
```

6.3.3.7 自動ブート・リセットの無効化

Tru64 UNIX の『インストール・ガイド』では、`boot_reset` コンソール環境変数を ON に設定することを勧めています。しかし、この設定は `boot_reset` コンソール環境変数を OFF に設定する必要がある論理パーティションとは互換性がありません。これは、パーティションのブートが他の (以前にブートされている) パーティションの運用を妨げないようにするためです。`boot_reset` コンソール環境変数が ON に設定されていると、ブート・

コマンド (P00>>>boot) の実行後にシステム全体がリセットされます。このリセットは、全部のパーティションの運用を即座に終了します。

次のコマンドを実行して、boot_reset コンソール環境変数を無効にします。

```
P00>>>set boot_reset off
```

6.3.3.8 メモリ・インターリーブ・モードの設定

interleave コンソール環境変数の値を none に設定します。

```
P00>>>set interleave none  
P00>>>init
```

インターリーブ・モードを none に設定するときに、コンソール画面に文字 Y が連続的にエコーされることがあります (Y 文字が数行にわたることもあります)。この出力は無視してください。

6.3.3.9 オペレーティング・システム・タイプの UNIX への設定

os_type コンソール環境変数の値を UNIX に設定します。

```
P00>>>set os_type UNIX
```

6.3.3.10 auto_action コンソール環境変数の設定

(リセット・スイッチを使用して) POWER-ON または RESET にした後、プロセッサを停止するには、次のコマンドを使用します。

```
P00>>>set auto_action halt
```

POWER-ON または RESET の後、オペレーティング・システムを自動的にブートするには、次のコマンドを使用します。

```
P00>>>set auto_action boot
```

6.3.4 パーティションの初期化

Tru64 UNIX をパーティションにインストールする前に、パーティションを初期化する必要があります。この操作は、ハードウェア・リソース (CPU, IOP, および MEM モジュール) をそれぞれのパーティションに割り当て、2 次パーティションのそれぞれに対してコンソールを生成します。次の手順に従ってください。

1. lp_count 環境変数の値にパーティションの数を設定します。たとえば、2 つのパーティションを有効にする場合は、次のようになります。

```
P00>>>set lp_count 2
```

2. パーティション 0 を初期化します。

```
P00>>>init
```

構成情報が (前述のように) 1 次コンソール画面上に表示され、その後にコンソール・プロンプト (P00>>>) が表示されます。

3. 2 次パーティションをすべて初期化します。

```
P00>>>lpinit
```

1 次コンソール上には、各パーティションについて、物理メモリの開始アドレスなど、一連のパーティション構成メッセージが表示されます。メモリ構成を変更した場合に、カーネルの再構築が必要かどうかを判断できるようアドレスを記録します。

一般的なパーティション構成を次に示します。

```
Partition 0: Primary CPU = 0
Partition 1: Primary CPU = 4
Partition 0: Memory Base = 000000000    Size = 080000000
Partition 1: Memory Base = 080000000    Size = 040000000
No Shared Memory
LP Configuration Tree = 128000
starting cpu 4 in partition 1 at address 040010001
starting cpu 5 in partition 1 at address 040010001
starting cpu 6 in partition 1 at address 040010001
starting cpu 7 in partition 1 at address 040010001
```

構成された 2 次パーティションのそれぞれについて、2 次コンソール画面に情報が表示され、その後にコンソール・プロンプト (P04>>> など) が続きます。lpinit コマンドを入力した後、2 次コンソールに構成情報が表示されるまでに 20 秒かかります。

6.3.5 インターリーブ・モード・エラーの修正

interleave 環境変数の設定に誤りがあると、コンソールに次のエラー・メッセージが表示されます。

```
Insufficient memory interleave sets to partition system.
Issue command "set interleave none" then reset system.
```

このエラーから回復するには、次のコマンドを入力します。

```
P00>>>set interleave none
```

```
P00>>>set lp_count 0
```

```
P00>>>init
```


その後、この項の手順を繰り返します。

6.3.6 オペレーティング・システムのインストール

パーティションを構成して初期化したら、それぞれのパーティションにオペレーティング・システムをインストールすることができます。オペレーティング・システムのインストールは、『インストレーション・ガイド』の説明に従って行います。

AlphaServer GS140 システムは、Tru64 UNIX がディスクの 1 つにプリインストールされた状態で出荷されます。このディスクは、パーティションのいずれか (通常、パーティション 0) のルート・ディスクとして使用することができます。プリインストール・ディスクを使用するには、それをブートしてから、インストレーションを完了するための手順に従います。省略時の設定では、`bootdef_dev` コンソール環境変数はプリインストール・ディスクを自動的にブートするように設定されています。そうでない場合は、6.3.3.1 項で記録した `bootdef_dev` の値を使用します。

注意

IOP モジュールの割り当て方によっては、FIS (出荷時インストール済みソフトウェア) ディスクの名前が変わり、パーティション 0 に割り当てられないこともあります。それぞれのパーティションで次のコマンドを使用して、ディスクの位置を確認します。

```
P##>>> show device
```

オペレーティング・システムは、CD-ROM から、またはネットワーク上のリモート・インストレーション・サーバ (RIS) からインストールすることもできます。すべてのパーティションに CD-ROM ドライブを構成することが現実には困難な場合や、RIS サーバが使用できないことがあります。ローカル・ネットワークが使用できる場合は、別の方法として、オペレーティング・システムを CD-ROM から 1 つのパーティションにインストールして、次にそのパーティションを、他のパーティションのための RIS サーバとして構成するという方法があります。リモート・インストレーション・サーバのセットアップについては、『*Sharing Software on a Local Area Network*』を参照してください。

6.3.7 パーティショニングされたシステムの管理

それぞれのパーティションで実行されているオペレーティング・システムは、あたかもパーティショニングされていないシステム上で実行されているかのように管理することができます。ただし、パーティショニングされたシステムを管理するときに、認識し、考慮すべき AlphaServer GS140 固有の運用特性がいくつか存在します。以降の各節では、これらのトピックについて説明します。

6.3.7.1 運用特性

パーティショニングされたシステムの通常の運用においては、一部の構成タスクおよび初期化タスクを繰り返さなければならないことがあります。これらのタスクには、パーティション間の干渉を防止するために、特別の予防措置が必要な場合があります。次の項では、これらのタスクについて説明します。

6.3.7.1.1 コンソール `init` コマンド (`P##>>>init`)

任意のパーティションのコンソール・プロンプトで `init` コマンドを入力すると、システム全体が初期化されます。これは、すべてのパーティションのオペレーティング・システムを即座に終了します。したがって、システム全体を再初期化する必要がない限り、`init` コマンドを実行してはなりません。

`init` コマンドを実行すると、すべてのパーティションを本当にリセットするかどうかを確認するプロンプトがコンソールに表示されます。`init` コマンドを取り消す場合は `no`、`init` コマンドを続行する場合は、`yes` と答えます。

6.3.7.1.2 オペレーティング・システムのシャットダウンとリブート

パーティションで実行されているオペレーティング・システムをシャットダウンし、コンソール・モード (`P##>>>` プロンプト) に戻るには、`shutdown` コマンドを使用します。たとえば次のようにします。

```
# /usr/sbin/shutdown -h +5 "Shutting down the OS"
```

`shutdown` コマンドはまた、オペレーティング・システムをシャットダウンしてからリブートすることもできます。たとえば次のようにします。

```
# /usr/sbin/shutdown -r +5 "Rebooting the OS"
```

6.3.7.2 中断されたオペレーティング・システムのブートの回復

オペレーティング・システムのブートが完了しなかった場合、または中断された場合は、コンソール・ブート・ドライバが矛盾した状態になります。この場合、コンソールには、次のメッセージが表示されます。

```
Inconsistent boot driver state.  
System is configured with multiple partitions.  
A complete INIT must be performed before rebooting.
```

次の手順を使用して、この状態から回復します。

1. 実行しているすべてのパーティションのオペレーティング・システムをシャットダウンします。
2. 1 次コンソールで、次のコマンドを実行します。

```
P00>>>set lp_count 0  
P00>>>init  
P00>>>set lp_count N
```

(ここで、N はパーティション数です。)

```
P00>>>init  
P00>>>lpinit
```

3. 各パーティションでオペレーティング・システムをブートします。たとえば次のようにします。

```
P00>>>boot  
P04>>>boot
```

6.3.7.3 プロセッサの停止

通常の運用状態では、プロセッサを作業で停止する必要はありません。オペレーティング・システムがシャットダウンされると、プロセッサは停止してコンソール・モードに入ります。何らかの理由で、オペレーティング・システムがハングした場合 (たとえば、ロード可能なデバイス・ドライバのデバッグ中) には、プロセッサを手作業で停止する必要があります。

注意

プロセッサを停止できないことはまれですが、停止できない場合には、4 点スイッチの OFF/ENABLE スイッチを一瞬 RESET の位置に合わせ、システムをリセットしてから、放す必要があります。

次の手順は、電源 OFF/ENABLE スイッチが ENABLE の位置にある場合のみ機能します。

1 次パーティション

1 次コンソール端末で Ctrl/p を入力すると、1 次プロセッサがコンソール・モードに入り、P##>>> プロンプトが表示されます。stop *N* コマンド (*N* はプロセッサ番号) を使用して、2 次プロセッサを停止することができます (ただし、通常、これは必要ありません)。コンソール・プロンプトと stop コマンドの定義についての詳細は、6.3.2.1 項を参照してください。

2 次パーティション

2 次パーティションは、2 次コンソール端末で Ctrl/p コマンドを実行しても停止しません。2 次パーティションをコンソール・モードにするには、次の手順に従います。

1. 1 次パーティションでオペレーティング・システムをシャットダウンします。

```
# /usr/sbin/shutdown -h +5 "Shutting down the OS"
```

2. 2 次パーティションの 1 次プロセッサを停止します。

```
P00>>>stop N
```

ここで *N* は、2 次パーティションの 1 次プロセッサの CPU 番号です (通常、2 次パーティションに割り当てられた CPU のうち、最も番号の小さいもの)。たとえば次のようにします。

```
P00>>>stop 4
```

6.3.7.4 電源 OFF/ENABLE スイッチの位置

通常のシステム運用では、電源 OFF/ENABLE スイッチは SECURE の位置になければなりません。これにより、Ctrl/p によって、プロセッサを誤って停止することを防げます。

6.3.7.5 コンソール環境変数の変更によるパーティションの再構成

論理パーティションを制御するコンソール環境変数 (名前が lp_ で始まるもの) は、2 次パーティションでは変更してはなりません。これらのコンソール環境変数は、すべてのパーティションをシャットダウンし、1 次パーティションのコンソール端末で新しい値を設定することによってのみ変更できます。

新しいパーティションのレイアウトを決定したら、次の手順に従って、パーティションを再構成します。

1. それぞれのパーティションでオペレーティング・システムをシャットダウンします。

```
# /usr/sbin/shutdown -h +5 "Shutting down to  
reconfigure partitions"
```

2. パーティションを無効にし、システムをリセットします。

```
P00>>>set lp_count 0  
P00>>>init
```

3. コンソール `set` コマンドを使用して、任意の、またはすべてのコンソール環境変数を変更します。6.3.3.4 項で説明したパーティションが2つ場合の例では、次のコマンドを使用します。

```
P00>>>set lp_count 2  
P00>>>set lp_cpu_mask0 f  
P00>>>set lp_cpu_mask1 f0  
P00>>>set lp_io_mask0 100  
P00>>>set lp_io_mask1 80  
P00>>>set lp_mem_mode isolate
```

4. 1 次パーティションを初期化します。

```
P00>>>init
```

5. 2 次パーティションをすべて初期化します。

```
P00>>>lpinit
```

6. 次のようなコマンドを使用して、それぞれのパーティションでオペレーティング・システムをブートします。

```
P00>>>boot  
P04>>>boot
```

6.3.7.6 ブート前の他のコンソール環境変数の確認

それぞれのパーティションでオペレーティング・システムをブートする前に、コンソール `show` コマンドを使用して、コンソール環境変数が正しい状態にあることを確認します。

```
P0##>>>show boot_reset
```

`boot_reset` 環境変数は、`off` でなければなりません。

```
P0##>>>show interleave
```

`interleave` 環境変数は、`none` でなければなりません。

```
P0##>>>show auto_action
```

auto_action 環境変数は、HALT または BOOT に設定します。

```
P0##>>>show os_type
```

os_type 環境変数は UNIX に設定しなければなりません。

6.3.7.7 ブート時の論理パーティショニング情報メッセージ

論理パーティションを構成して有効にすると、各パーティションの情報メッセージがオペレーティング・システムによって表示されます。これらのメッセージは、ブートストラップ処理の初期段階にコンソール端末に表示されます。次に、パーティションが2つあるシステムで通常見られる例を示します。

```
Partition 0
-----
LP_INFO: 2 partition(s) established via lp_count
LP_INFO: primary processor for partition 0 is CPU 0
LP_INFO: partition 0 CPU allocation mask = 0xf
LP_INFO: partition 0 IOP allocation mask = 0x100
LP_INFO: Memory partitioning mode set to isolate
LP_INFO: partition 0 memory starting address = 0x0

Partition 1
-----
LP_INFO: 2 partition(s) established via lp_count
LP_INFO: primary processor for partition 1 is CPU 4
LP_INFO: partition 1 CPU allocation mask = 0xf0
LP_INFO: partition 1 IOP allocation mask = 0x80
LP_INFO: Memory partitioning mode set to isolate
LP_INFO: partition 1 memory starting address = 0x80000000
```

これらのメッセージは、次の情報を示します。

- アクティブなパーティションの数
- 現在のパーティションの1次プロセッサの数
- 現在のパーティションに割り当てられているプロセッサ
- 現在のパーティションに割り当てられている I/O ポート・モジュール
- メモリのパーティショニング・モード (常に isolate でなければならない)
- 現在のパーティションのメモリ開始アドレス

6.3.8 ハードウェアの管理と保守

AlphaServer GS140 では、パーティションは共通の物理格納装置とハードウェア (たとえば、電源、システム・バス、および制御パネルの電源スイッチ) を共用します。次のハードウェア管理および保守タスクは、個々のパーティ

ションで行うことはできません。パーティションを無効にし、システムをリセットして、パーティショニングされていない状態にする必要があります。

完全なシステム再初期化を必要とするタスクは、次のとおりです。

- システム・ハードウェアに対する修正保守と予防的保守
- I/O コントローラのファームウェア・アップグレードをはじめとする AlphaServer GS140 ファームウェア・アップグレードのインストール
- システムのハードウェア・コンポーネント (CPU, メモリ, IOP, PCI バス, I/O コントローラ, および I/O デバイス [スワップ可能ディスクを除く]) の追加と削除
- 名前の前に lp_ が付くコンソール環境変数を変更することによる、パーティションのハードウェア・リソース割り当ての変更
- フロッピー・ディスク・ドライブからの ECU (Eisa Configuration Utility) または RCU (Raid Configuration Utility) の実行

6.3.8.1 技術サポートの利用

特定の問題について技術サポート部門に連絡する必要がある場合、システムがパーティショニングされていることを顧客サービスの担当者に知らせることが重要です (特にサービス操作でリモート診断を使う場合)。サービス・コールをかけるとき、システムが論理パーティションを使用していることを、はっきりと教えてください。

論理パーティショニング・ソフトウェアは、顧客サービス担当者が、システムがパーティショニングされているかどうかを判断するための2つの方法を提供します。オペレーティング・システムのスタートアップ時に出力される LP_INFO メッセージは、スタートアップ ASCII メッセージの一部として、バイナリ・エラー・ログにも記録されます。オペレーティング・システムの任意のインスタンスで `sizer -P` コマンドを実行し、システムのパーティショニング状態を表示することができます。

```
# sizer -P
Host hostname is instance 1 of 2 partitions.
Physical memory starts at address 0x80000000.
Memory mode is isolate.
Processors assigned to instance 1: 4 5 6 7
IO Port (s) assigned to instance 1: slot 7
```

システムがパーティショニングされていない場合、次のメッセージが表示されます。*hostname* は、システム名です。

```
Host hostname is not partitioned.
```

6.3.8.2 ハードウェア管理と保守タスクの実行

管理や保守のタスクを行う前に、すべてのパーティションの運用を停止し、システムをパーティショニングされていない状態に戻す必要があります。次の手順を使用して、パーティションをシャットダウンします。

1. それぞれのパーティションでオペレーティング・システムをシャットダウンします。

```
# /usr/sbin/shutdown -h +5 "Shutting down for maintenance"
```

2. 1 次コンソール端末で次のコマンドを実行して、パーティションを無効にします。

```
P00>>>set lp_count 0
```

3. 1 次パーティションの *auto_action* コンソール環境変数を HALT に設定します。

```
P00>>>set auto_action halt
```

次の手順の 1 で *auto_action* 環境変数をリセットして、パーティションを初期化して、リブートしなければならない場合があります。

4. 1 次コンソール端末で次のコマンドを入力し、システムを再初期化します。

```
P00>>>init
```

システムが *P00>>>* プロンプトに戻ったら、システム管理および保守タスクを実行することができます。システム管理および保守タスクが完了したら、次の手順を使用してパーティションを再初期化し、パーティションをリブートします。

1. コンソール環境変数が正しい値に設定されていることを確認します。

```
P00>>>show lp*
P00>>>show boot_reset
P00>>>show interleave
P00>>>show auto_action
```


boot_reset 環境変数が off に設定され、interleave 環境変数が none に、auto_action 環境変数が HALT または BOOT に設定されていなければなりません。

2. lp_count 環境変数をパーティションの正しい数に設定します。たとえば次のようにします。

```
P00>>>set lp_count 2
```

3. 1 次パーティションを初期化します。

```
P00>>>>init
```

4. 2 次パーティションをすべて初期化します。

```
P00>>>lpinit
```

5. それぞれのパーティションでオペレーティング・システムをブートします。システムのハードウェア構成を変更したり、ハードウェア・リソースを別のパーティションに割り当て直したりした場合、カーネルの再構築が必要になることがあります。パーティションについてカーネルの再構築が必要かどうかを判断するには、6.3.9 項の手順を使用します。

パーティションでカーネルの再構築が必要でない場合、オペレーティング・システムをブートします。

```
P##>>>boot
```

ここで ## は、パーティションの 1 次プロセッサの CPU 番号です。

6.3.9 UNIX カーネルの再構築が必要なハードウェア変更

システムのハードウェア構成を変更した場合、カーネルの再構築が必要になる場合があります。次の表に、カーネルの再構築が必要な、ハードウェア構成の変更を示します。

変更点	要件
プロセッサ – CPU モジュールの追加，削除，または再割り当て	任意のパーティションの lp_cpu_mask# 環境変数を変更した場合，カーネルの再構築は必要ない。デュアル CPU モジュールにある，両方のプロセッサを同じパーティションに割り当てる必要がある。

I/O プロセッサ – IOP モジュールの追加，削除，または再割り当て

IOP モジュールを追加または削除した場合，カーネルを再構築する。変更パーティションのカーネルだけを再構築すればよい。IOP モジュールをパーティション間で移動した場合は，両方のパーティションでカーネルの再構築が必要である。lp_io_mask# 環境変数は，IOP モジュールを割り当てる。

I/O バスおよび I/O コントローラを追加または削除した場合，影響のあるパーティションについて，カーネルの再構築が必要になる。

メモリ・モジュール – メモリ・モジュール構成の変更

1 次パーティション (パーティション 0) では，メモリ・モジュール構成を変更した場合，カーネルの再構築は必要ない。

2 次パーティションのカーネルは，特定のメモリ・アドレス (すなわち，そのパーティションの物理メモリ開始アドレス) で実行されるように構築する必要がある。特定のタイプのメモリ再構成を行うと，このアドレスが変わるので，カーネルの再構築が必要になる。パーティションのメモリ・サイズが増減すると，それより大きい番号のパーティションのメモリ開始アドレスがすべて変わる。

たとえば，パーティション 0 にある 2GB のメモリ・モジュールを 4GB のメモリに交換した場合，パーティション 1 のメモリ開始アドレスは 2GB 大きくなる。この例では，カーネルの再構築が必要である。

メモリ・モジュール構成を変更した後に，2 次パーティションのカーネルがブートに失敗した場合，カーネルを再構築する必要がある。

各パーティションのメモリ開始アドレスは，
P00>>>lpinit コマンドを実行するたびに 1 次コンソールに表示される。

6.3.9.1 パーティションごとの UNIX カーネルの再構築方法

次の手順は，カーネルを再構築する方法の説明です。これは，『システム管理ガイド』に記載されている通常のカーネルの構築方法の特殊なケースです。この手順では，6.3.4 項に説明されているとおりにパーティションが初期化され，カーネルの再構築を必要とするパーティションが，P##>>> コン

ソール・プロンプトを表示して停止していると想定します。以下の詳細については、『システム管理ガイド』のカーネル構成情報を参照してください。

- カーネルのブートおよびシングルユーザ・モード・プロンプト
- カーネルの保存とコピー

1. 汎用カーネルをシングルユーザ・モードでブートします。

```
P##>>>boot -fl s -fi genvmunix
```

2. ファイル・システムを検査してマウントします。

```
# /sbin/bcheckrc
```

ファイル・システムのマウントについての情報は、『システム管理ガイド』を参照してください。

3. このパーティションにホスト名 (システム名) を設定します。

```
# hostname NAME
```

4. カーネルを再構築します。

```
# doconfig
```

注意

doconfig に `-c` オプションを指定してカーネルを再構築してはなりません。

5. 現在のカーネルを保存します。

```
# cp /vmunix /vmunix.save
```

6. 新しいカーネルをインストールします。ここで、`SYSNAME` は、ローカル・ホスト名です。

```
# cp /sys/SYSNAME/vmunix /vmunix
```

7. ファイル・システムをアンマウントします。

```
# umount -a
```

8. オペレーティング・システムを停止します。

```
# sync  
# sync  
# halt
```

9. 新しいカーネルをブートします。

```
P##>>>boot
```

6.3.10 回復不能なハードウェア・エラーのマシン・チェックへの対応

ハードウェア・エラーは、次の2つに大別されます。

- 回復可能なエラーは、ハードウェアによって是正され、オペレーティング・システムに報告されます。オペレーティング・システムは、回復可能なエラーはバイナリ・エラー・ログに記録し、通常のシステム運用を続行します。回復不能なハードウェア・エラーでは、通常のシステム運用を即座に終了し、何らかの形の是正措置(システム・リセットなど)を行う必要があります。
- 回復不能なハードウェア・エラーは、マシン・チェックとしてオペレーティング・システムに報告されます。オペレーティング・システムは、次のようなパニック・メッセージを表示してクラッシュします。

```
panic (cpu 0):tlaser: \  
MACHINE CHECK Non-recoverable hardware error
```

次にシステムはクラッシュ・ダンプを書き出し、リブートするか停止します(auto_action コンソール環境変数の設定が BOOT か HALT によって異なります)。

ハードウェア・エラーの中には、オペレーティング・システムをリブートするために、完全にシステムをリセットしなければならないエラーもあります。

システム全体にかかわるハードウェア障害については、オペレーティング・システムがクラッシュ・ダンプを書き出した後に、強制システム・リセットを行います。リセットが完了した後で、auto_action が BOOT に設定されていれば、コンソール・ファームウェアが自動的にすべてのパーティションを再初期化します。次のコマンドを使用して、それぞれのパーティションでオペレーティング・システムをブートします。

```
P00>>>boot  
P##>>>boot
```

それ以外の場合、システムは停止し、コンソール・モード(P00>>> プロンプト)に入ります。これが発生した場合、次のコマンドを入力して、パーティションを再起動し、オペレーティング・システムをリブートします(ここで *N* は、パーティションの数です)。

```
P00>>>set lp_count N  
P00>>>init  
P00>>>lpinit
```

P00>>>boot

2 次パーティションについては、boot コマンドを入力します。

P##>>>boot

ローカル (パーティションの中に存在する) のハードウェア障害については、影響を受けるパーティションで実行されているオペレーティング・システムは、クラッシュ・ダンプを書き出した後に、無条件で停止します。これにより、シャットダウンが可能になるまで、他のパーティションは稼働し続けることができます。影響を受けるパーティションを再起動するには、次の手順を使用して、完全なシステム・リセットを行う必要があります。

1. 稼働中のそれぞれのパーティションでオペレーティング・システムをシャットダウンします。

```
# /usr/sbin/shutdown -h +5 "Shutting down for error recovery"
```

2. 1 次コンソール端末で、次のコマンドを入力します。

```
P00>>>set lp_count 0
P00>>>init
```

3. 次のプロンプトがコンソールに表示されます。

```
Do you really want to reset ALL partitions?(Y/<N>)
```

Y を押して、リセットを実行します。リセットが完了し、auto_action が BOOT に設定されていれば、コンソール・ファームウェアがすべてのパーティションを自動的に再初期化します。

4. 次のコマンドを使用して、それぞれのパーティションでオペレーティング・システムをブートします。

```
P00>>>boot
P##>>>boot
```

そうでない場合は、次のコマンドを入力します (N はパーティションの数です)。

```
P00>>>set lp_count N
P00>>>init
P00>>>lpinit
P00>>>boot
```

それぞれの 2 次パーティションについて、次のコマンドを入力します。

```
P##>>>boot
```

これらの回復手順で、すべてのパーティションの完全なシステム運用を再開できない場合、OFF/ENABLE スイッチを一瞬だけ RESET の位置に動かしてから放すことで、システムを手動でリセットします。リセットが完了したら、回復手順を再度実行します。それでも障害が起こる場合は、技術サポート部門に連絡してください。

6.3.11 論理パーティショニングのエラー・メッセージ

エラー状態（たとえば、無効なパーティション構成）が生じた場合、エラー・メッセージがパーティションのコンソール端末に表示されます。現在のパーティションの 1 次プロセッサは、エラー・メッセージを表示した後に停止し、コンソール・プロンプトに戻ります。このようなエラーから回復するには、論理パーティショニング・コンソール環境変数を修正して、パーティションをリブートします。

表示されるのは、次のようなエラー・メッセージです。

```
LP_ERROR:invalid partition count (lp_count = #, max nodes = #)
```

lp_count コンソール環境変数の設定が誤っています。値が 0 より小さいか、AlphaServer GS140 がサポートするパーティションの最大数を超えています。

```
LP_ERROR:no CPUs for partition (check lp_cpu_mask)
```

lp_cpu_mask# (# は、現在のパーティション番号を表す) の値が、誤って設定されています。このパーティションにプロセッサが割り当てられていません。

```
LP_ERROR:no IOP for partition (check lp_io_mask)
```

lp_io_mask# (# は、現在のパーティション番号を表す) の値が、誤って設定されています。このパーティションに I/O ポート・モジュールが割り当てられていません。

```
LP_ERROR:lp_count > 1, but partitions not initialized  
Please execute 'lpinit' command at >>> prompt
```

このメッセージは、パーティションは構成されているが初期化されていないことを示します。

LP_ERROR:must set lp_mem_mode [share or isolate]

lp_mem_mode コンソール環境変数が設定されていないか、設定が誤っています。論理パーティションの場合、lp_mem_mode を isolate に設定しなければなりません。

Bootstrap address collision, image loading aborted

カーネルのリンク・アドレスがパーティションのメモリ開始アドレスに一致しません。このエラーからの回復方法については、6.3.9 項を参照してください。

6.3.12 コンソール・ファームウェアのエラーまたは情報メッセージ

コンソール・ファームウェアは、特定のイベント(システム・リセットやパーティションのスタートアップなど)についての安全チェックをいくつか実装しています。これらのチェックにより、パーティション間の干渉が防止されます。異常が検出されると、次のいずれかのメッセージがそのパーティションのコンソールに表示されます。

Do you really want to reset ALL partitions?(Y/<N>)

このメッセージは、init コマンドの実行、または boot_reset コンソール環境変数が ON であることによってブートした結果、システム・リセットが要求された後に表示されます。このメッセージは、操作を続行すると、すべてのパーティションの運用が停止され、システムがリセットされると警告するものです。リセットが必要な場合、稼働中のパーティションのオペレーティング・システムをすべてシャットダウンしてから、リセットを行います。

Auto-Starting secondary partitions...

このメッセージは、コンソール・ファームウェアが論理パーティションを初期化していることを示します(自動的に lpinit コマンドを実行します)。自動起動は、システム・リセット(または電源投入)の後に発生します。コンソール・ファームウェアは、以下の条件が満たされていると、すべてのパーティションでオペレーティング・システムをブートします。

- auto_action コンソール環境変数が BOOT に設定されている
- 電源投入時に RESET スイッチを使用して(init コマンドによってではなく)リセットを開始する

```
Insufficient memory interleave sets to partition system.  
Issue command "set interleave none" then reset system.
```

このメッセージは、interleave コンソール環境変数の設定が誤っていることを示します。設定を none に変更します。

```
Insufficient memory modules to partition system.
```

各パーティションには、専用のメモリ・モジュールが必要です。パーティションの数を減らすか、パーティションごとにメモリ・モジュールをインストールします。

このメッセージは、lp_count コンソール環境変数が正しく設定されていない可能性があることを示します。たとえば、パーティションが2つあるのに対し、lp_count が4に設定されている場合などです。この場合、実際のパーティションの数に合うように lp_count を設定します。

```
Inconsistent boot driver state.  
System is configured with multiple partitions.  
A complete INIT must be performed before rebooting.
```

オペレーティング・システムのブートが不完全または中断されたために、コンソール・ブート・ドライバが矛盾した状態になっています。この状態からの回復方法についての詳細は、6.3.7.2 項を参照してください。

```
Do you want to attempt to boot secondary partitions  
anyway? (Y/<N>).
```

このメッセージは、(おそらく lp_ コンソール環境変数の設定が誤っているために) コンソールがパーティションの設定の中に矛盾を検出したことを示します。操作を続行しても安全であることが確かな場合を除き、この質問には N と答え、矛盾を解決する必要があります。

```
TIOP # not configured in any partition.  
Non-existent TIOP # configured in a partition.
```

これらのメッセージ (一緒に、または個別に) は、lp_io_mask# コンソール環境変数の設定が誤っていることを示します。マスクが0または誤った IOP モジュール・スロット番号に設定されている可能性があります。設定を修正して、lpinit コマンドを再実行してください。

```
Secondary partitions have already been started.
```

このメッセージは、パーティションをスタートした後に、2度目の lpinit コマンドを実行したことを示します。オペレーティング・シ

システムをブートする前に、lp_ コンソール環境変数の値を確認する必要があります。

```
CPU # not configured in any partition.  
No valid primary processor specified for partition #.
```

このメッセージでは、CPU 番号 (#) は 1 つの CPU であることも、CPU のリストであることもあります。

これらのメッセージ (一緒に、または個別に) は、lp_cpu_mask# コンソール環境変数の設定が誤っていることを示します。マスクが 0 または誤った CPU 番号に設定されている場合もあります。設定を修正して、lpinit コマンドを再実行してください。

6.4 AlphaServer 1000 および 1000A の構成情報

以下の構成上の制限事項は、AlphaServer 1000 と AlphaServer 1000A システムにのみ適用されます。

6.4.1 EISA Configuration Utility Version 1.10

この注意事項は、埋め込み Cirrus VGA グラフィックス・コントローラのユーザに適用されます。

ECU (EISA Configuration Utility) のバージョン 1.10 を実行するとき、VGA グラフィックス・コントローラの省略時の設定は Disabled です。以前のバージョンでは、省略時の設定は Enabled でした。

ECU の以前のバージョンによって設定されているシステムで、ECU バージョン 1.10 を初めて実行するときは、埋め込み VGA グラフィックス・コントローラの設定は、自動的に Disabled に設定されます。省略時の値を変更するには、ECU を実行し、ステップ 3 を選択します。詳細を表示して編集し、終了する前に VGA グラフィックス・コントローラを Enabled に設定します。オペレーティング・システムのブートに先立って VGA グラフィックス・コントローラを Enabled に設定しないと、X サーバが起動せず、オペレーティング・システムをブートしたときには、システムでサポートされるのは汎用コンソールとなります。

6.4.2 グラフィックス解像度

1 MB のビデオ RAM を搭載した組み込みの Cirrus ビデオを内蔵する AlphaServer 1000A システムの省略時のグラフィックス解像度は、1024 ×

768 です。オプションの 512 KB ビデオ RAM がない場合、オペレーティング・システムは 640 × 480 (省略時) または 800 × 600 の解像度だけをサポートします。

512 MB のビデオ RAM を搭載した組み込みの Cirrus ビデオを内蔵する AlphaServer 1000 システムの省略時のグラフィックス解像度は、640 × 480 です。この構成は 800 × 600 の解像度もサポートします。

800 × 600 の解像度を使用するには、`/usr/lib/X11/xdm/Xservers` ファイルで次の行を編集します。

```
:0 local /usr/bin/X11/X
```

この行を次のように変更します。

```
:0 local /usr/bin/X11/X "-screen0 800x600"
```

CDE セッション・マネージャに 800 × 600 の解像度を使用するには、`/usr/dt/config/Xservers` と `Xservers.conf` ファイルの次の行を編集します。

```
:0 Local local@console /usr/bin/X11/X :0
```

この行を次のように変更します。

```
:0 Local local@console /usr/bin/X11/X :0 -screen0 800
```

XDM または CDE 用にこれらのファイルを編集する前に、使用するシステム・モニタが 800 × 600 の解像度をサポートすることを確認してください。

6.5 AlphaServer GS シリーズの構成情報

以下の構成上の制限事項は、AlphaServer GS システムにのみ適用されます。

6.5.1 1 次 CPU での OLAR エラーの可能性

オペレーティング・システムのバージョン 5.1A 以降、AlphaServer GS80、GS160、および GS320 システムで、1 次 CPU をオフラインにして取り外すことができるようになっていました。1 次 CPU をオフラインにすると、1 次 CPU の役割が自動的に別の CPU に割り当てられます。この操作によって問題が断続的に派生する可能性があるため、現時点では 1 次 CPU をオフラインにしないでください。1 次 CPU は通常、CPU0 です。どの CPU が 1 次 CPU になっているかを確認するには、`pset_info` コマンドを使います。

```
# pset_info
number of processor sets on system = 1
```

```
pset_id # cpus # pids # threads load_av created
0 4 89 453 0.13 07/12/2001 17:25:28
```

```
total number of processors on system = 4
```

```
cpu # running primary_cpu pset_id assigned_to_pset
0 1 1 0 07/12/2001 17:25:28
1 1 0 0 07/12/2001 17:25:28
2 1 0 0 07/12/2001 17:25:28
3 1 0 0 07/12/2001 17:25:28
```

この出力は、CPU0 が 1 次 CPU であることを示します。

1 次 CPU をオフラインにすると、操作はおそらく正常に行われ、新しい 1 次 CPU (たとえば、CPU1 など) が自動的に選択されます。しかし、1 次 CPU だった CPU を再びオンラインにしようとしたとき、次のようなエラーが発生することがあります。

```
Processor X failed to start
console callback PARTITION, POWER-HW timed_out.
wf_hal_pwr_ctl: call to prom_power failed with status [ffffffe6]
```

この状態から回復するには、SRM コンソールを使ってシステムを初期化する必要があります。それには、オペレーティング・システムをシャットダウンしなければなりません。この問題は、コンソール・ファームウェアの将来のリリースで修正されます。

6.5.2 CPU の電源切断と投入を連続的に繰り返さないこと

連続的に実行されるテスト・ループで OLAR 管理コマンドを使用すると、CPU の信頼性低下を招く可能性があります。OLAR 管理コマンドは、CPU モジュールへの DC 電力供給を断ちます。シェル・スクリプトを使って CPU の電源切断と投入を連続的に繰り返すことはしないようにしてください。

CPU モジュールの DC/DC コンバータは、電源切断と投入の上限が 1000 回まで規定されています。CPU の電源切断と投入の回数がこの値を超えないようにしてください。

6.5.3 ホット追加に関する制限事項

本リリースのオペレーティング・システムは、GS80、GS160、および GS320 CPU のホット追加をサポートします。ただし、次の制限があります。メモリがなく、かつ、少なくとも 1 つの CPU がない状態でブートされた Quad

Building Block (QBB) には、CPU をホット追加できません。追加した場合、システム・パニックが発生します。メモリがあり、少なくとも 1 つの CPU がある状態でブートされた QBB には、追加 CPU を必要に応じてホット追加できます (この制限は、カーネルの将来のアップデートではなくなります)。

6.6 Personal Workstation 433au , 500au , および 600au システム

以下の構成上の制限事項は、Personal Workstation クラスのシステムにのみ適用されます。

6.6.1 64ビット PCI オプション・カード

64 ビット PCI スロットであるスロット 4 と 5 は、『*Systems and Options Catalog*』に記載のカードのみを対象としています。これらのスロット (下記 *n*) のいずれかに、サポートされていないカードが接続されていると、コンソール・ファームウェアによってシステム運用が禁止され、次のエラーが表示されます。

```
Illegal device detected on primary bus in physical slot n  
Power down the system and remove the unsupported  
device from slot n
```

6.6.2 省略時のキーボード・マップの誤り

Personal Workstation 433au , 500au , または 600au クラスのシステムで PCXLA-NA キーボードを使用する場合、正しいキーマップを使うようにキーボード・ドライバを再構成しない限り、キーは正しくマップされません。

キーボード・ドライバを再構成するには、次のコマンドを実行します。

```
# sysconfig -r gpc_input kbd_scancode=2
```

または、sysconfigdb コマンドを使って、次のエントリを /etc/sysconfigtab ファイルに追加できます。

```
gpc_input:  
kbd_scancode = 2
```

sysconfig コマンドを使用してドライバを再構成した場合には、システムをリブートするたびにコマンドを実行する必要があります。sysconfigdb ユーティリティを使用した場合、変更した情報がリブートにまたがって維持されるため、ユーザによる他の操作は必要ありません。

個々のデバイス・タイプの管理

デバイスをインストールして構成するために必要な情報のほとんどは、デバイスに添付されているオーナーズ・マニュアルに記載されています。オーナーズ・マニュアルには、デバイスのインストール手順のほか、Tru64 UNIX オペレーティング・システムのリリースに対応する特別な構成要件や使用上の制限事項などが指定されています。この情報に関しては多くの場合、オペレーティング・システムのリリースごとの『リリース・ノート』に追加の情報が記載されています。

『リリース・ノート』には、デバイスの使用にあたって知っておくべき既知の問題や一時的な制限についての説明があります。本章では、構成オプション、永続的な構成上の要件、使用上の制限事項に関する情報を提供し、次の項目について取り上げます。

- ストレージの管理 (7.1 節)
- ホスト・バス・アダプタの管理 (7.2 節)
- グラフィックス・アダプタの管理 (7.3 節)

注意

本章では、特定のデバイス・オプションとオペレーティング・システム・オプションの構成方法を説明します。これらのデバイスには、プラットフォーム依存のものもあります。つまり、デバイスが特定のプロセッサ・モデルだけでサポートされているということです。同様に、オペレーティング・システム機能の中には、特定のプロセッサ・モデルだけで利用できるものがあります。これら以外のオペレーティング・システム・オプションについては、構成によって異なります。詳しくは、システムのハードウェア・ドキュメントを参照してください。

新しいオプションを追加したり、システムのハードウェアを変更したりするときは、『インストレーション・ガイド』および、ハードウェアまたは

ファームウェアのドキュメントの指示に従ってください。ただし、新しいオプションがオペレーティング・システムの最新のバージョンだけでサポートされている場合には、次の手順でアップグレードを実施します。

1. オペレーティング・システム・ソフトウェアをアップデートします。
2. ファームウェアをアップグレードします。
3. ハードウェアをアップグレードするか、新しいオプションをインストールします。
4. Tru64 UNIX 『インストレーション・ガイド』の指示に従ってシステム・カーネルを再構築します。

7.1 ストレージ・デバイス

この節では、Tru64 UNIX においてストレージ・デバイスおよびストレージ・アレイを構成し、管理する方法について具体的な情報を提供します。以下の項目について取り上げます。

- パラレル・スキャニング。大規模なストレージ構成でも短い時間でブートできるようにする機能です (7.1.1 項)
- コンソール・レベルのマルチパス・サポートに関する制限事項 (7.1.2 項)
- クラッシュ・ダンプを得るためにブート・デバイスとスワップ・デバイスを正しく指定するファイバ・チャネル・システムの構成 (7.1.3 項)
- ディスクを RAID アレイから取り外したときに、そのディスク・ラベルを利用できるようにする方法 (7.1.4 項)
- 故障した HSZ40 コントローラと HSZ50 コントローラの交換方法 (7.1.5 項)
- 故障した HSZ70 コントローラの交換方法 (7.1.6 項)
- HSZ80 コントローラと HSG80 コントローラの下ディスク上にある永続的予約をクリアする方法 (7.1.7 項)
- フロッピー・ディスク・ドライブの代わりに CD 書き込みドライブを使用することに関する制限事項 (7.1.8 項)
- UFS フォーマットまたは DOS フォーマットのフロッピーを使えるようにフロッピー・ディスク・ドライブを構成する方法 (7.1.9 項)

注意

HSZ コントローラまたは HSG コントローラを管理するには、SWCC (StorageWorks Command Console) Version 2.3 以上を使う必要があります。Version 2.3 以前のバージョンは、Tru64 UNIX Version 5.0 以上ではサポートされていません。

HSZ シリーズの一部のコントローラは、Tru64 UNIX Version 5.1B から利用できなくなりました。廃止されたハードウェア・オプションについては、『リリース・ノート』を参照してください。

7.1.1 パラレル・スキャンング

パラレル・スキャンングは、Tru64 UNIX Version 5.1B の新しい機能で、システムの起動時間を短縮することを目的としています。システム構成の省略時の設定では、パラレル・スキャンングは無効になっています。この機能を有効にするには、この節で後述するように、ある構成変数の値を設定する必要があります。

SCSI バスとファイバ・チャンネル・バスのパラレル・スキャンングを有効にすると、システムはすべてのバスのスキャンを (順番にではなく) 同時に開始します。これによって、デバイスを検出する時間が短縮されます。これによって、ストレージ・デバイスがある程度の量または大量に接続されているシステムでは、ブートの時間を大幅に短縮できることがあります。

ただし、パラレル・スキャンングは新しく追加されるデバイスに対するデバイス名の割り当てに影響を与えます。スキャンによってバスが同時並列に照会されるため、新しいデバイスに割り当てられる名前は順番が無作為になります。システムは、スキャンのときにデバイスを検出した順番に従ってデバイス名を生成します。

たとえば、ストレージ・アレイに /dev/disk/dsk8 から /dev/disk/dsk32 までの名前が付いているディスクがあったとします。そこに新しいデバイスを追加すると、スキャンによって検出されるタイミングによっては、そのデバイスに /dev/disk/dsk159 というデバイス名が付きます。これが問題になるのは、ストレージ構成のデバイスに順番に名前を与えたい場合だけです。

したがって、デバイス名を特定の順番にしておく必要があるなら、システムが新しく追加されたデバイスをスキャンする前に、パラレル・スキャンング

を無効にしておく必要があります。パラレル・スキャンングを有効または無効にする方法はいくつかあります。設定の内容は、設定の方法に応じて、再設定するまで、またはシステムをリブートするまで有効です。パラレル・スキャンングを一時的に有効または無効にするには、次の手順に従います。

1. システムをシャットダウンします。

シャットダウンのオプションについては、『システム管理ガイド』と `shutdown(8)` を参照してください。

2. 新しいホスト・バス・アダプタと新しいデバイスをインストールします。
3. 次のコマンドを使ってシステムを対話的にリブートします。

```
P0>>> boot -flag ai
UNIX boot - Friday December 15, 2000

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

システムが対話プロンプト (#) を表示します。

4. 次のコマンドを使ってパラレル・スキャンングを (このブートだけで) 無効にします。

```
# vmunix io:parallel_edt_scan=0
```

5. システムが起動していて稼働中の場合、次のコマンドを使ってパラレル・スキャンングを無効にできます。

```
# sysconfig -r io_parallel_edt_scan=0
```

システムが起動していて、特定のバス・イベント (バス・リセットなど) をより高速に処理させたいければ、次のコマンドを使ってパラレル・スキャンングを有効にします。

```
# # sysconfig -r io_parallel_edt_scan=1
```

パラレル・スキャンングを永続的に有効にするには (つまり、システムのブートがあっても設定が維持される)、`sysconfigdb` コマンドか `dxkerneltuner` ユーティリティを使います。次の手順に従ってください。

1. `root` ユーザになって、次の `io` サブシステム・スタンザ・ファイルを作成します。

```
io:
parallel_edt_scan=1
```


このファイルを `parallel.stanza` という名前にします。io スタンザ・ファイルがすでにある場合には、新しいファイルを作成せずに、既存のファイルに前記の行を追加します。

2. 次のコマンドを使って、io サブシステムに対応する既存のエントリがないか確認します。

```
# sysconfigdb -l io
```

この手順の結果に応じて、手順 3 で使用するコマンド・オプション (`-a` または `-u`) が決まります。

3. 次のいずれかのコマンドを使ってパラレル・スキニングを構成します。
 - ターゲット・ファイルにエントリを追加するには、次のコマンドを使います。

```
# sysconfigdb -a -f parallel.stanza io
```

- 既存の io サブシステム・エントリを、スタンザに指定されているサブシステム・エントリと置き換えるには、次のコマンドを使います。

```
# sysconfigdb -u -f parallel.stanza io
```

7.1.2 コンソール・レベルのマルチパス・サポートに関する制限事項

AlphaServer 1000, AlphaServer 1000A, および AlphaServer 2x00 の各システムのコンソール・ファームウェアは、複数バスによるフェールオーバー・モードが有効になっている HSZ70, HSZ80, または HSG80 RAID Array コントローラのストレージ・ユニットについては、複数のブート・デバイスまたはダンプ・デバイスの選択をサポートしていません。

コンソールは、ブート元またはダンプ先のストレージ・ユニットへの可視のパスがなければなりません。複数バスによるフェールオーバー・モードによって、コントローラが別のコントローラにフェールオーバーした場合、故障したコントローラのサービスを受けていたデバイスはすべて、代替パスを通じてのみ可視となります。したがって、システムをブートする前に、コンソール環境変数 `bootdef_dev` を、ブート・デバイスに可視のパスに設定してください。

オペレーティング・システムのブートが完了すると、マルチパス・サポートが完全に利用できる状態となります。

7.1.3 ファイバ・チャネルのダンプの構成

システムの構成にファイバ・チャネル・デバイスが含まれている場合には、オペレーティング・システムで定義されているブート・ディスクとスワップ・ディスクの WWID (ワールドワイド識別子) を使ってそのコンソール・ポートを構成する必要があります。この構成は、クラッシュ・ダンプを可能にするために必要です。

システムの要件と構成設定については、次の手順で示します。

1. システムには、最低 1.25 GB から物理メモリの 2 倍のサイズまでに構成されたスワップ領域が必要です。たとえば、物理メモリが 4GB なら、ディスク上のスワップ領域は最低 5GB とします。スワップ・パーティションの追加とスワップ領域の構成方法については、『システム管理ガイド』を参照してください。
2. ブート・デバイスとスワップ・デバイスはすべて、4 つのコンソール・ポートのどれかを使うように構成する必要があります。各ブート・デバイスまたはスワップ・デバイスのコンソール WWID 番号が、次の手順に従って表示される WWID 番号に一致することを確認します。

- a. 次のように `consvar` コマンドを使って、コンソール・ポート (N_n) と $WWID_n$ のマッピングを調べます。

```
# consvar -g N1
# consvar -g wwid0
```

4 つのポートと WWID 値のすべてについて、このコマンドを繰り返します。

- b. ファイバ・チャネルの各ブート・デバイスとスワップ・デバイスのデバイス名を探します (`dev/disk/dskN`)。それは、`/etc/fstab` ファイルを調べ、AdvFS ルート・ドメインについては `showfdmn` を使い、スワップ・デバイスについては `swapon -s` コマンドを使います。

または、『システム管理ガイド』の説明に従って SysMan Station を使います。

- c. ファイバ・チャネルの各ブート・デバイスとスワップ・デバイスのハードウェア識別子 (HWID) を探すには、前の手順で取得したデバイス名を使います。たとえば次のようにします。

```
# /sbin/hwmgr view devices | grep devname
```

- d. ファイバ・チャネルの各ブート・デバイスとスワップ・デバイスの WWID を取得するには、それぞれの HWID を指定して次のコマンドを実行します。

```
# /sbin/hwmmgr show scsi -id HWID -full
```

- e. ファイバ・チャネルの各ブート・デバイスとスワップ・デバイスについて、hwid コマンドによって表示される WWID がコンソール・ポートにマップされている WWID に一致することを確認します。

- 3. デバイスの WWID とコンソール・ポートの WWID が一致しなければ、次の手順に従います。

- a. 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。

```
# /sbin/shutdown +5 "system going down for  
console reconfiguration"
```

- b. wwidmgr ユーティリティを『*Wwidmgr Users Manual*』の説明に従って使って、コンソールの WWID の設定をデバイスの WWID に一致させる。このマニュアルは、システムの Firmware CD-ROM の `../documentation` ディレクトリにあります。

7.1.4 アレイからのディスクの取り外し

新しい LSM ユニットの RAID アレイに構築したとき、または、RAID アレイの 1 つ以上のディスクをホスト・バス・アダプタに直結したとき、ディスクのラベルが使用できない状態場合があります。これによって、ディスクの新しいブロック・ゼロが、以前のユニットの旧ブロック・ゼロと同じブロックになることがあります。システムは、この不完全なラベルが、ディスクの領域をはみ出している、有効なものとして扱ってしまいます。ディスク・ラベルを修正するには、次の手順を実行します。

- 1. ディスク・ラベルを修正するには、次の手順を実行します。

```
# disklabel -z dskNN
```

- 2. 新しい省略時のラベルを作成するか、プロト・ファイルに基づいて事前構成されたラベルを適用するには、次のいずれかのコマンドを使います。

```
# disklabel -rwn dskNN  
# # disklabel -Rr dskNN PROTOFILE
```

7.1.5 故障した HSZ40 コントローラと HSZ50 コントローラの交換

本節では、次のデュアル・ユニット・モデルの故障したデュアル冗長コントローラを交換する方法を説明します。

- HSZ40 と HSZ50。シングル・フェールオーバー・モードまたは透過フェールオーバー・モードのどちらでも可能。ただし、マルチバス・モードは不可です。
- HSZ70。シングル、透過、マルチバスの各フェールオーバー・モードが可能。デュアル冗長構成でない構成には、シングル・スピンドル・ディスクの場合の手順に従います。

コントローラが故障したら、次の手順に従って故障したコンポーネントを交換して、構成を再スタートします。

注意

HSOF コマンドについて、コントローラのドキュメントにいくつかの注意書きが記載されています。データ喪失のリスクを避けるために、先に進む前に次の CLI コマンドをよく理解してください。

- SET FAILOVER COPY=configuration-source
 - SET NOFAILOVER
 - SET MULTIBUS_FAILOVER COPY=configuration-source
-

Tru64 UNIX オペレーティング・システムが、コンソール・プロンプト (P0>>>) を表示している状態でないことを確認します。表示されている場合には、次の手順に進む前にシングル・ユーザ・モードにシステムをブートします。

1. 正常な (機能する) コントローラで、HSOF CLI を使って、故障したコントローラを停止します。

```
HSZnn> set nofailover
```

2. 故障したコントローラが停止したことを確認するために、緑色に光る状態表示 LED が点滅していないことを確認します。

3. ハードウェアのドキュメントの説明に従って、故障したコントローラを物理的に取り外します。
4. UNIX のコマンド・プロンプトに対して、次のコマンドを入力します。

```
# /sbin/hwmgr scan scsi -bus N
```

-scan オプションは、非同期処理を指示します。このコマンドを実行すると、すぐにプロンプトに戻りますが、特にシステムに多数のデバイスが接続されている場合には、スキャンがカーネルの中で実行が続いている場合があります。故障デバイスの検出されたかどうかを知るには、次のように Event Manager (EVM) `evmwatch` コマンドを使ってハードウェア・イベントを監視します。

```
# evmwatch -A -f '[name sys.unix.hw.*]'
```

テープ・デバイスが接続されていると、スキャンが終了するまで時間がかかる場合があります。スキャンの対象となるバスを指定するには、-bus 修飾子を使います。次のコマンドの出力の中で、場所を示すフィールドを調べて正しいバス番号を確認します。

```
# hwmgr view devices
```

HSZ ユニットがデュアル構成からシングル構成に移行する様子が表示されます。

5. ハードウェアのドキュメントに従って、交換用のコントローラをインストールします。
6. ハードウェアのドキュメントに従って、交換用のコントローラを開始します。
7. 手順 4 に示した `hwmgr -scan scsi` コマンドを再度実行します。
8. 正常なコントローラで、次のいずれかの HSOE CLI コマンドを入力して、コントローラを適切なフェールオーバー・モードでデュアル冗長構成に戻します。

```
> set failover copy=this  
> set multibus_failover copy=this
```

HSZ ユニットがシングル構成からデュアル構成に移行することを示すメッセージが表示されれば、操作は正常に行われました。

7.1.6 故障した HSZ70 コントローラの交換

環境によっては、非常に厳密な回復手順または故障手順が要求され、システムが I/O パスを回復できないとき、HSZ70 コントローラが非常に短い時間でフェールオーバーしなければならないことがあります。しかし、故障したコントローラの置き換えなど、稼働中の構成を変更したい場合には、この短いフェールオーバー時間が問題を引き起こすことがあります。回復手順または故障手順の制限を避けて、システムが障害回復にかかる時間を長くするには、一時的にあるシステム構成パラメータを設定します。

次の点に注意してください。

- 透過フェールオーバー構成では、回復または故障の状態になる運用モードに置くことはできません。回復モードまたは故障モードは、マルチバス構成にのみ適用されます。HSZ70 コントローラはこのマルチバス構成をサポートします。HSZ80 コントローラもマルチバス構成をサポートしますが、回復時間の短さの影響を受けません。
- HSZ40 と HSZ50 コントローラは、シングル・フェールオーバー・モードまたは透過フェールオーバー・モードの場合にのみ構成できます。したがって、HSZ40 と HSZ50 コントローラは、回復時間の短さの影響を受けません。コントローラの構成手順については、7.1.5 項などに記載されている一般の手順に従ってください。

次の手順は、マルチバス・フェールオーバー・モードの HSZ70 コントローラを交換するために一時的に回復時間を再設定する方法を示します。交換する HSZ70 コントローラのペアに対するシステム I/O が非常に多いときにコントローラの交換を行うと、デバイスのファントム・エントリができてしまうことがあります。これを避けるには、コントローラのペアに対する I/O を減らすか完全になくしてからコントローラを交換します (コントローラのポート静止ボタンについては、HSZ70 のハードウェア・ドキュメントを参照してください)。

注意

コントローラのコマンド行インタフェース (HSOF) に関するドキュメントに、データの喪失についていくつかの警告が記載されています。データ喪失のリスクを避けるために、先に進む前に次の HSOF コマンドをよく理解してください。

- SET FAILOVER COPY=*configuration-source*
 - SET NOFAILOVER
 - SET MULTIBUS_FAILOVER COPY=*configuration-source*
-

1. Tru64 UNIX のシステム・プロンプトで、移行の手順の間だけ、もう一方の回復時間の設定にしておくために、次のコマンドを入力して切り替えを行います。

```
# /sbin/sysconfig -r cam_disk rec_use_alt_params=1
```

2. 次のコマンドを実行して、EVM (Event Manager) のバックグラウンド・タスクを設定して、ハードウェア・イベントを監視します。

```
# evmwatch -A -f '[name sys.unix.hw.*]'  
6771
```

このタスクによって、後述の手順で hwmgr コマンドの結果として生成されるハードウェア・イベントが表示されます。タスクに割り当てられたプロセス ID を記録しておきます。上の例では、6771 です。

3. 正常な (機能する) コントローラで、HSOF CLI を使って、故障したコントローラを停止します。

```
HSZ70> set nofailover
```

緑色に光る状態表示 LED が点滅していなければ、故障したコントローラは停止しており、次の手順に進むことができます。

4. ハードウェアのドキュメントの説明に従って、故障したコントローラを物理的に取り外します。

5. Tru64 UNIX のシステム・プロンプトに対して、次のコマンドを入力します。

```
# /sbin/hwmgr scan scsi
```

コンソールには、HSZ70 コントローラがデュアル構成からシングル構成に移行する様子を示すメッセージが表示されます。

scan オプションは、非同期処理を指示します。このコマンドを実行すると、すぐにプロンプトに戻りますが、特にシステムに多数のデバイスが接続されている場合には、スキャンがカーネルの中で実行が続いている場合があります。バックグラウンドで動作中の EVM タスクは、交換用コントローラの登録に関するメッセージがあれば表示します。

スキャンが可能な限り短時間で終わるようにするには、`hwmgr` コマンドを使ってコントローラが接続されているバスを調べ、そのバスだけを対象とするスキャンを実行します。次のコマンドの出力の中で、場所を示すフィールドを調べて正しいバス番号を確認します。

```
# /sbin/hwmgr view devices
```

6. ハードウェアのドキュメントの説明に従って、交換用の HSZ70 コントローラをインストールしてスタートします。
7. Tru64 UNIX のシステム・プロンプトに対して、次のように `scan` コマンドを再度入力します。

```
# /sbin/hwmgr scan scsi
```

8. 正常なコントローラ (交換したばかりのコントローラではなく) で次の HSOF コマンドを入力します。

```
hsz70> set multibus_failover copy=this
```

このコマンドによって、コントローラのペアが適切なフェールオーバー・モードで冗長デュアル・モードに戻ります。

9. Tru64 UNIX のシステム・プロンプトで、省略時の回復時間に戻すために、次のコマンドを入力して切り替えを行います。

```
# /sbin/sysconfig -r cam_disk rec_use_alt_params=0
```

HSZ70 コントローラがシングル構成からデュアル構成に戻るとき、コンソールに表示されるメッセージによって処理が正常に行われたことがわかります。

7.1.7 HSZ80 コントローラと HSG80 コントローラの下ディスク上にある永続的予約のクリア

TruCluster Server のクラスタ・メンバは、共用ストレージへのアクセスを調整するために永続的予約を使用します。これらの永続的予約は、デバイスや論理ボリュームへのアクセスを制御し、現在のクラスタに属さないシステムに対する防護壁を築きます。基盤のオペレーティング・システムは、これらの永続的予約を扱い方法を知りません。基盤オペレーティング・システムが、永続的予約のあるディスクにアクセスしようとする、I/O エラー・メッセージが返されます。そして I/O はすべて失敗します。

以前に HSZ80 または HSG80 コントローラのもとでクラスタ化されていたディスクにアクセスしようとするシステムの基盤オペレーティング・システム

ムをブート (または再インストール) すると、基盤オペレーティング・システムからそれらのディスクが見えないことがあります。多くの場合、これはディスクが以前にクラスタに属していて、クラスタのメンバのどれかがそのディスクに対する予約を設定し、共用バスの他のホスト (または、このクラスタに属さないメンバ) がディスクにアクセスできないようにしたからです。この問題は、クラスタのメンバまたはクラスタ自身がシャットダウンされたときに予約がクリアされていなかったことによります。

予約を削除するには、`/usr/sbin/cleanPR clean` コマンドを使います。`cleanPR` スクリプトは、接続されている HSZ80 および HSG80 デバイスにある永続的予約をすべて探し出してクリアします。

注意

クラスタに属しているシステムで `cleanPR` スクリプトを実行してはなりません。稼働中のクラスタがディスク・アクセスを制御するために使用している予約を削除するべきではありません。削除するとデータが失われることがあります。

次の 2 つの例を考えてみましょう。

- 最初の例では、基盤オペレーティング・システムを含むディスクをブートし、`cleanPR` コマンドを実行します。

以前にクラスタの中で使われていたストレージにアクセスしようとして永続的予約に遭遇した場合には、オペレーティング・システムをブートして、`/usr/sbin/cleanPR clean` コマンドを入力して予約をクリアします。

- 2 つ目の例では、HSG80 または HSZ80 コントローラに制御されるディスクにオペレーティング・システムをインストールしようとしており、ディスクに永続的予約があり、基盤オペレーティング・システムがインストールされていないものとします。

この例は、`cleanPR` スクリプトを実行できるオペレーティング・システムがないので、前の例より複雑です。しかし、`cleanPR` スクリプトは Tru64 UNIX Operating System CD-ROM に含まれています。したがって、インストールを CD-ROM から開始し、シングル・ユーザ・シェルを起動して `/usr/sbin/cleanPR clean` コマンドを実行して予約をクリアできます。

7.1.8 CD-R/W の書き込みオプション

システムによっては、CD-ROM リーダと 1.44 MB のフロッピー・ディスク・デバイスの代わりに、CD-R/W 書き込みデバイスがコンソール・ストレージとして搭載されています。フロッピー・ディスク・デバイスとは異なり、CD-R/W デバイスは、コンソールでのシステム構成の保存などのコンソール操作では利用できません。しかし、マルチユーザ・モードでは、記録可能な CD-R と CD-RW 媒体にファイルを書き込むことができます。

CD-R/W の書き込み方法については、`cdrecord(1)` および `mkisofs(8)` を参照してください。手順が説明されているオンラインの Best Practice 『コンパクト・ディスクへのデータの書き込み』は Web 上で次の場所から入手できます。
http://tru64unix.compaq.co.jp/document/bp/misc_bp.html

フロッピー・ディスク・デバイスの代わりに CD-R/W デバイスを装備するシステムには、次の制約が適用されます。

- CD-R/W デバイスだけを装備するシステムでは、フロッピー・ドライブを使う必要のある手順は当てはまらない。
- `mttools` コマンド・オプションと CDE の `dxmttools` ユーティリティを使って DOS フォーマットのフロッピー・ディスクを読み書きすることはできない。
- CD-ROM をマウントする際のコマンド・オプションについては、`mount(8)` を参照してください。

7.1.9 フロッピー・ディスクの構成

次の手順は、フロッピー・ディスクを Tru64 UNIX で使うために構成する方法を示します。

1. 次のコマンドを使って、フロッピー・ドライブの種類とそのデバイス特殊ファイルを調べます。

```
# hwmgr view devices | grep floppy
69: /dev/disk/floppy0c          3.5in floppy      fdi0-unit-0
```

このコマンド出力は、1 つだけあるフロッピー・ドライブが SCSI デバイスではなく、fdi 接続デバイスであることを示します。そのデバイス特殊ファイルの名前は `floppy0c` です。ただし、ドライブでフォーマットをするときはその a パーティション (`floppy0a`) を使います。

2. 未使用のフロッピー・ディスクをドライブに挿入し、次のコマンドを入力してディスクをフォーマットします。

```
# /sbin/fddisk -fmt -f /dev/rdisk/floppy0a
Disk type: 3.50 inch, HD (1.44MB)
Number of sectors per track: 18
Number of surfaces: 2
Number of cylinders: 80
Sector size: 512
interleave factor: 2:4
Formatting disk...
Percentage complete: Format complete, checking...
Quick check of disk passes OK.
```

3. フロッピー・ディスクにディスク・ラベルを書き込みます。

```
# disklabel -rw /dev/rdisk/floppy0a
/dev/rdisk/floppy0a:
2880 sectors in 80 cylinders of 2 tracks, 18 sectors
1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 64 i/g)
super-block backups (for fsck -b #) at:
32, 640, 1184, 1792, 2336,
```

4. フロッピー・ディスク上に UFS ファイル・システムを作成します。

```
# newfs /dev/rdisk/floppy0a
```

5. フロッピー・ディスクを使用できるようにマウントします。

```
# mount /dev/disk/floppy0a /mnt
```

このコマンドは、/mnt ディレクトリが存在し、現在使われていないと想定しています。マウント・ディレクトリがなければ、次のコマンドを実行して作成します。

```
# mkdir /floppy_disk
```

詳細については、fddisk(8) を参照してください。

DOS フォーマットのフロッピー・ディスクには、mttools コマンド群を使ってアクセスできます。それには、適切なソフトウェア・サブセットがインストールされており、プログラムが /usr/ucb ディレクトリに存在する必要があります。オプション・ソフトウェア・サブセットの詳細については、『インストール・ガイド』を参照してください。

mttools コマンド群でフロッピー・ディスクを使う前に、次のようにしてターゲット・デバイスを作成する必要があります。

1. /dev/disk ディレクトリには、/dev/disk/floppyNa、および /dev/disk/floppyNc という名前の 2 つのデバイス特殊ファイルがフロッピー・ディスク・パーティションとして含まれている必要があります。これらのファイルがない場合には、dsfmgr -n コマンドを使って作成します。
2. 次のようにして、フロッピーの c パーティションを mttools のターゲット・ファイルにリンクします。

```
# ln -s /dev/disk/floppy0c /dev/disk/floppy
```

3. フロッピー・ドライブの構成をテストするために、DOS フォーマットのディスクを挿入し、次のコマンドを実行します。

```
# /usr/ucb/mttools/mdir
Volume in drive A is "volume_name."
Directory for A: /
```

```
file type size date time
file type size date time
```

/usr/ucb/mttools ディレクトリにあるコマンドを使って、MS-DOS フォーマットのフロッピー・ディスクを管理できます。SCSI 接続のフロッピー・ドライブについては、mttools(1) を参照してください。

7.2 ホスト・バス・アダプタ (HBA) の管理

本節では、SCSI アダプタなど、特定のホスト・バス・アダプタ (HBA) の管理についての情報を提供します。HBA の管理方法と HBA に関する情報の入手方法を説明します。また、HBA の特定のモデルの構成方法や操作方法に関する情報のほか、構成時の制限についても説明します。

- AlphaServer 1000A および 2100A システム上の KZPSA (7.2.1 項)
- KZPBA (Qlogic ISP1040B) CAM エラー (7.2.2 項)

7.2.1 AlphaServer 1000A および 2100A システム上の KZPSA

KZPSA は、PCI/Single ChannelのFWD、SCSI-2 アダプタです。

AlphaServer 1000A および 2100A クラスのシステムでは、KZPSA SCSI アダプタが PCI/PCI ブリッジを介して接続されている場合には、ファームウェア

アのアップデートはサポートされません。詳細については、ハードウェアのインストール・マニュアルを参照してください。

7.2.2 KZPBA (Qlogic ISP1040B) の構成

Qlogic ISP1040B オプションを含むシステムでは、システムのブート時に次のような CAM エラーが発生することがあります。

```
pci2000 at pci0 slot 8
isp0 at pci2000 slot 0
isp0: QLOGIC ISP1020A
cam_logger: CAM_ERROR packet
cam_logger: bus 0
isp_probe
NVRAM parameters invalid, using driver Fast10 defaults
```

このエラーを修正するには、`eepromcfg` ユーティリティを使って NVRAM のパラメータを正しい設定にする必要があります。`eepromcfg` ユーティリティは、*Alpha Systems Firmware Update CD-ROM* の `/mnt-pnt/utility` ディレクトリにあります。ユーティリティの使用法については、同じディレクトリにある `readme.txt` ファイルを参照してください。

注意

KZPBA SCSI コントローラには 2 つの種類があります。
KZPBA-CA/CX は、Qlogic 製の ISP1020/1040 平衡 PCI/UltraSCSI シングル・チャネル・コントローラです。KZPBA-CB/CY は、Qlogic 製の ISP1020/1040 不平衡 PCI/UltraSCSI コントローラです。どちらのコントローラも、SRM コンソールでは Qlogic ISP1020/1040 コントローラと識別されます。

システムに含まれているのがどちらであるかを識別するには、外部コネクタの 68 ピン (HD) の付近にある記号を探します。外部コネクタの左側にある記号が `-< >>` なら、コントローラは ISP1020/1040 不平衡コネクタです。平衡 PCI コントローラは、`-< >` という印が付いています。

7.3 グラフィックス・アダプタ

本節では、システムにインストールされているグラフィックス・アダプタに関する情報を取得する方法を説明します。また、特定のモデルのグラフィックス・アダプタを構成する方法についても説明します。

注意

本節で取り上げるオプション設定の変更を加える前に、使用しているビデオ・モニタがその設定をサポートしていることを確認してください。Tru64 UNIX 『*QuickSpecs*』のサポート対象リストに含まれているモニタについては、オーナーズ・マニュアルの「*Video Resolutions*」と題する節を参照してください。

7.3.1 関連ドキュメントとリソース

グラフィックス・アダプタの構成についての詳細は、以下のドキュメントを参照してください。

- 『*X Window System Administrator's Guide*』 – X ディスプレイを構成する方法について一般的な情報を提供する。
- `xdec(1X)` – X Window System サーバおよび関連するシステム・ファイルについて利用できるオプションが記述されている。
- `xdpyinfo(1X)` – Xserver に関する情報を表示する。
- `xgc(1X)` – システムのグラフィックス能力を実演して見せるプログラム。
- `comet(7)` – PowerStorm 4D10T, Elsa GLoria, IntraServer Combo アダプタのデバイス・ドライバを記述する。
- `glXIntro(3)` – OpenGL アプリケーション・プログラミング・インタフェースの紹介。
- `sys_attrs_s3trio(5)` と `sys_attrs_vga(5)` – カーネルのグラフィックス・アダプタ・サブシステムに関する情報。

グラフィックスに関わる Best Practices ドキュメントは、オンラインで Web 上の次の場所で入手できます。

<http://tru64unix.compaq.co.jp/document/bp/index.html>

最新のドキュメントは次のとおりです。

- Adjusting Your Screen Settings
- Configuring Multiple Monitors on a Single System

7.3.2 グラフィックス・コントローラに関する情報の取得

現在インストールされ、使われているグラフィックス・コントローラのモデルを知るには、次のコマンドを入力します。

```
# /usr/sbin/sizer -gt  
COMET
```

hwmgr コマンドを使用して、次の例のようにグラフィックス・コントローラに関する情報を表示することもできます。

```
# /sbin/hwmgr get attribute -category graphics_controller  
57:  
name = comet0  
category = graphics_controller  
sub_category = 3D  
model = COMET  
power_mgmt_capable = 1  
video_memory_size = 4 MB  
num_planes = 8  
num_colors = 256  
X_resolution = 1024  
Y_resolution = 768  
registration_time = Thu Jul 25 13:12:24 2002  
user_name = (null) (settable)  
location = (null) (settable)  
software_module = (null)  
state = available  
state_previous = unknown  
state_change_time = none  
event_count = 0  
last_event_time = none  
access_state = online  
access_state_change_time = none  
capabilities = 0  
indicted = 0  
indicted_probability = (null)  
indicted_urgency = (null)  
disabled = 0
```

この例では、最初のコマンドがグラフィックス・アダプタの名前を示します。2 つ目のコマンドが、アダプタに関連付けられている属性 (プロパティ) を表示します。アダプタの名前 (comet0) とそのハードウェア識別子 (HWID)

である 57 という値を使って、ハードウェア階層のどの位置にあるかなど、グラフィックス・アダプタに関する他の情報を取得できます。

```
# /sbin/hwmgrr view hierarchy
52: connection pci3slot6
57: graphics_controller comet0
```

上の出力例は、わかりやすいように表示内容の一部を省略しています。SysMan Station を使用して、グラフィックス・アダプタの場所を調べたり、属性を表示させたりすることもできます。第 2 章を参照してください。

7.3.3 3DLabs OXYGEN VX1 PCI/AGP グラフィックス・コントローラ

3DLabs OXYGEN VX1 PCI グラフィックス・コントローラ (システム・オプション SN-PBXGF-AB) は、3DLabs GLINT R3 グラフィック・チップを使用した 2D コントローラです。このコントローラは、デュアル・ディスプレイと 24 のカラー・プレーンをサポートします。最大解像度は、垂直走査周波数 70Hz で 1920 × 1200 ピクセル (16ビット)、垂直走査周波数 85 Hz で 1280 × 1024 ピクセル (24ビット)。カードには、15 ピンの VGA コネクタが 1 つ付いています。

3DLabs OXYGEN VX1 AGP グラフィックス・コントローラ・アクセラレータ・モジュールは、単一の拡張スロットを持つ、32 ビット AGP グラフィックス・オプションで、サポート対象システムで 2D グラフィックスを高速化します。このモジュールは、3DLabs の GLINT R3 グラフィックス・チップを使用しています。

これらがサポートする Tru64 UNIX のリリースにおけるインストールと構成の方法については、『*3DLabs OXYGEN VX1 PCI Graphics Controller Installation Guide*』に記載されています。このマニュアルは、次の場所からダウンロードできます。

<http://www.compaq.com/alphaserver/download/ek-vx1gc-ig.pdf>

Tru64 UNIX に関して、次の制限事項がハードウェアのドキュメントに詳しく記述されています。

- シングル・ヘッド構成の PCI グラフィックス・コントローラをサポートする Tru64 UNIX のファームウェアのバージョンは V5.8 以上、マルチヘッド構成の場合は V5.9 以上です。
- シングル・ヘッド構成の AGP グラフィックス・コントローラをサポートする Tru64 UNIX のファームウェアのバージョンは V6.0 以上です。

- 複数のカラーマップはサポートされていません。アプリケーションがカラーマップをインストールしたり、インストール解除したりしてはなりません。
- 3DLabs OXYGEN VX1 コントローラは、一度に 1 つのビジュアル・タイプ (8 ビット PseudoColor) だけをサポートします。
- 特に指定しなければ、バッキング・ストアとセーブ・アンダーは有効です。

7.3.4 Radeon 7500 PCI/AGP グラフィックス・コントローラ

ATI Radeon 7500 AGP および PCI (システム・オプション 3X-PBXGG-AB および 3X-PBXGG-AA) は、ATI の RV200 グラフィックス・チップを使った 3D/2D グラフィックス・コントローラです。AGP の方は、シングル・スロットの 32 ビット・カードで、2 倍速/4 倍速で AGP Specification 2.0 に準拠します。PCI の方は、シングル・スロットの 32 ビット・カードで、66MHz のバス速度で、PCI 2.2 および hot-plug 1.1 標準規格に準拠します。

Radeon 7500 カードは、64 MB のオンボード DDR ビデオ・メモリを搭載し、2 つの独立のディスプレイ・コントローラを持ち、Digital Flat Panel Monitors (DVI-I) をサポートします。標準の CRT モニタ用のコネクタを持つ主ビデオ出力ポートは、75 Hz の垂直走査周波数で 2048 × 1536 ピクセル、24 ビット・ピクセルの解像度まで対応します。二次ビデオ出力ポートは、Digital Flat Panel Monitor を 75 Hz の垂直走査周波数で 1280 × 1024、24 ビット・ピクセルの解像度まで、または、CRT モニタを 75 Hz で 2048 × 1536、24 ビット・ピクセルの解像度までサポートします。

7.3.4.1 Radeon 7500 の識別

システムのブート時に、システムにインストールされているコントローラの数によって異なりますが、Radeon 7500 によって次のようなメッセージが表示されます。

```
radeon0 at pci0 slot 13
AGP 0.99 on Titan @ 0x00000000 16777216MB
radeon_initialize: Initialized radeon 1.0.0 20010105
```

システムがシングル・ユーザ・モードまたはマルチユーザ・モードのときに、次の hwmgr コマンドを使ってハードウェア識別子 (-id オプションに指定する) を調べ、コントローラの属性 (プロパティ) を表示します。

```
# /sbin/hwmgrr view hierarchy | grep graphics_controller
57:          graphics_controller radeon0
# /sbin/hwmgrr get attributes -id 57
name = radeon0
category = graphics_controller
sub_category = 2D
model = radeon
power_mgmt_capable = 1
video_memory_size = 0 MB
num_planes = 8
num_colors = 256
X_resolution = 1280
Y_resolution = 1024
registration_time = Fri Jul 19 09:00:33 2002
user_name = (null) (settable)
location = (null) (settable)
software_module = (null)
state = available
state_previous = unknown
state_change_time = none
event_count = 0
last_event_time = none
access_state = online
access_state_change_time = none
capabilities = 0
indicted = 0
indicted_probability = (null)
indicted_urgency = (null)
disabled = 0
```

この出力はあくまでも例です。コマンドによる実際の出力は、オペレーティング・システムによって決まるコントローラの現在の設定に一致しないことがあります。構成の詳細については、7.3.4.4 項および 7.3.4.5 項を参照してください。実行中のシステムでコントローラの解像度を知るには次のコマンドを使います。

```
# /usr/sbin/sizer -gr
1024x768
```

7.3.4.2 使用上の制限事項

次の制限事項は Tru64 UNIX の最新のリリースに適用されます。

- カラーマップ情報

Radeon 7500 グラフィックス・コントローラが一度にサポートできるインストール済みカラーマップは 1 つだけです。複数のカラーマップはサポートされていません。これについては、省略時のビジュアル・モード

が PseudoColor である 8 ビットの色深度に切り替えるときに気を付ける必要があります。一度に 2 つ以上の PseudoColor カラーマップを使おうとすると、カラーマップのちらつきが発生します。

アプリケーションがカラーマップをインストールしたり、インストール解除したりしてはなりません。これらの操作を実行できるのは、ウィンドウ・マネージャだけです。ただし、インストールまたはインストール解除するカラーマップに関する情報は、アプリケーションがウィンドウ・マネージャに提供しなければなりません。この情報を提供するには、Xlib 関数である `XSetWMColormapWindows()` を使用します。この関数は、指定したウィンドウの `WM_COLORMAP_WINDOWS` プロパティを設定します。呼び出しの構文については、`XSetWMColormapWindows(3X11)` を参照してください。

・ バッキング・ストアとセーブ・アンダーに関する情報

バッキング・ストアとセーブ・アンダーは無効になっています。アプリケーションは、これらの機能が利用できると想定してはなりません。また、露出イベントを処理できなければなりません。たとえば、`DefaultColormapOfScreen(3X11)` (`DoesBackingStore`) を参照してください。

詳細については、Radeon 7500 のハードウェア・ドキュメントを参照してください。

7.3.4.3 Radeon 7500 のビデオ・モード

Radeon 7500 グラフィックス・コントローラ (アダプタ) によってサポートされているビデオ・モードの一覧を次に示します。

表 7-1: Radeon 7500 のビデオ・モード (解像度)

解像度 (ピクセル)	色深度 (ビット・ピクセル)	垂直走査周波数 (Hz)
640x480	24, 16, 8	60, 72, 75, 85
800x600	24, 16, 8	60, 72, 75, 85
1024x768	24, 16, 8	60, 70, 75, 85
1152x864	24, 16, 8	60
1280x1024	24, 16, 8	60, 75, 85
1600x1200	24, 16, 8	60, 65, 75, 85

表 7-1: Radeon 7500 のビデオ・モード (解像度) (続き)

解像度 (ピクセル)	色深度 (ビット・ピクセル)	垂直走査周波数 (Hz)
1920x1440	24, 16, 8	60, 75
2048x1536	24, 16, 8	60, 65, 70, 75

Radeon 7500 グラフィックス・コントローラが一度にサポートできるインストール済みカラーマップは 1 つだけです。省略時のビジュアル・モードは次のとおりです。

- 色深度 (ビット・ピクセル) – 24 ビット TrueColor
- 解像度 (ピクセル) – 1024x768
- 垂直走査周波数 – 70 Hz

7.3.4.4 Xserver の構成設定

ビデオ・モードは、Xserver の構成設定ファイルである `/usr/var/X11/Xserver.conf` ファイルに指定されています。詳細は `xdec(1X)` を参照してください。ビデオ・モードを変更するには、必要に応じて次のパラメータを変更します。

- `-screen` – スクリーン解像度を指定する。
- `-depth` – 色深度を指定する。
- `-vclass` – デフォルトのビジュアル・モードを指定する。
- `-vsync` – 垂直走査周波数を指定する。

次の手順は、グラフィックス・モードの変更方法を示します。

1. リモート端末から `root` ユーザとしてログインします。
変更を加えるには、リモート端末ログインまたはコンソールのキャラクタ端末を使います。CDE で `root` としてログインして作業を行うと、ビデオ解像度が変わったときに問題に遭遇する可能性があります。
2. `/usr/var/X11/Xserver.conf` ファイルのオリジナルを保存しておき、変更を加えた新しいバージョンで問題が生じたときに元に戻せるようにします。たとえば次のようにします。

```
# cp /usr/var/X11/Xserver.conf /usr/var/X11/Xserver.confORIG
```

3. 元の `/usr/var/X11/Xserver.conf` を編集して必要な変更を加えます。ファイルの終わりに、ユーザ・コマンド・オプションを指定する `args` の指定があります。

```
! you specify command line arguments here
args <
    -pn
>
```

たとえば、省略時のビデオ・モードを、解像度 1280 × 1024、8 ビット・ピクセル、PseudoColor、75 Hz で動作するように変更するには、`Xserver.conf` ファイルの `args` セクションを次のように変更します。

```
args <
    -pn -screen 1280x1024 -depth 8 -vclass PseudoColor -vsync 75
```

`Xserver.conf` ファイルを保存します。

4. 次のようにして `Xserver` を再起動します。

```
# /usr/sbin/init.d/xlogin stop
# /usr/sbin/init.d/xlogin start
```

7.3.4.5 CDE Xserver の構成設定

CDE ユーザ環境については、`/var/dt/Xservers` ファイルを次のように変更することによって省略時のビデオ・モードを変更できます。この手順は、CDE にのみ当てはまるもので、他の X 互換のユーザ環境では使用できません。

1. リモート端末から `root` ユーザとしてログインします。
2. `/etc/dt/config` ディレクトリが存在することを確認します。存在しない場合には、次のようにして作成します。

```
# mkdir /etc/dt/config
# chmod 755 /etc/dt/config
```

3. 省略時の `Xservers` ファイルを次のようにコピーします。

```
# cp /usr/dt/config/Xservers etc/dt/config/Xservers
```

ファイルの終わりに、ユーザ・コマンド・オプションを指定する次のような箇所があります。

```
:0    Local local@console /usr/bin/X11/X :0
```

4. `/etc/dt/config/Xservers` ファイルを編集して適切なオプションを追加します。たとえば、省略時のビデオ・モードを、解像度 1280 ×

1024 , 8 ビット・ピクセル , PseudoColor , 75 Hz で動作するように変更するには , Xservers ファイルを次のように変更します。

```
:0 Local local@console /usr/bin/X11/X :0 -screen 1280x1024 -depth 8  
-vclass PseudoColor -vsync 75
```

この行に改行が含まれていてはなりません。上の指定例では , わかりやすくするために 2 行に分けています。

5. 次のようにして Xserver を再起動します。

```
# /usr/sbin/init.d/xlogin stop  
# /usr/sbin/init.d/xlogin start
```

旧式のハードウェアの管理

この章では、次のような特定の旧式のハードウェア・デバイスまたはオプションの追加および構成の手順について説明します。

- PCMCIA カード (8.1 節)
- CalComp グラフィックス・タブレット (8.2 節)
- Computer Interconnect (CI) バスと Hierarchical Storage Controllers (HSC) のサポート (8.3 節)

8.1 PCMCIA (PC カード) のサポート

オーナーズ・マニュアルおよびオペレーティング・システムの各リリースの『*QuickSpecs*』に記載されているとおり、一部のプロセッサでは、PCMCIA (PC カード) が使用できます。動作が確認されているカードは少数ですが、標準規格に準拠するカードであれば動作する可能性があります。

PCMCIA カードの構成に必要な手順は、次のとおりです。

1. ハードウェアとオペレーティング・システムが、PCMCIA (PC カード) をサポートしていることを確認します。他に必要な構成手順がないか、アダプタのベンダやカードのベンダからの情報を参照してください。不明な点があれば、各ベンダにお問い合わせください。
2. バス・タイプ (ISA または EISA) を調べます。この手順により、使用しているコンソールの構成方法が決まります。
3. アダプタを取り付けて、適切なコンソール・コマンドを使用して構成します。
4. カスタム・カーネルを構成し、デバイス特殊ファイルを作成します。オペレーティング・システムの初期のインストレーションおよび構成時にアダプタが取り付けられていて、カードが挿入されている場合は、この手順が不要なこともあります。この場合は、オペレーティング・システムがカードを検出し、カーネル構成エントリとデバイス特殊ファイルを

作成します。カーネルの構成については『システム管理ガイド』を参照し、デバイス特殊ファイルについては 1.5 節を参照してください。

5. `/etc/remote` ファイルをアップデートします。
6. 必要に応じて、カードの挿入およびイジェクトを行います。

カードの構成についての情報や、特定のリリースでの制約事項については、`pcmcia(7)` を参照してください。モデム接続については、`modem(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。

8.2 CalComp グラフィックス・タブレット

オーナーズ・マニュアルおよび Tru64 UNIX オペレーティング・システムの各リリースの『*QuickSpecs*』に記載されているとおり、一部のプロセッサでは、CalComp DrawingBoard III タブレットが使用できます。Xserver の Xinput 拡張をサポートしている他の入力デバイスも、同様の構成で動作します。このタブレットのソフトウェアをシステムにインストールする際には、システム・マウスをエミュレートするようにタブレットを構成できます。

タブレットを構成する手順は、次のとおりです。

1. `/usr/var/X11/Xserver.conf` ファイルに次のような行があることを確認します。

```
input <
<_dec_xi_db3 lib_dec_xi_db3.so XiDb3Init /dev/tty00:1:12:12:16:\
1:8:1000:1:1 >
>
```

指定されている `tty` は、タブレットが接続されているシリアル・ポート (COMM) です。

2. 次の構文を使って、このファイルの最後の行に、タブレットの設定を指定します。

```
device:mode:tabletWidth:tabletHeight:numbtms:corePointer:mouseScale:\
resolution:Xincrement:Yincrement
```

データ・フィールドの説明については、`calcomp(7)` を参照してください。

3. タブレットをシステムに接続し、電源を入れます。
4. Xinput 拡張がタブレットを認識するように、次のコマンドを入力して Xserver を再起動します。

```
# /usr/sbin/shutdown -r +5 \
"Turning on support for the Calcomp Drawingboard III tablet"
```


再スタートが完了すると、タブレットは Xserver 内に構成され、使用できるようになっています。

詳細および使用上の制約事項については、`calcomp(7)` を参照してください。

8.3 CI および HSC ハードウェアのサポート

CI (コンピュータ・インターコネクト) バスは、プロセッサと HSC (階層性記憶域制御装置) をコンピュータ・ルーム環境で接続する高速デュアルバス・バスです。HSC は組み込みのインテリジェント大容量記憶域制御装置の I/O サブシステムで、CI バスに接続された複数のホスト・ノードからディスクとテープにアクセスできるようにします。

注意

Tru64 UNIX の実装には以下の制約があります。

- 1 つの CI バスに接続できる HSC は最大 4 台までである。
- 1 つの CI バスには 1 つのホストが接続できる。
- どのような環境下でも、Tru64 UNIX ノードを OpenVMS クラスタ・メンバとして接続することはできない。OpenVMS システムおよび Tru64 UNIX システムを同じ CI バス上に含むような構成はサポートされません。

Tru64 UNIX は、CI ポート・アダプタおよび HSC 用のシステム通信アーキテクチャ (SCA) をサポートします。SCA はポートおよびクラス・ドライバをサポートし、TMSCP (tms) と MSCP (ra) デバイスの処理方法を標準化します。SCA は各機能を別々のアーキテクチャ層に分散し、ある層でのソフトウェアの変更が他の層に与える影響を最小限に抑えます。

8.3.1 ハードウェアの設定、制限事項、およびリビジョン・レベル

ハードウェアの物理的なコンポーネントと設定については HSC ハードウェア・マニュアルを、また、プロセッサおよびサポートされるデバイスについては該当するハードウェア・マニュアルを参照してください。CI アダプタ付きのプロセッサのみが HSC 構成をサポートします。

HSC ハードウェアを設定する場合には、HSC パラメータの取得または設定、リモート・システム間の接続の監視、ディスクまたはテープの状態の確認などを行うコマンドを実行するために、HSC に端末を接続します。

CI バス上の最大ホスト数は 16 です。CI バス上のホスト番号は 0 ~ 15 の番号でなければなりません。

注意

特に重要なパラメータは、システム ID およびシステム名です。スター・カブラ上のいずれのシステム ID またはノード名も重複してはなりません。

8.3.2 ソフトウェアのインストールおよび制約

インストール用ソフトウェアはシステムのコンポーネントの確認と構成に役立ちます。使用するプロセッサを対象とするインストールの基本的な手順をよく理解した上で、インストールを開始してください。

Tru64 UNIX ソフトウェアのインストール時にアクセス可能な MSCP (ra) ディスク・デバイスは、それぞれのユニット・プラグ番号によって、次のように明確に識別されなければなりません。

- ユニット・プラグ番号は 0 ~ 254 でなければならない。
- 各ユニット・プラグ番号は一意でなければならない。異なる制御装置に接続されていても、2 つのディスクが同じユニット・プラグ番号を持つことはできません。たとえば、制御装置 A 上のディスク用のユニット・プラグ番号が 5 であり、制御装置 B 上の別のディスク用のユニット・プラグ番号も 5 である場合には、どちらかの番号を変えなければなりません。
- ユニット・プラグ番号が一意のディスクは、2 個の異なる制御装置と接続することができる (デュアルまたはポーティング)。システム構成ファイル内でデバイス・エントリを指定する方法については、ra(7) を参照してください。

8.3.3 構成ファイルのエントリ

インストール・ソフトウェアにより、HSC のコンポーネントは必ずカーネル内に構成されます。また、`/usr/sys/conf/NAME` システム構成ファイルに記述されます。`NAME` には大文字でシステムを指定します。

『システム管理ガイド』に、CI あるいは HSC 構成に対応する次のエントリについての説明があります。

- `scs_sysid` パラメータの説明
- CI アダプタの仕様
- 制御装置とデバイスの仕様

8.3.4 HSC 制御装置および HSC ディスクのブート

Tru64 UNIX ソフトウェアは、DEC 7000 プロセッサおよび DEC 10000 プロセッサ上での HSC ディスクのブートをサポートします。HSC 制御装置が故障した場合は、HSC 制御装置に接続されているディスクにはアクセスできません。このようなディスクにアクセスすると、アクセスしたシステムは HSC のリブートが完了するまで中断したままになります。HSC ディスクをブートする方法についての詳細は、プロセッサ・ハードウェア付属のマニュアルを参照してください。

8.3.5 複数のホスト間でのディスク/テープ・ユニットの共用

HSC は複数のホスト間で共用することができますが、複数の Tru64 UNIX システムが同じパーティションに同時に書き込むことを防止するためのソフトウェア・インターロック・メカニズムは備えていません。そのため、次の制限事項があります。

- 共用されたディスク・ユニットに対しては読み取り操作のみが可能。書き込み可能ファイル・システムは共用できない。
- ディスクを共用する場合、ディスクは書き込み保護されたハードウェアでなければならない。
- 各ホストは、`mount` コマンドに読み取り専用オプション `-r` を指定してファイル・システムをマウントしなければならない。
- 単一のホストのみが、書き込み可能ファイル・システムを含むディスクをマウントすることができる。
- 複数の書き込み装置がパーティションを共用する必要がある場合は、ネットワーク・ファイル・システム (NFS) を使用する。

CI バス上のホスト間のディスク・ユニット所有権を調整する必要があります。たとえば、各ホストにディスク・ユニット番号を範囲指定して割り当てます。ディスク・アクセスを排他的ホスト・システムに制限するように

HSC 制御装置に指示することもできます。この設定により、CI バス上の他のホストが誤ってディスクにアクセスするのを防ぐことができます。詳細については、`radisk(8)` を参照してください。特に `-e` オプションおよび `-n` オプションを参照してください。

HSC 制御装置に接続されるテープ・ドライブは共用することができます。この機能によってテープ・ドライブの使用範囲を拡大することができます。アクセス・メカニズムにより、同時アクセスではなくシリアルなドライブの共用が可能になります。

索引

数字および記号

3Dlabs OXYGEN VX1

オプション..... 7-20

3Dlabs OXYGEN VX1 グラフィックス・アダプタ

構成 7-20

A

addrlimit フィールド..... 6-8

AdvFS

故障ディスクの交換..... 3-22

ファイル・システム

diskconfig による作成 5-6

AGP グラフィックス・アダプタ 7-20

AlphaServer 2x00

コンソールのマルチパス・サポート 7-5

AlphaServer TS202c 6-2

mksas コマンド..... 6-6

構成オプション 6-9

制限事項 6-3

AlphaServer 1000A システム

ECU 構成..... 6-45

KZPSA PCI SCSI アダプタ .. 7-16

解像度の制限事項 6-45

コンソールのマルチパス・サポー

ト 7-5

AlphaServer 1000 システム

ECU 構成 6-45

解像度の制限事項 6-45

AlphaServer 2100A システム

KZPSA PCI SCSI アダプタ .. 7-16

AlphaServer オンライン情報... 1-9

Alpha VME シングルボード・コンピュータ 6-1

B

Best Practices

グラフィックス・アダプタ... 7-18

binlogd

エラー・コード 6-3

デーモン..... 6-3

BOOTP..... 6-10

BSD pty 4-10

登録解除..... 4-11

C

CalComp DrawingBoard

構成 8-2

CAM 4-1

cam_data.c ファイルの変換 .. 4-2, 4-8

エラー 7-17

動的デバイス認識..... 4-2

CD-R/W..... 7-14

CDE Xserver

変更	7-25
cdrecord コマンド	7-14
CD-ROM の書き込み	7-14
CDSL	1-19
cdslinvchk コマンド	1-14
再作成	1-14
デバイス特殊ファイル	1-14
CI	8-3
構成	8-3
cma_dd サブシステム	6-9
component サブシステム	3-3
カテゴリ	3-8
CPU (中央演算処理装置)	1-2
CPU 情報の参照	2-7
hwmgr コマンド	1-8
管理	1-2
システム階層	1-2
ホット・スワップ	1-1

D

DDR	4-2
cam_data.c ファイルの変換 ...	4-7
ddr.dbase ファイル	4-7
EMC Symmetrix	4-6
help オプション	4-4
SCSI-2 標準	4-3
StorageTek	4-4
エントリの例	4-4
データベース	4-7
データベースの同期	4-7
データベース・フィールド	4-6
データベースへの変更のコンパイル	4-7

標準に対する準拠	4-3
ddr_config コマンド	4-2
help オプション	4-4
TagQueueDepth パラメータ ..	4-7
dd コマンド	5-14
データ・ディスクのクローニング	5-14
devswmgr	5-5
/dev ディレクトリ	1-17
df コマンド	5-16
空きディスク領域の確認	5-16
diskconfig	1-19, 5-6
disklabel コマンド .	1-19, 5-6, 7-7
-e オプションの使用	5-12
省略時ラベルの書き込み	5-11
ディスク・パーティションサイズの変更	5-12
ラベルのクリア	5-15
DMCC	6-1
dn_setup ユーティリティ	1-28
DOS フォーマットのプロッピー・ディスク	7-14
DSA ディスク	5-4n
dsfmgr コマンド	1-1, 1-17, 1-27, 1-29
dump_exmem_addr	6-12
dump_exmem_include	6-13
dump_exmem_size	6-12
dump_to_memory	6-12
dumpfs コマンド	5-16
ディスク空き領域の確認	5-16
du コマンド	5-17
使用中のブロックの報告	5-17
dxmtools ユーティリティ	7-14

E

EBMnn	6-1
ECU	6-1
AlphaServer 1000A システム	6-45
AlphaServer 1000 システム ..	6-45
eia0_TFTP_blocksize コンソール変 数	6-12
EISA 設定ユーティリティ	6-1
EMC Symmetrix	4-6
/etc/bootptab ファイル	6-11
/etc/ddr.dbase サードパーティ製のデバイス .	4-4
フィールド定義	4-4
/etc/disktab ファイル	5-6
/etc/dt/config ディレクトリ ...	7-25
/etc/fstab	1-17
/etc/inetd.conf ファイル	6-10
/etc/securettys ファイル	4-11
EVM ハードウェアの管理	1-5

F

fdi コントローラ	1-2
ftp	6-10
ftpd	6-10

G

glXIntro	7-18
-----------------------	------

H

HBA	1-3, 1-9, 7-16
------------------	----------------

KZPBA	7-17
--------------------	------

HP Insight Manager	1-6
---------------------------------	-----

HSC

制御装置の故障	8-5
制限事項	8-5
ホストの共用	8-5

HSC (階層性記憶域制御装置)

構成	8-3
----------	-----

HSG

故障またはクラッシュしたディス ク	3-22
コントローラ	7-3
ハードウェアの管理	3-22

HSV	1-4
------------------	-----

HSZ	1-4, 1-24
------------------	-----------

故障またはクラッシュしたディス ク	3-22
コントローラ	7-3
ハードウェアの管理	3-22

HSZterm	7-3
----------------------	-----

HSZ コントローラ

故障	7-8
----------	-----

HWID	3-10
-------------------	------

一意性	1-15
-----------	------

hwmgr

refresh オプション	3-28
---------------------	------

hwmgr コマンド

グラフィックス・アダプタ ...	7-19
使用法	3-3

I

I/O サブシステム	7-4
-------------------------	-----

J

joind 6-10

K

kmknod コマンド 4-1
KZPBA 7-17
 CAM エラー 7-17
KZPSA 1-9
KZPSA PCI SCSI アダプタ
 AlphaServer 1000A システム 7-16
 AlphaServer 2100A システム 7-16

L

LBN
 ディスクのパーティショニング 5-9
 ディスク・ラベル 5-9
loader コマンド 6-3
LSM
 disklabel コマンド 7-7
 故障ディスクの交換 3-22

M

M_EXEMPT タイプ 6-8
MAKEDEV コマンド 4-1
miniroot 6-9
mkfdmn コマンド 5-6
mkisofs コマンド 7-14
mknod コマンド 4-1
mksas コマンド 6-2, 6-9
 addlist ファイルの例 6-7
 ftp 6-7
 su 6-7

 -t オプションの使用 6-8
 メッセージ 6-7
mttools コマンド 7-14, 7-15

N

name サブシステム 3-3
newfs 5-6

O

OpenGL 7-18

P

PCI
 グラフィックス・アダプタ ... 7-20
PCMCIA カード
 構成 8-1
Personal Workstation 433au ,
 500au , および 600au システム
 キーボード 6-48
 サポートされているオプション・
 カード 6-48
PseudoColor 7-25
psrinfo コマンド 2-7
pty ドライバ 1-17, 4-1
 (擬似端末 も参照)
 securettys 4-11
 作成 4-10
 追加 4-9

Q

QIC テープ 1-24
Qlogic ISP1040B 7-17

quot コマンド
使用中のブロックの確認 5-18

R

Radeon 7500..... 7-18
CDE の構成設定 7-25
Xserver の構成設定 7-24
解像度 7-23t
垂直走査周波数 7-23t
制限事項 7-22
ビデオ・モード 7-21
radisk ユーティリティ 5-4n
RAID 1-4, 3-21
SWCC 1-11
故障またはクラッシュしたディスク 3-22
サポート 1-4
ディスクの移動 7-7
ハードウェアの管理 3-22
RAID (Redundant Array of Independent Disks) 1-4
RAID からの移動
ディスクの移動 7-7
rcmgr コマンド 6-11
refresh オプション 3-28
RZ..... 1-23

S

S3Trio..... 7-18
SAS カーネル 6-9
savecore コマンド 6-3, 6-13
SBC..... 6-1

/sbin/init.d/dhcp コマンド 6-12
SCA..... 8-3
SCSI..... 1-3, 3-3
HBA 7-16
KZPSA HBA 7-16, 7-17
WWID 1-16
アダプタ (コントローラ)..... 7-16
サポートされている標準 4-3
ディスク 1-23
デバイス認識 4-2
デバイス名 1-17
テープ 1-23
動的デバイス認識 4-2
scu コマンド 1-13, 5-4
SRM コンソール 1-10
StorageTek Tape 4-4
StorageWorks コマンド・コンソール 1-4, 7-3
SVR4 **pty** ネーム・スペース 4-8
SWCC..... 1-4, 7-3
sys_attrs_generic..... 6-3
sys_attrs_s3trio..... 7-18
sys_attrs_vga..... 7-18
sys_check..... 1-5
sysconfigdb コマンド 7-5
sysconfigtab..... 4-9
SysMan
グラフィックス・アダプタ... 7-20
ハードウェアの管理... 1-5, 1-13
SysMan Menu
参照
CPU データ 2-7
クラスタ 2-5

デバイス	2-6
ハードウェア階層	2-3
ハードウェア	2-1
ハードウェアの管理.....	2-2
ハードウェアの管理の作業....	2-1
SysMan Station	
参照	
デバイス・プロパティ	2-8
ハードウェア・データ.....	2-8
デバイスのモニタリング	2-8
ハードウェア・タスク	2-9
ハードウェアの管理....	1-13, 2-1
ハードウェアのモニタリング .	2-1
SYSV_PTY	4-11

T

TL895	3-17
TZ	1-23

U

UFS	
ファイル・システム	
diskconfig による作成	5-6
/usr/bin/X11/X ファイル	7-25

V

/var/dt/Xservers ファイル.....	7-25
VGA	7-18
VME	6-1
vmstat コマンド	6-8
vmzcore ファイル	6-3
VX1	7-20

W

Web リソース	1-9
WWID	1-1, 1-16
デバイス識別子	3-16
デバイス属性	3-16
データベース	1-15
メディア・チェンジャ	3-17

X

Xdec	7-18
xdpyinfo	7-18
xgc	7-18
xlogin	
起動と停止.....	7-25
Xserver	
CDE	7-25
/var/dt/Xservers ファイル	7-25
起動と停止.....	7-25
構成	7-24
ビデオ・モード	7-24
Xserver.conf ファイル.....	7-24

あ

アダプタ	1-2, 1-9
アップグレード, ハードウェア..	7-1
アプリケーション	1-9

い

インストール	
新規ハードウェア・オプション	7-1

え

永続.....	3-26
エラー	
CAM.....	7-17
デバイス特殊ファイル	1-29
ブート	1-29
エラー・コード	
回復可能エラー	6-3
環境	6-3
しきい値の報告	6-3
ダブル・エラー停止.....	6-3
メモリ	6-3

お

オプション・カードの追加	1-2
重なっているディスク・パーティション	
確認	5-15
オンライン情報	
AlphaServer.....	1-9
アダプタ	1-9
スイッチ	1-9
ストレージ.....	1-9
テープ	1-9
ドライバ.....	1-9
ハブ	1-9
ファイバ・チャネル.....	1-9
ファームウェア	1-9

か

階層.....	3-7
---------	-----

階層性記憶域制御装置 (HSC)....	8-3
解像度.....	7-19
パラメータ.....	7-24
解像度の制限事項	
AlphaServer 1000A システム	6-45
AlphaServer 1000 システム ..	6-45
書き込み, CD-ROM	7-14
カテゴリ, コンポーネント	3-8
画面解像度	7-19
環境モニタリング	6-3
カーネル	
cma_dd サブシステム	6-9
ddr_config の使用	4-2
miniroot	6-9
SAS.....	6-9
静的再構成.....	4-1
動的再構成.....	4-1
ドライバ.....	4-1
ネットワーク・ブート可能...	6-2,
6-9	
カーネル・セット管理.....	3-2

き

擬似端末	4-1, 4-9
BSD STREAMS ベース	4-9
clist ベース	4-9
追加	4-9
デバイス特殊ファイルの作成	4-10
共通アクセスメソッド (CAM) ...	4-1
共用	
ディスク.....	3-17
デバイス.....	1-16
キーボード	

Personal Workstation 433au ,
500au , および 600au システ
ム 6-48

く

クラスタ 3-13
CDSL ファイル 1-14
参照 2-5
クラスタのルート・ディスク
故障またはクラッシュ 3-25
クラッシュしたデバイス 3-22, 3-23
クラッシュ・ダンプ 6-12
ファイバ・チャネル・ストレ
ージ 7-6
グラフィックス・アダプタ 7-18
3Dlabs OXYGEN VX1 7-20
Best Practices 7-18
comet ドライバ 7-18
Elsa GLoria 7-18
hwmgr の使用 7-19
IntraServer Combo 7-18
PCI 7-20
PowerStorm 7-18
S3Trio 7-18
SysMan の使用 7-20
VGA 7-18
省略時の設定 7-19
グラフィックス・モード
特定 7-24
変更 7-24
クローニング・バックアップ ... 1-42

こ

構成オプション

デバイス名 1-17
故障デバイス 3-22, 3-23
コンソール 1-10
StorageWorks コマンド 7-3
コマンド 1-9
ストレージ・デバイス 7-14
ファームウェア 1-9
マルチパス・サポートに関する制限
事項 7-5
>>> コンソール・プロンプト ... 1-10
コントローラ 1-2
コンピュータ・インタコネ
クト・バス **CI (Computer
Interconnect)** 8-3
コンポーネント
サブシステム 1-4

さ

サイズ指定コマンド
グラフィックス解像度 7-24
参照
CPU 情報 2-7
クラスタ 2-5
システム (デバイス) の階層 ... 2-3
デバイス 2-6

し

磁気テープ・ドライブ
静的ドライバの追加 4-12
システム構成ファイル
擬似デバイスのエントリ 4-9
システム通信アーキテクチャ
(**SCA**) 8-3
システムのスタートアップ 1-29

システム・パーティション 6-13
システム・ファイル
 ハードウェア・データベース 1-14
システム・ファームウェア 1-10
周辺機器 3-9
除外メモリ 6-3
 M_EXEMPT タイプ 6-8
 ダンプ 6-12
 割り当て 6-8
色深度 7-24
シングルボード・コンピュータ.. 6-1

す

垂直走査周波数
 Radeon 7500 7-23t
スイッチ 1-9
スクリーン 7-19
スタンザ・ファイル 7-5
ストレージ・アダプタ 1-9
ストレージ・オンライン情報 1-9
スロット 1-3

せ

セクタ
 定義済み 5-9

そ

属性
 コンポーネント 3-9

た

ダブル・エラー停止 6-3
ダンプ
 メモリへのダンプ 6-12
 リモート 6-12

つ

追加
 ディスク 1-4
 テープ 1-33

て

ディスク 7-7
 DSA 5-4n
 RZ (SCSI) 1-3
 位置の検索 2-7
 移動 3-21
 永続 3-26
 抹消または登録解除 3-27
 カテゴリ 3-8
監視
 df コマンド 5-16
 領域 5-16
管理の例 3-6
共用 3-17
クローニング 5-14
交換 3-22
故障またはクラッシュ 3-22
コピー 5-14
参照
 プロパティ 2-8

識別子	3-10	カテゴリ	1-29
自動的に追加	1-4	共用	1-16
手作業による追加	4-11	クラス	1-29
使用量の確認	5-17	構成	1-17
静的ドライバの追加	4-12	参照	2-6
属性	3-9	追加または削除	1-5, 1-29
探索	3-7	テスト	1-13
デバイス特殊ファイル	1-25, 1-33	データベース	4-2
デバイス名	1-17	特定デバイスの管理	8-1
名前	3-16, 3-26	抹消	3-19
抹消または登録解除	3-27	ユーティリティ	
パーティショニング	5-6	関連	1-13
表示	3-13	情報の記載箇所	5-1
別名	5-6	デバイス固有のオプション	7-1
ベース名	5-7	デバイス特殊ファイル	1-19, 1-29
抹消	3-19	dn_setup	1-28
ラベル	5-9	dsfmgr	1-1
ワールドワイド識別子	3-16	RZ (ディスク) デバイス	1-23
ディスク構成ユーティリティ	1-19	TZ (テープ) デバイス	1-23
ディスクの移動	3-21, 7-7	エラー	1-29
ディスクの使用量	5-15	管理	1-27
ディスク・パーティション		再割り当て	1-33
重なっているパーティション		手作業による作成	4-1
ン	5-13	探索	2-6
サイズ	5-9	デバイス・カテゴリ	1-29
サイズの変更	5-12	デバイス・クラス	1-29
省略時ラベルの書き込み	5-11	データベースの修復	1-31
定義済み	5-9	番号の付け替え	1-42
パラメータの変更	5-12	フロッピー・ディスク	7-14
ディスク領域		変換	1-26
空き領域の確認	5-16	抹消	1-32
再割り当て	5-9	デバイスの命名	
使用中のブロックの確認	5-18	パラレル・スキャニング	7-3
デバイス	1-4, 3-13	無作為な名前の順番	7-3
WWID	1-16	命名の順番	7-3

デバイスのモニタリング 2-8
 デバイスのユーティリティ 5-1
 デバイス名 1-17, 3-16
 検証 1-28
 ディスク 1-26
 テープ 1-26
 変換 1-26
 テープ
 QIC 1-24
 TZ (SCSI) 1-3
 オンライン情報 1-9
 共用 3-17
 交換 3-23
 故障またはクラッシュ 3-23, 3-25
 自動的に追加 1-4
 デバイス特殊ファイル 1-24, 1-25
 デバイス特殊ファイルの変換 1-26
 デバイス名 1-17
 名前 3-16
 表示 3-13
 抹消 3-19
 ワールドワイド識別子 3-16
 テープ・ドライブ
 故障またはクラッシュ 3-23
 手作業によるテープ・ドライブの追加 4-11
 デーモン
 binlogd 6-3
 joind 6-10

と

動的デバイス認識 4-2
 ドライバのロード 4-2

トラブルシューティング
 デバイス 3-22, 3-23
 トランザクション 3-15

ね

ネットワーク・ブート可能カーネル
 AlphaServer TS202c 6-2
 ftp 6-7
 mksas コマンド 6-6
 作成 6-6, 6-9

は

バス 1-2
 ハブ 1-9
 パラレル・スキャニング 7-3
 parallel_edt_scan 7-4
 設定の有効化と無効化 7-3
 番号の付け替え (デバイス・ファイル) 1-42
 パーティション 5-6, 5-9
 ハードウェア
 アップグレード 7-1
 管理の概念 1-1
 特定デバイスの管理 8-1
 ハードウェアの管理 1-1
 CDE アプリケーション・マネージャ 1-13
 component サブシステム 3-3
 CPU (中央演算処理装置) 1-2
 devswmgr 5-5
 diskconfig 5-6

HSZ デバイスと HSG デバイス	構成オプション	7-1
ス	コントローラ	1-2
HSZ デバイスと HSV デバイス	コンポーネントの永続化	3-26
HWID	コンポーネントの自動的な追加	1-4
hwmgr コマンド	コンポーネント名	3-26
hwmgr コマンド・オプション	登録解除	3-27
MAKEDEV	抹消	3-27
name サブシステム	サブシステム	3-3
3-26, 3-27	参照	
RAID デバイス	CPU データ	2-7
RZ (SCSI) デバイス	クラスタ	2-5
SCSI CAM デバイスのテスト	システム階層	2-3
SCSI サブシステム	デバイス	2-6
SCSI デバイス	デバイス・プロパティ	2-8
scsi の編集	システム・デバイスの階層	1-4
scu	システム・トポロジ	1-4
sysconfig	システム・ファイル	1-14
SysMan	手作業によるデバイスの追加	4-11
SysMan Menu のハードウェア・タスク	手作業による方法	4-1
スク	使用例	3-6
SysMan Station	属性	3-9
SysMan Station からのタスクの起動	ディスク・デバイス特殊ファイル	1-25
アダプタ	ディスクのコピー	5-14
アプリケーション	ディスクの使用量	5-17
永続	ディスクの使用量の監視	5-17
抹消	ディスクのパーティショニング	5-6
オプション・カードの追加	ディスク領域	5-16
概要	ディスク領域の監視	5-16
カテゴリ	デバイス	
環境	共用	3-17
関連ユーティリティ	故障	3-22, 3-23
クラスタ	故障したクラスタ・ルート	3-25
グラフィックス・アダプタ	識別子	3-16
構成	表示	3-13

- 命名 3-16
- デバイス固有のオプション 7-1
- デバイス・テスト 1-13
- デバイス特殊ファイル 1-1
- デバイスの概要 1-3
- デバイスの共用 3-17
- デバイスの削除 3-19
- デバイスの命名 3-16
- デバイス・ユーティリティ 5-1
- テープ・デバイス特殊ファイ
ル 1-25
- ドキュメント 1-7
- トランザクション 3-15
- バス 1-2
- バス・スロット 1-3
- ハードウェア識別子 (HWID) . 3-10
- ハードウェアの位置の特定 3-7
- ハードウェアの探索 3-7
- 表示
 - 階層 3-7
 - クラスタ 3-13
 - コンポーネント・カテゴリ . 3-8
 - コンポーネント属性 3-9
 - デバイス 3-13
- プロセッサ固有の情報 6-1
- モデル 3-2
- ワールドワイド識別子
(WWID) 3-16

ひ

- ビデオ設定 7-18
- 解像度 7-19
- 複数のモニタ 7-19

- ビデオ・モード
 - Radeon 7500 7-23t
 - 構成 7-24
 - 特定 7-24

ふ

- ファイバ・チャネル 1-9, 4-6
 - クラッシュ・ダンプ 7-6
 - リンク速度 3-11
- ファイル・システム
 - diskconfig による作成 5-6
 - 空き領域 5-15
 - 空き領域の監視 5-16
 - 使用量の監視 5-17
 - 設定の表示 5-11
- ファームウェア 1-9
- 複数のモニタ 7-19
- プロセッサ固有の情報 6-1
- フロッピー・ディスク
 - mttools 7-15
 - UNIX 向けのフォーマット ... 7-14
 - インタフェース 1-2
 - デバイス特殊ファイル 7-14
- フロッピー・ドライブ 7-14
 - 構成設定 7-14
- ブート 1-29
 - sanity_check オプション 1-29
 - エラー 1-29
 - パラレル・スキャンニングの構成 7-3

ほ

- ホスト・バス・アダプタ . 1-3, 7-16

ま

マルチパス・サポートに関する制限事項..... 7-5

む

無効なパス 3-28

め

メディア・チェンジャ
 WWID 3-17
メモリ
 エラー , しきい値の報告 6-3
 巡視によるエラーの検出 6-3
 除外 6-3, 6-8
メモリ巡視機能 6-3

ゆ

有効なパス 3-28

ら

ラベル
 クリア 5-15

り

リンク速度 3-11

ろ

論理パーティション
 構成 6-13
論理ブロック番号 5-9

わ

ワールドワイド ID 1-1
ワールドワイド識別子..... 1-16

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
ハードウェア管理ガイド
AA-RTFJA-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