



Solaris 7 のシステム管理 (追補)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806-2178-10
1999 年 12 月

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

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

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

Sun, Sun Microsystems, AnswerBook2, AnswerBook, Java, JDK, Power Management, Sun Enterprise は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

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

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

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

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

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

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *Solaris 7 System Administration Supplement*

Part No: 806-1650-10

Revision A

© 1999 by Sun Microsystems, Inc.



目次

はじめに vii

1. 新規機能の概要 1

2. デバイス構成の強化 7

SPARC: ホットプラグ可能なシステムボードのための動的再構成 7

デバイス構成の改善 (devfsadm) 8

x86: cfgadm コマンドによる PCI ホットプラグ 9

 x86: ホットプラグ機能をサポートするハードウェア 9

 x86: cfgadm コマンドによる PCI ホットプラグ 9

▼ x86: PCI スロット構成情報を表示するには 10

▼ x86: PCI アダプタカードを取り外すには 11

▼ x86: PCI アダプタカードを追加するには 12

 x86: PCI 構成の問題の解決 14

SCSI ホットプラグ機能 (cfgadm) 15

 アタッチメントポイント 15

 cfgadm コマンドによる SCSI ホットプラグ接続 17

▼ すべてのデバイスの構成情報を表示するには 18

▼ SCSI コントローラを構成解除するには 19

▼ SCSI コントローラを構成するには 20

▼ SCSI デバイスを構成するには 20

- ▼ SCSI コントローラを切り離すには 21
- ▼ SCSI コントローラを接続するには 23
- ▼ SCSI バスに SCSI デバイスを追加するには 24
- ▼ SCSI コントローラにある同一のデバイスを置き換えるには 25
- ▼ SCSI デバイスを取り外すには 27
- 3. ファイルシステムの強化 29
 - DVD のサポート 29
 - Universal Disk Format ファイルシステムの使用 30
 - UDF の機能と利点 30
 - ハードウェアとソフトウェアの要件 31
 - ▼ DVD-ROM デバイスを接続するには 31
 - ▼ DVD-ROM デバイス上のファイルにアクセスするには 32
 - ▼ UDF ファイルシステムのパラメータを表示するには 32
 - ▼ UDF ファイルシステムを作成するには 33
 - ▼ UDF ファイルシステムのタイプを調べるには 33
 - ▼ UDF ファイルシステムを検査するには 33
 - ▼ UDF ファイルシステムをマウントするには 34
 - ▼ UDF ファイルシステムをマウント解除するには 34
 - ▼ デバイスに UDF ファイルシステムとボリューム名のラベルを付けるには 34
 - UFS ファイルシステムでのアクセス時刻更新の遅延 35
- 4. 診断機能の強化 37
 - コアファイル管理の強化 (coreadm) 37
 - 構成可能なコアファイルパス 38
 - コアファイル名の拡張 39
 - コアファイル名パターンの設定 40
 - setuid プログラムがコアファイルを生成するように設定する 40
 - ▼ 現在のコアダンプ構成を表示するには 41
 - ▼ コアファイル名パターンを設定するには 41

- ▼ コアファイル名パターンを表示するには 42
- ▼ プロセス別コアファイルパスを有効にするには 42
- ▼ グローバルコアファイルパスを有効にするには 42
 - コアファイルで発生する問題の解決 43
- リモートシステム用の新しいコンソール機能 43
 - 実行レベル変更時の補助コンソールメッセージの使用 45
 - 対話式ログインセッション時の `consadm` コマンドの使用 46
- ▼ 補助 (リモート) コンソールを有効にするには 47
- ▼ 補助 (リモート) コンソールのリストを表示するには 47
- ▼ システムリブート時に補助 (リモート) コンソールを有効にするには 48
- ▼ 補助 (リモート) コンソールを無効にするには 49
- 改良されたシステムブートとエラーメッセージ形式 50
 - システムブートとエラーメッセージ形式の変更 50
- ▼ システムメッセージ ID を有効にするには 51
- ▼ システムメッセージ ID を無効にするには 51
- 5. システム管理の注意事項 53
 - BIND 8.1 への移行 53
 - AnswerBook2 サーバーソフトウェアの更新 53
 - SPARC: Power Management ソフトウェアでの `cron` ジョブの実行 54

はじめに

『Solaris™ 7 のシステム管理 (追補)』では、Solaris 7-11/99、8/99、5/99、および 3/99 ソフトウェアリリースの新しいシステム管理機能の使い方を説明します。

関連マニュアル

このマニュアルでは、Solaris の新しいリリースにおいて追加および変更された機能について説明します。ここでの説明は、すでにリリースされている Solaris 7 のマニュアルセットの内容を補足または変更するものです。

Sun のマニュアルの注文方法

Sun Software Shop プログラムを利用して、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルまたは AnswerBook2™ CD をご注文いただけます。

マニュアルのリストと注文方法については、<http://www.sun.com/software/shop> を参照してください。

Sun のオンラインマニュアル

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

表記上の規則

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

表 P-1 表記上の規則

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

表 P-1 表記上の規則 続く

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

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

コード例は次のように表示されます。

■ C シェルプロンプト

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

■ Bourne シェルおよび Korn シェルのプロンプト

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

■ スーパーユーザーのプロンプト

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

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

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

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

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

一般規則

- 「x86」という用語は、一般に Intel 8086 ファミリに属するマイクロプロセッサを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Pentium III、Celeron の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。このマニュアルでは、このプラットフォームのアーキテクチャ全体を指すときに「x86」という用語を使用し、製品名では「Intel 版」という表記で統一しています。

新規機能の概要

この章では、Solaris 7-11/99、8/99、5/99、および 3/99 リリースのオペレーティング環境に追加された新しい機能について説明します。

表 1-1 Solaris 7 の新規機能

機能	サポート開始 リリース番号	説明
インストール (これらの機能については、『Solaris 7 のインストール (追補)』を参照してください)		
AnswerBook2 ab2cd スクリプトの更新	8/99	ab2cd スクリプトの更新により、ユーザーは AnswerBook2 文書を Solaris 7 Documentation CD から直接読み込むことができます。
システム識別時の DNS 構成	5/99	以前の Solaris リリースでは、マシンは NIS または NIS+ クライアントとしてのみ構成できました。Solaris 7-5/99 リリースでは、システム識別ユーティリティを使用して、システムを DNS クライアントとして構成できます。
Solaris Product Registry	3/99	Solaris Product Registry はインストールされたソフトウェアを管理するためのツールです。次のような作業を行うことができます。 <ul style="list-style-type: none">■ インストールおよび登録されているソフトウェアとソフトウェア属性の一覧の表示■ ソフトウェアのアンインストール■ インストーラの選択と起動

表 1-1 Solaris 7 の新規機能 続く

機能	サポート開始 リリース番号	説明
パッチアナライザ	3/99	パッチアナライザはシステムを解析し、Solaris 7 オペレーティング環境を Solaris 7-11/99 オペレーティング環境にアップグレードするときに削除またはダウングレードされるパッチを調べます。この解析はアップグレード時または Solaris 7 のシステム稼動中に実行できます。
(Intel 版のみ) PAE モードのサポートの追加	3/99	Pentium Pro のリリースで、Intel は拡張プロセッサに PAE (Physical Address Extension) というモードを採用しました。PAE モードを使用すると、複数のデータベースを実行したり、メモリーを大量に消費するアプリケーションを実行することができます。また、マシンで多数のオンラインユーザーをサポートできます。 この機能については、『Solaris 7 - 11/99 ご使用にあたって (Intel 版)』を参照してください。
システムとネットワークの管理		
Universal Disk Format (UDF) ファイルシステム	11/99	UDF ファイルシステムは、DVD (digital versatile disc または digital video disc の略) と呼ばれる光学媒体に情報を格納するための、業界標準フォーマットです。
DVD	11/99	このリリースでは、DVD (digital versatile disc または digital video disc の略) をサポートしています。
(Intel 版のみ) cfgadm コマンドによる PCI ホットプラグ機能	11/99	cfgadm コマンドは、x86 システム上でサポートされる PCI コントローラ用の PCI ホットプラグ機能を提供するように更新されました。
デバイス構成の改良 (devfsadm)	11/99	devfsadm コマンドは /dev および /devices ディレクトリ内の特殊デバイスファイルを管理するのに使用され、再構成が行われた場合にデバイスの構成変更を検知するように更新されました。
(SPARC™ 版のみ) Inter-Domain Network (IDN)	11/99	Inter-Domain Network (IDN) は、Dynamic System Domain (DSD) 間の高速ネットワーク接続機能を提供します。これにより DSD は、ケーブリングや特別なハードウェアを使用せずに、TCP/IP などの標準ネットワークインタフェースにより互いに通信できます。 IDN の詳細や、IDN がご使用のサーバーをサポートしているかについては、ハードウェアのマニュアルを参照してください。
coreadm	8/99	coreadm コマンドのサポートにより、Solaris システム管理者のコアファイル管理が改善されました。

表 1-1 Solaris 7 の新規機能 続く

機能	サポート開始 リリース番号	説明
cfgadm による SCSI ホットプラグ機能の拡張	8/99	cfgadm コマンドに、サポート可能な SCSI ハードウェアに対する SCSI ホットプラグ機能が追加されました。
BIND 8.1 への移行	8/99	この節は、『Solaris ネーミングの設定と構成』と『Solaris ネーミングの管理』における Bind 8.1 への移行の説明に対する更新情報です。
AnswerBook2 サーバー ソフトウェアの更新	8/99	AnswerBook2 バージョン 1.4.1 サーバーソフトウェアの更新により、ナビゲーションと性能が改善されました。
リモートコンソール	5/99	consadm コマンドを含む新しいシステムコンソール機能です。システム管理者およびサービスプロバイダはシステムで発生する問題をシリアルポートに接続されたモデムでシステムにダイヤルインし、コンソールメッセージを補助デバイスに出力することによってリモートから解決できます。
(SPARC 版のみ) 動的再 構成	5/99	動的再構成により、サービスプロバイダはシステムの動作中に、ホットプラグ可能なシステムボードを追加、取り外し、および取り替えることができ、リブートにかかる時間を節約できます。詳細は、ハードウェアのマニュアルを参照してください。
UFS ファイルシステムで のデフォルトのアクセス 時刻の更新	3/99	新しいマウントオプション dfratime と nodfratime を使用すると、UFS ファイルシステムでのアクセス時刻更新の遅延を有効または無効にできます。
改良されたシステムブ ートとエラーメッセージ形 式	3/99	syslog ログ機能で生成するメッセージに、数値による識別子、モジュール名、およびタイムスタンプを追加することにより、システムブートとエラーメッセージ形式が改良されました。
(SPARC 版のみ) 一時停 止しているシステムでの cron ジョブの処理の改 良	3/99	Power Management™ ソフトウェアを使用してシステムを一時停止および再開するとき、システムが一時停止している間に cron ジョブがスケジュールされているかどうかを検査します。このようなジョブがある場合、システムが再開されると、最初の cron ジョブのインスタンスが実行されます。この改良は Solaris 7 リリースから利用可能になり、Solaris 7 - 3/99 リリースでマニュアルに掲載されました。

開発者向け機能 (これらの機能については、『Solaris 7 のソフトウェア開発 (追補)』を参照してください)

表 1-1 Solaris 7 の新規機能 続く

機能	サポート開始 リリース番号	説明
(SPARC 版のみ) SCSI HBA ドライバ用の動的再構成のサポート	11/99	このリリースでは、動的再構成 (DR) サポートが SCSI デバイス用に更新されました。 ご使用のサーバーが DR をサポートするかどうかについては、ハードウェアのマニュアルを参照してください。
PCI ホットプラグ機能	11/99	PCI ホットプラグ機能が更新され、x86 サーバー用デバイスドライバの作成に関する情報が含まれるようになりました。
8 ビットビジュアルサポート	8/99	8 ビットビジュアル共有ライブラリにより、24 ビットハードウェアしか持たないデバイスドライバから 8 ビットビジュアルアプリケーションを表示できます。
『Solaris 7 64 ビット開発ガイド』の更新	3/99	『Solaris 7 64 ビット 開発ガイド』の更新内容の一覧です。
クラスタ対応デバイスドライバ用の DDI インタフェースの更新	3/99	この概要は、デバイスドライバ開発者向けに、デバイスクラス概念、および必要なインタフェースの変更と追加を紹介します。
Java™ Development Kit (これらの機能については、『Solaris 7 のソフトウェア開発 (追補)』を参照してください)		
JDK™ 1.1.7_08	11/99	JDK 1.1.7_08 には、次の拡張機能が含まれます。 <ul style="list-style-type: none"> ■ ユーロ通貨サポートの拡張 ■ 高性能 Just In Time (JIT) コンパイラ <p>JDK の最新バージョンに対応する Java Runtime Environment (JRE) が Web からダウンロードできます。</p> <p>http://www.sun.com/solaris/java</p>

表 1-1 Solaris 7 の新規機能 続く

機能	サポート開始 リリース番号	説明
Java 2 SDK 1.2.1_03	11/99	<p>Java 2 SDK 1.2.1_03 は、Java 2 に基づいた Java テクノロジーの最初の Solaris でのリリースです。これには、次の拡張が含まれます。</p> <ul style="list-style-type: none"> ■ スケーラビリティと性能の著しい向上 ■ メモリー管理システムの拡張 ■ 高性能でスケーラビリティのある Java Virtual Machine (JVM) ■ 高速な Java スレッドの同期化 ■ プログラムにおけるメモリーリークを発見するヒープ検査ツールによる、診断機能の拡張 ■ Just In Time (JIT) コンパイラの最適化
<p>デスクトップユーザー (これらの機能については、『Solaris 7 ユーザーズガイド (追補)』を参照してください)</p>		
Personal Digital Assistant (PDA) Sync	11/99	<p>PDA Sync により、カレンダーマネージャなどの Sun のアプリケーションのデータを、PDA 上の類似したアプリケーションのデータと同期させることができます。また、アプリケーションやデータベースをワークステーションまたはサーバーから PDA にインストールすることができます。</p>
Netscape™ アプリケーションの起動用ウィンドウ	11/99	<p>デスクトップアプリケーションフォルダウィンドウには、次の新しいアイコンが追加されました。</p> <ul style="list-style-type: none"> ■ Netscape ■ Netscape Composer ■ Netscape Mail ■ Netscape News
Xserver 11R6.4	11/99	<p>Xserver 11R6.4 は次の新機能を提供します。</p> <ul style="list-style-type: none"> ■ XPrint ■ Xkeyboard ■ Display Power Management Signalling ■ Xinerama ■ Color Utilization Policy ■ WebEnabledX <p>これらの機能用に新しいマニュアルページが提供されます。</p>

デバイス構成の強化

この章では、次の新しいシステム管理の情報について説明します。

- 7ページの「SPARC: ホットプラグ可能なシステムボードのための動的再構成」
- 8ページの「デバイス構成の改善 (devfsadm)」
- 9ページの「x86: cfgadm コマンドによる PCI ホットプラグ」
- 15ページの「SCSI ホットプラグ機能 (cfgadm)」

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 7 - 11/99 のマニュアルページには、「*Solaris 7 Reference Manual Collection*」には記載されていない新しい情報が提供されています。

SPARC: ホットプラグ可能なシステムボードのための動的再構成

この機能は、Solaris 7 - 5/99 ソフトウェアリリースで変更されました。

動的再構成 (DR) を使用すると、システムボードが故障しても、サーバーを停止せずにそのボードを論理的に切り離し、物理的に取り外して交換し、そして論理的に接続することができます。これにより、停止時間とリブートにかかる時間を節約できます。ボードを切り離した後、交換用のボードがなくても、サーバーは動作を継続できます。ドメインをサポートするサーバーでは、動的再構成 (DR) を使用すると、あるドメインから別のドメインに故障していないボードを割り当てることがで

きます。一部のサーバーでは、CPU およびメモリーボードも論理的に接続または切り離すことができます。

通常、システム管理者が DR コマンドを実行し、サービスプロバイダが実際のボードの交換を行います。このバージョンの DR をサポートするのは、特定の SPARC サーバーだけです。サーバーが DR をサポートするかどうかについては、ハードウェアのマニュアルを参照してください。

デバイス構成の改善 (devfsadm)

この機能は、Solaris 7 - 11/99 ソフトウェアリリースで更新されました。

devfsadm コマンドにより、/dev および /devices ディレクトリにある特殊デバイスファイルを管理する機能が改善されています。これには動的再構成イベントのサポートも含まれます。

以前の Solaris リリースでのデバイス構成は、/devices ディレクトリにある物理デバイスエントリを管理する drvconfig と、/dev ディレクトリにある論理デバイスエントリを管理する 5 つのリンクジェネレータ devlinks、disks、tapes、ports、および audlinks によって処理されていました。

これらのユーティリティは、ホットプラグ可能なデバイスに対応しておらず、複数のインスタンスを持つデバイスにも柔軟な対応は行えませんでした。互換性を維持する目的から drvconfig と他のリンクジェネレータは devfsadm ユーティリティへのシンボリックリンクになっています。ホットプラグ可能なデバイスの詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

再構成ブート処理と、動的再構成イベントに応じての /dev および /devices ディレクトリの更新は、devfsadm コマンドのデーモンバージョンである devfsadmd によって処理されます。このデーモンは、システムがブートしたとき /etc/rc* スクリプトによって起動されます。

devfsadmd は再構成イベントにより発生するデバイス構成の変更を自動的に検知するため、このコマンドを対話的に実行する必要はありません。

詳細は、devfsadm(1M) のマニュアルページを参照してください。

x86: cfgadm コマンドによる PCI ホットプラグ

cfgadm コマンドは Solaris 7 - 11/99 ソフトウェアリリースにおいて更新され、x86 システムでサポートされる PCI アダプタカードに対し PCI ホットプラグ機能を提供します。ここでの説明は、『Solaris のシステム管理 (第 1 巻)』の「デバイスの管理」で説明されているデバイスの管理に関する情報を補足するものです。

PCI アダプタカードのホットプラグ機能により、システムをリブートすることなく、実行中のシステムに PCI アダプタカードを追加したり削除したりする機能を提供します。

Sun の高可用性方針の一部であるこの機能は、代替パスを提供するというような付加的な階層化製品や、デバイスに障害が発生した場合フォルトトレランスを提供するフェイルオーバーソフトウェアなどと組み合わせて使用できるように設計されています。

高可用性ソフトウェアがない場合は、障害を起こしたデバイスを交換するときに、アプリケーションの停止、デバイスの構成解除、追加または削除操作をすべて手動で行うことになります。

x86: ホットプラグ機能をサポートするハードウェア

現在、PCI ホットプラグ機能は Intel プラットフォームでサポートされています。ホットプラグ機能をサポートする PCI ハードウェアのリストについては、『Solaris 7 (Intel Platform Edition) 11/99 Hardware Compatibility List』を参照してください。

x86: cfgadm コマンドによる PCI ホットプラグ

次の節では、いくつかのホットプラグ操作を説明し、PCI アダプタカードについてホットプラグを行う手順を説明します。

PCI アダプタカードの取り外し

接続されているシステムリソースがアクティブではない PCI アダプタカードは、デバイスドライバがホットプラグ機能をサポートする場合、取り外すことができます。

PCI アダプタカードがアクティブなシステムリソースになっている場合、取り外すことはできません。取り外せる PCI アダプタカードは次の条件を満たす必要があります。

- デバイスドライバがホットプラグ機能をサポートしている
- 基幹となるリソースは、別のパスでアクセスできる

たとえば、システムに Ethernet カードが 1 枚だけインストールされている場合は、ネットワーク接続を解除せずに Ethernet カードを取り外すことはできません。ネットワーク接続を維持しながら取り外すには、別の階層化ソフトウェアのサポートが必要です。

PCI アダプタカードを取り外す詳細な手順は、11ページの「x86: PCI アダプタカードを取り外すには」を参照してください。

PCI アダプタカードの取り付け

PCI アダプタカードは次の条件を満たすとき、システムに追加できます。

- 空きスロットがある
- デバイスドライバがこのアダプタカードに対するホットプラグ機能をサポートしている

PCI アダプタカードを取り付ける詳細な手順は、12ページの「x86: PCI アダプタカードを追加するには」を参照してください。

次の例では、わかりやすくするため PCI アタッチメントポイントだけを示しています。実際のシステムに示されるアタッチメントポイントは、システム構成により異なります。

▼ x86: PCI スロット構成情報を表示するには

cfgadm(1M) コマンドを使用すると、PCI ホットプラグ可能なデバイスとシステムのスロットの状態を表示できます。

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

```
# cfgadm pci
Ap_Id                               Receptacle  Occupant    Condition
```

(続く)

```

pci1:hpc0_slot0          empty          unconfigured unknown
pci1:hpc0_slot1          empty          unconfigured ok
pci1:hpc0_slot2          empty          unconfigured ok
pci1:hpc0_slot3          connected     configured ok
pci1:hpc0_slot4          empty          unconfigured ok
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                    Type           Information
pci1:hpc0_slot0          unknown       Slot 7
pci1:hpc0_slot1          unknown       Slot 8
pci1:hpc0_slot2          unknown       Slot 9
pci1:hpc0_slot3          Ethernet/HP   Slot 10
pci1:hpc0_slot4          unknown       Slot 11

```

論理 ap_id である pci1:hpc0_slot0 は、そのホットプラグ可能スロット Slot 7 (このスロットの物理的な識別子) の論理 ap_id です。構成要素 hpc0 は、このスロットのホットプラグ可能アダプタカードを表し、pci1 は PCI バスインスタンスを表します。Type フィールドは、そのスロットにある PCI アダプタカードの種類を表します。

▼ x86: PCI アダプタカードを取り外すには

1. スーパーユーザーになります。
2. コントローラが入っているスロットを調べます。

```

# cfgadm
Ap_Id                    Receptacle    Occupant      Condition
pci1:hpc0_slot0          empty         unconfigured  unknown
pci1:hpc0_slot1          empty         unconfigured  unknown
pci1:hpc0_slot2          empty         unconfigured  unknown
pci1:hpc0_slot3          connected     configured    ok
pci1:hpc0_slot4          empty         unconfigured  unknown

```

3. デバイスを開いているアプリケーションを停止します。
たとえば、これが Ethernet カードの場合は、ifconfig(1M) を使用してインタフェースを停止します。

4. デバイスの構成を解除します。

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

5. デバイスが構成解除されていることを確認します。

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

6. スロットへの電源を切ります。

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

7. デバイスが切り離されていることを確認します。

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

8. スロットラッチを開き、ボードを取り外します。

▼ x86: PCI アダプタカードを追加するには

1. スーパーユーザーになります。
2. ホットプラグ可能なスロットを確認し、ラッチを開きます。
3. アダプタカードをホットプラグ可能なスロットに挿入します。

4. 挿入し、ラッチを閉じたら、アダプタカードがどのスロットに入っているかを確認します。

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

5. スロットへの電源を入れます。

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

6. スロットが接続されていることを確認します。

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

7. PCI ホットプラグ可能なアダプタカードを構成します。

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

8. スロットに入っているアダプタカードの構成を確認します。

```
# cfgadm pci1:hpc0_slot3
Ap_Id                Receptacle  Occupant    Condition
pci1:hpc0_slot3     connected   configured  ok
```

9. 新しく追加したデバイスの場合は、サポートするソフトウェアを構成します。
たとえば、これが Ethernet カードの場合は、`ifconfig(1M)` コマンドを使用してインタフェースを設定します。

x86: PCI 構成の問題の解決

エラーメッセージ

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

原因

無効な移行を実行しようとした。

解決方法

`cfgadm -c` コマンドを正しく実行したかを確認してください。`cfgadm` を使用して、現在の受容体 (`receptacle`) と占有装置 (`occupant`) の状態を調べ、`ap_id` が正しいか確認してください。

エラーメッセージ

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

原因

指定したアタッチメントポイントが見つかりません。

解決方法

アタッチメントポイントが正しいかどうかを確認してください。`cfgadm` コマンドを使用して、使用可能なアタッチメントポイントのリストを表示します。また、アタッチメントポイントがまだその物理バスにあるかどうかを確認してください。

注 - `cfgadm` コマンド以外にも、ホットプラグ操作に役立つコマンドがあります。`prtconf(1M)` コマンドでは、Solaris がハードウェアを認識するかどうかを確認します。ホットプラグ可能なアダプタカードの構成が正常に終わった後 `prtconf` コマンドを使用すると、新しいハードウェア用の特定の PCI バスインスタンスが `prtconf` 出力に表示されるかどうかを確認できます。`prtconf` 出力にはまた、新しいハードウェア用のドライバが接続されているかどうかを示されます。

SCSI ホットプラグ機能 (cfgadm)

Solaris 7-8/99 ソフトウェアリリースで更新された `cfgadm` コマンドでは、サポートされる SCSI コントローラに対し SCSI ホットプラグ機能が使用できます。この情報は、『Solaris のシステム管理 (第 1 巻)』の「デバイスの管理」に記載されているデバイス管理に関する情報の補足です。

ホットプラグ機能とは、システムの動作中にシステム構成要素を物理的に追加、削除、または置き換えることができる機能です。動的再構成とは、システム構成要素をホットプラグできる機能や、さらにシステムにおいてシステムリソース (ハードウェアとソフトウェア) を移動する、またはシステムからシステムリソースを取り除かずにそれらを無効にする機能全般をいいます。

`cfgadm` コマンドの機能には、次のものがあります。

- システム構成要素の状態の表示
- システム構成要素のテスト
- 構成要素の変更
- 構成ヘルプメッセージの表示

`cfgadm` コマンドを使って SCSI 構成要素を再構成することの利点は、システムの動作中に構成要素を追加、削除、または置き換えることができることです。その他の利点として、`cfgadm` コマンドでは、SCSI 構成要素の追加、削除、または置き換えに必要な手順が作業全体に渡って示されます。SCSI 構成要素をホットプラグ接続するための手順については、`cfgadm(1M)` のマニュアルページと、17ページの「`cfgadm` コマンドによる SCSI ホットプラグ接続」を参照してください。

注 - すべての SCSI コントローラが `cfgadm` コマンドによるホットプラグ接続をサポートしているわけではありません。

アタッチメントポイント

`cfgadm` コマンドでは、アタッチメントポイントに関する情報が表示されます。これは、動的再構成操作が行われる、システムの場所のことです。

アタッチメントポイントは、次の 2 つから構成されます。

- 占有装置 (occupant)。システムに組み込むハードウェアリソースを表します。

- 受容体 (receptacle)。占有装置のリソースを受け入れる場所のことです。

アタッチメントポイントは、論理的および物理的アタッチメントポイント ID (ap_id) によって指定されます。物理 ap_id は、アタッチメントポイントの物理パス名です。論理 ap_id は物理アタッチメントポイントのパス名を表す、ユーザーにわかりやすい名前です。ap_id の詳細は、`cfgadm(1M)` のマニュアルページを参照してください。

SCSI host bus adapter (HBA) の論理 ap_id、または SCSI コントローラは、通常、`c0` などのコントローラ番号で表します。

SCSI HBA にコントローラ番号が割り当てられていない場合は、内部的に生成された一意の識別子が与えられます。次は、SCSI コントローラの一意的識別子の例です。

```
fas1:scsi
```

SCSI デバイスの論理 ap_id は、通常、次のようになります。

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

次の例の `c0` は SCSI HBA の論理 ap_id です。

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

デバイス識別子は、一般に `/dev` ディレクトリの中のデバイスの論理デバイス名から抽出されます。たとえば、論理デバイス名が `/dev/rmt/1` であるテープデバイスの論理 ap_id は次のようになります。

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

SCSI デバイスの論理 ap_id を `/dev` ディレクトリの論理名から抽出できない場合は、内部的に生成された一意の識別子が使用されます。次は、テープデバイス `/dev/rmt/1` の識別子の例です。

```
c0::st4
```

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

`cfgadm` コマンドは、共通の一連の状態 (構成された、構成解除された、など) や一連の操作 (接続、構成、構成解除) などに関し、すべてのリソースとすべての動的再構成操作を表します。これらの全般的な状態や操作の詳細は、`cfgadm(1M)` のマニュアルページを参照してください。

SCSI HBA のアタッチメントポイントに対する占有装置 (occupant) と受容体 (receptacle) の状態は、次のとおりです。

受容体 (receptacle) の状態	説明	占有装置 (occupant) の状態	説明
empty	SCSI HBA には該当せず	configured	バスに 1 つ以上のデバイスが構成されている
disconnected	バスが静止している	unconfigured	構成されているデバイスはない
connected	バスがアクティブである		

SCSI デバイスに対する占有装置 (occupant) と受容体 (receptacle) の状態のマッピングは、次のとおりです。

受容体 (receptacle) の状態	説明	占有装置 (occupant) の状態	説明
empty	SCSI デバイスには該当せず	configured	デバイスが構成されている
disconnected	バスが静止している	unconfigured	デバイスは構成されていない
connected	バスがアクティブである		

SCSI アタッチメントポイントの条件は、他の条件を示す特別なハードウェアがない限り、「不明」です。SCSI 構成要素の構成情報を表示する手順については、18ページの「すべてのデバイスの構成情報を表示するには」を参照してください。

cfgadm コマンドによる SCSI ホットプラグ接続

次の各項では、cfgadm コマンドによる SCSI ホットプラグ接続について説明します。

この節の手順では、cfgadm コマンドを使用して SCSI 構成要素のホットプラグ接続を行う方法を特定のデバイスを例に説明します。ユーザーが指定するデバイス情

報や `cfgadm` コマンドにより表示されるデバイス情報は、システム構成によって異なります。

▼ すべてのデバイスの構成情報を表示するには

SCSI コントローラ `c0` や `c1`、それらに接続されているデバイスは、`cfgadm` コマンドで表示できるデバイス構成情報の一例です。

注 - `cfgadm` コマンドによってサポートされない SCSI デバイスは、`cfgadm` コマンドの出力に表示されません。

1. スーパーユーザーになります。
2. システムのアタッチメントポイントに関する情報を表示します。

```
# cfgadm -l
Ap_Id                Receptacle  Occupant    Condition
ac0:bank0            connected   configured  ok
ac0:bank1            connected   configured  ok
c0                   connected   configured  unknown
c1                   connected   configured  unknown
sysctrl0:slot0      connected   configured  ok
sysctrl0:slot1      connected   configured  ok
```

この例の `c0` と `c1` は、2つの SCSI コントローラを表しています。このコマンド出力の他の `ap_id` (`ac0:bank1` や `sysctrl0:slot0` など) は、他のタイプのシステム構成要素を表しています。

3. システムの **SCSI** コントローラとそれに接続されているデバイスの情報を表示します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
ac0:bank0            connected   configured  ok
ac0:bank1            connected   configured  ok
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0     connected   configured  unknown
c1::dsk/c1t4d0      connected   configured  unknown
c2                   connected   unconfigured unknown
```

(続く)

続き

```
sysctrl0:slot0          connected    configured  ok
sysctrl0:slot1          connected    configured  ok
```

cfgadm -l の出力では、SCSI HBA は表示されますが、SCSI デバイスは表示されないことに注意してください。ディスクやテープなどの SCSI デバイスの情報を表示するには、cfgadm -al コマンドを使用してください。

▼ SCSI コントローラを構成解除するには

SCSI コントローラを構成解除する例として、SCSI コントローラ c1 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。
2. **SCSI** コントローラを構成解除します。

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

3. **SCSI** コントローラが構成解除されているか確認します。

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

c1 の行の Occupant 列が unconfigured になっていますが、これは SCSI バスに、構成された占有装置 (occupant) がないことを示します。

▼ SCSI コントローラを構成するには

SCSI コントローラを構成する例として、SCSI コントローラ c1 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。
2. **SCSI** コントローラを構成します。

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

3. **SCSI** コントローラが構成されているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0     connected   configured  unknown
c1::dsk/c1t4d0      connected   configured  unknown
```

前の `unconfigure` 手順は SCSI バス上のすべてのデバイスを削除しました。現在すべてのデバイスはシステムに構成されます。

▼ SCSI デバイスを構成するには

SCSI デバイスを構成する例として、SCSI ディスク c1t4d0 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。
2. 構成するデバイスを特定します。

```

cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0     connected   configured  unknown
c1::dsk/c1t4d0      connected   unconfigured unknown

```

3. 特定の **SCSI** デバイスを構成します。

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

4. **SCSI** デバイスが構成されているかを確認します。

```

# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0     connected   configured  unknown
c1::dsk/c1t4d0      connected   configured  unknown

```

▼ SCSI コントローラを切り離すには

SCSI デバイスの切り離しは慎重に行ってください。特に、ルート (/)、usr、var、スワップパーティションなどの重要なファイルシステムが入っているディスクのコントローラに関連する場合は、注意が必要です。動的再構成ソフトウェアは、システムハングの原因となるすべてのケースを検出できるわけではありません。このコマンドの使用には充分注意してください。

SCSI デバイスを切り離す例として、SCSI コントローラ c1 を使用します。

1. スーパーユーザーになります。
2. デバイスを切り離す前に、それが接続されているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0     connected   configured  unknown
c1::dsk/c1t4d0      connected   configured  unknown
```

3. SCSI コントローラを切り離します。

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



注意 - このコマンドは、`cfgadm -c connect` コマンドが出されるまで、SCSI バスのすべての入出力動作を中断します。`cfgadm` コマンドは、重要なパーティションが切り離されるのを防止するため、基本的な検査を行います。すべてのケースを検出できるわけではありません。このコマンドを不適切に使用すると、システムハングが起り、システムリブートが必要になることがあります。

4. SCSI バスが切り離されているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0      connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   disconnected  configured  unknown
c1::dsk/c1t10d0     disconnected  configured  unknown
c1::dsk/c1t4d0      disconnected  configured  unknown
```

コントローラとそれに接続されているすべてのデバイスがシステムから切り離されています。

▼ SCSI コントローラを接続するには

SCSI コントローラを接続する例として、SCSI コントローラ c1 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。
2. デバイスを接続する前に、それが切り離されているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0       connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   disconnected configured  unknown
c1::dsk/c1t10d0      disconnected configured  unknown
c1::dsk/c1t4d0       disconnected configured  unknown
```

3. SCSI コントローラを接続します。

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

4. SCSI コントローラが接続されているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant    Condition
c0                   connected   configured  unknown
c0::dsk/c0t0d0       connected   configured  unknown
c0::rmt/0            connected   configured  unknown
c1                   connected   configured  unknown
c1::dsk/c1t10d0      connected   configured  unknown
c1::dsk/c1t4d0       connected   configured  unknown
```

▼ SCSI バスに SCSI デバイスを追加するには

SCSI バスに SCSI デバイスを追加する例として、SCSI コントローラ `c1` を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

注 - デバイスを追加するときは、デバイスが接続されている SCSI HBA (コントローラ) の `ap_id` を指定するのであって、デバイス自体の `ap_id` を指定するものではありません。

1. スーパーユーザーになります。

2. 現在の SCSI 構成を確認します。

```
# cfgadm -al
ap_id          Receptacle  Occupant    Condition
c0             connected   configured  unknown
c0::dsk/c0t0d0 connected   configured  unknown
c0::rmt/0      connected   configured  unknown
c1             connected   configured  unknown
c1::dsk/c1t10d0 connected   configured  unknown
```

3. SCSI デバイスを SCSI バスに追加します。

```
# cfgadm -x insert_device c1
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

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

SCSI バス上の入出力動作は、ホットプラグ操作が進行している間は停止しません。

b. デバイスを接続し、電源を入れます。

c. 新しいデバイスを挿入した

後、Enter y if operation is complete or n to abort (yes/no)?
というプロンプトに対して y と入力します。

4. デバイスが追加されているか確認します。

```
# cfgadm -al
Ap_Id                               Receptacle  Occupant    Condition
c0                                   connected   configured  unknown
c0::dsk/c0t0d0                       connected   configured  unknown
c0::rmt/0                             connected   configured  unknown
c1                                   connected   configured  unknown
c1::dsk/c1t10d0                       connected   configured  unknown
c1::dsk/c1t4d0                       connected   configured  unknown
```

新しいディスクがコントローラ c1 に追加されています。

▼ SCSI コントローラにある同一のデバイスを置き換えるには

SCSI コントローラにある同一のデバイスを置き換える例として、SCSI ディスク c1t4d0 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。

2. 現在の **SCSI** 構成を確認します。

```
# cfgadm -al
Ap_Id                               Receptacle  Occupant    Condition
c0                                   connected   configured  unknown
c0::dsk/c0t0d0                       connected   configured  unknown
c0::rmt/0                             connected   configured  unknown
c1                                   connected   configured  unknown
c1::dsk/c1t10d0                       connected   configured  unknown
```

(続く)

続き

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

3. SCSI バスのデバイスを同じタイプのデバイスで置き換えます。

```
# cfgadm -x replace_device c1::dsk/c1t4d0
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

- a. Continue (yes/no)? というプロンプトに対して y と入力して、次に進みます。
SCSI バス上の入出力動作は、ホットプラグ操作が進行している間は停止しません。
- b. 取り外すデバイスの電源を切り、デバイスを取り外します。次に置き換え用のデバイスを取り付けます。このデバイスは、取り外すデバイスと同じタイプで、同じアドレス (ターゲットと **LUN**) のものでなければなりません。デバイスの電源を入れます。
- c. 新しいデバイスを挿入した後、Enter y if operation is complete or n to abort (yes/no)? というプロンプトに対して y と入力します。

4. デバイスが置き換えられているか確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant  Condition
c0                   connected  configured unknown
c0::dsk/c0t0d0      connected  configured unknown
c0::rmt/0            connected  configured unknown
c1                   connected  configured unknown
c1::dsk/c1t10d0     connected  configured unknown
c1::dsk/c1t4d0      connected  configured unknown
```

▼ SCSI デバイスを取り外すには

SCSI コントローラのデバイスを取り外す例として、SCSI ディスク c1t4d0 を使用します。

ここでは SCSI アタッチメントポイントだけを示します。実際のシステムに表示されるアタッチメントポイントは、システム構成によって異なります。

1. スーパーユーザーになります。
2. 現在の **SCSI** 構成を確認します。

```
# cfgadm -al
Ap_Id                Receptacle  Occupant  Condition
c0                   connected  configured unknown
c0::dsk/c0t0d0      connected  configured unknown
c0::rmt/0            connected  configured unknown
c1                   connected  configured unknown
c1::dsk/c1t10d0     connected  configured unknown
c1::dsk/c1t4d0      connected  configured unknown
```

3. システムから **SCSI** デバイスを取り外します。

```
# cfgadm -x remove_device c1::dsk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

- a. Continue (yes/no)? というプロンプトに対して `y` と入力して、次に進みます。
SCSI バス上の入出力動作は、ホットプラグ操作が進行している間は停止しません。
- b. 取り外すデバイスの電源を切り、デバイスを取り外します。
- c. 新しいデバイスを挿入した後、Enter `y` if operation is complete or `n` to abort (yes/no)? というプロンプトに対して `y` と入力します。

4. デバイスがシステムから取り外されているか確認します。

```
# cfgadm -al
Ap_Id          Receptacle  Occupant    Condition
c0             connected   configured  unknown
c0::dsk/c0t0d0 connected   configured  unknown
c0::rmt/0      connected   configured  unknown
c1             connected   configured  unknown
c1::dsk/c1t10d0 connected   configured  unknown
```

ファイルシステムの強化

この章では、次の新しいシステム管理の情報について説明します。

- 29ページの「DVD のサポート」
- 30ページの「Universal Disk Format ファイルシステムの使用」
- 35ページの「UFS ファイルシステムでのアクセス時刻更新の遅延」

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 7-11/99 のマニュアルページには、「*Solaris 7 Reference Manual Collection*」には記載されていない新しい情報が提供されています。

DVD のサポート

Solaris 7-11/99 リリースでは、特定の SPARC および Intel ベース上のシステムにおいて DVD (digital versatile disc または digital video disc の略) デバイスをサポートします。

UDF ファイルシステムを通しての、DVD デバイス情報へのアクセスについては、次の節を参照してください。この情報は、『*Solaris のシステム管理 (第 1 巻)*』に記載されている情報を補足するものです。

Universal Disk Format ファイルシステムの使用

Solaris 7 - 11/99 リリースは、DVD デバイスに情報を格納するための業界標準フォーマットである Universal Disk Format (UDF) をサポートします。この情報は、『Solaris のシステム管理 (第 1 巻)』に記載されている情報を補足するものです。

UDF ファイルシステムは、動的にロード可能な 32 ビットおよび 64 ビットモジュールとして提供されます。また、SPARC および Intel プラットフォーム上でのファイルシステムの作成、マウント、検証を行うシステム管理ユーティリティも提供されます。Solaris UDF ファイルシステムは、サポートされる ATAPI および SCSI DVD デバイス、CD-ROM デバイス、ディスクおよびフロッピーディスクドライブに対応しています。さらに、Solaris UDF ファイルシステムは UDF 1.50 仕様に完全に準拠しています。

UDF ファイルシステムサポートは、次の新しいパッケージに提供されます。

- SUNWudfr — 32 ビットカーネル構成要素
- SUNWudfxr — 64 ビットカーネル構成要素
- SUNWudf — /usr 構成要素

UDF の機能と利点

Solaris 7 - 11/99 リリースでは、UDF ファイルシステムは次の機能を提供します。

- 業界標準の CD-ROM および DVD-ROM が UDF ファイルシステムを含む場合、それにアクセスする機能
- 異なるプラットフォームやオペレーティングシステム間で情報を交換できる柔軟性
- UDF フォーマットに基づく DVD ビデオ仕様を使用した、インタラクティブ性に富む、放送可能レベル品質のビデオと高品質サウンドを備えた新しいアプリケーションを実装する機能

次の機能は、この UDF ファイルシステムリリースには含まれません。

- disc-at-once 機能および incremental recording 機能を持つような、追記型 (write-once) 媒体や CD-RW および DVD-RAM のサポート。

- ディスク使用量、ACL、トランザクションロギング、ファイルシステムのロック、ファイルシステムのスレッドなどの UFS 構成要素。これらは UDF 1.50 仕様の必須機能ではありません。

ハードウェアとソフトウェアの要件

UDF ファイルシステムには次の要件があります。

- Solaris 7-11/99 リリース
- サポートされている SPARC または Intel のプラットフォーム
- サポートされている CD-ROM または DVD-ROM デバイス

UDF の互換性について

最初の Solaris UDF ファイルシステム実装には次のものが含まれます。

- 業界標準の、読み書き可能な UDF バージョン 1.50 のサポート
- 完全に国際化されたファイルシステムユーティリティ

▼ DVD-ROM デバイスを接続するには

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

```
# touch /reconfigure
```

3. システムをシャットダウンし、電源を切ります。

```
# init 0
```

4. **DVD-ROM** デバイスを接続します。
5. システムの電源を入れます。

▼ DVD-ROM デバイス上のファイルにアクセスするには

1. **DVD-ROM** デバイスが自動的にマウントされることを確認します。

```
$ ls /cdrom
```

注 - システムに CD-ROM と DVD-ROM デバイスが 1 つずつある場合、CD-ROM は /cdrom/cdrom0、DVD-ROM は /cdrom/cdrom1 という名前を付けられます。システムに DVD-ROM デバイスだけがある場合は、/cdrom/cdrom0 を試してください。

2. `ls` コマンドにより内容を表示します。

```
$ ls /cdrom/cdrom1
Copyright filea fileb
```

共通デスクトップ環境 (CDE) ファイルマネージャの自動表示は、まだ実装されていません。CDE ファイルマネージャの他の機能 (コピー機能を実行するドラッグ&ドロップやイメージツールなど) はすべて使用できます。

▼ UDF ファイルシステムのパラメータを表示するには

`mkfs` コマンドを使用すると、UDF ファイルシステムのパラメータを表示できます。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムのパラメータを表示します。

```
# mkfs -F udfs -m /dev/rdisk/device-name
```

▼ UDF ファイルシステムを作成するには

mkfs コマンドを使用すると、UDF ファイルシステムを作成できます。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムを作成します。

```
# mkfs -F udfs /dev/rdsk/device-name
```

3. **UDF** ファイルシステムをマウントすることによって、それが作成されていることを確認します。

詳細は、mkfs_udfs(1M) のマニュアルページを参照してください。

▼ UDF ファイルシステムのタイプを調べるには

fstyp コマンドを使用すると、UDF ファイルシステムのタイプを確認できます。

1. スーパーユーザーになります。
2. ファイルシステムが **UDF** ファイルシステムかどうかを確認します。

```
# fstyp -v /rdev/dsk/device-name
```

▼ UDF ファイルシステムを検査するには

fsck コマンドを使用すると、UDF ファイルシステムの整合性を検査できます。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムを検査します。

```
# fsck -F udfs /dev/rdsk/device-name
```

詳細は、fsck_udfs(1M) のマニュアルページを参照してください。

▼ UDF ファイルシステムをマウントするには

UDF ファイルシステムをマウントします。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムをマウントします。

```
# mount -F UDFS /dev/dsk/device-name /mount-point
```

3. **UDF** ファイルシステムがマウントされていることを確認します。

```
# ls /mount-point
```

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

▼ UDF ファイルシステムをマウント解除するには

UDF ファイルシステムをマウント解除します。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムをマウント解除します。

```
# umount /dev/dsk/device-name
```

▼ デバイスに UDF ファイルシステムとボリューム名のラベルを付けるには

UDF ファイルシステムについて、ファイルシステムとボリューム名を作成します。

1. スーパーユーザーになります。
2. **UDF** ファイルシステムについて、ファイルシステムとボリューム名を作成します。

```
# labelit -F UDFS /dev/rdisk/device-name fsname volume
```

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

UFS ファイルシステムでのアクセス時刻更新の遅延

この機能は、Solaris 7 - 3/99 ソフトウェアリリースで追加されたものです。

この情報は、『Solaris のシステム管理 (第 1 巻)』の「UFS ファイルシステム」と、『Solaris 移行ガイド』にある UFS ファイルシステムのマウントに関する追加説明です。

2 つの新しい `mount` オプション `dfratime` と `nodfratime` を使用すると、UFS ファイルシステムでのアクセス時刻更新の遅延を有効または無効にできます。

この機能を有効にすると、ファイルシステムのアクセス時刻更新の書き込みは、アクセス時刻の更新以外の理由でディスクがアクセスされるまで遅延されます。デフォルトの動作は `dfratime` です。`nodfratime` オプションを使用すると、この機能を無効にできます。

さらに、`mount` オプション `noatime` を使用すると、`dfratime/nodfratime` の設定にかかわらず、アクセス時刻の記録を無効にできます。

UFS マウントオプションの詳細については、`mount_ufs(1M)` のマニュアルページを参照してください。

診断機能の強化

この章では、次の新しいシステム管理の情報について説明します。

- 37ページの「コアファイル管理の強化 (coreadm)」
- 43ページの「リモートシステム用の新しいコンソール機能」
- 50ページの「改良されたシステムブートとエラーメッセージ形式」

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 7-11/99 のマニュアルページには、「*Solaris 7 Reference Manual Collection*」には記載されていない新しい情報が提供されています。

コアファイル管理の強化 (coreadm)

この機能は、Solaris 7-8/99 ソフトウェアリリースで更新されました。

Solaris 7 ソフトウェアリリースの更新には、coreadm コマンドを含む、強化されたコアファイル生成機能が含まれます。この情報は、『Solaris のシステム管理 (第 2 巻)』の「ソフトウェアの問題解決の概要」に記載されている、ソフトウェアの問題解決に関する情報の補足です。

このリリースで導入される coreadm コマンドでは、ユーザーによるコアファイルの命名規則の設定により、柔軟なコアファイル管理が提供されます。たとえば、coreadm コマンドでは、すべてのプロセスコアファイルを 1 つのシステムディレクトリに置くようにシステムを構成できます。したがって、Solaris のプロセスや

デーモンが異常終了した場合には、特定のディレクトリのコアファイルを調べればよい。そのため問題の追跡が容易になります。

これまでの Solaris プロセスコアダンプ機能には、次のような制約がありました。

- プロセスコアファイルは、プロセスの現在の作業ディレクトリに生成されます。したがって、Solaris のすべてのデーモンは、一般に初期化の過程で `chdir` を使ってルート (/) ディレクトリに移るため、相互のコアファイルが上書きされます。
- `statd` など多くのシステムデーモンが `setuid` 操作を行いますが、セキュリティ上の理由により、問題が発生した場合にコアファイルを生成しません。

構成可能なコアファイルパス

次の 2 つの構成可能な新しいコアファイルパスは、個別に有効または無効にできません。

■ プロセス別コアファイルパス

プロセス別コアファイルパスが有効な場合には、プロセスが異常終了するとコアファイルが生成されます。プロセス別コアファイルパスはプロセス自身の子プロセスに継承されます。このパスはデフォルトで有効になっており、パスには `core` が指定されています。

生成されたプロセス別コアファイルは、そのプロセスの所有者に、所有されます。このファイルを読み込むことができるのは、プロセスの所有者とスーパーユーザーだけです。

■ グローバルコアファイルパス

グローバルコアファイルパスが有効な場合には、追加のコアファイルが、指定したグローバルコアファイルパスのディレクトリにプロセス別コアファイルと同じ内容で、作成されます。このパスはデフォルトでは無効になっています。

生成されたグローバルコアファイルは、読み取り/書き込み可能ファイルとしてスーパーユーザーに所有されます。非特権ユーザーがこのファイルを表示することはできません。

プロセスが異常終了すると、これまでの Solaris リリースと同じように、このプロセスは現在のディレクトリにコアファイルを生成します。しかし、グローバルコアファイルパスが有効で、たとえば、`/corefiles/core` に設定されていると、異常終了した各プロセスは、2 つのコアファイルを生成します。1 つは現在の作業ディレクトリに、もう 1 つは `/corefiles` ディレクトリに生成します。

デフォルトでは、以下のようにこれまでの Solaris のコアファイル生成機能と変わらないように設定してあります。

- setuid プロセスが、グローバルパスやプロセス別パスを使ってコアファイルを生成しない。
- グローバルコアファイルパスが無効に設定されている。
- プロセス別コアファイルパスが有効に設定されている。
- プロセス別コアファイルパスが core に設定されている。

コアファイル名の拡張

グローバルコアファイルディレクトリが有効な場合、コアファイルは次の表に示す変数を使用してそれぞれ区別されます。

変数名	変数の定義
%p	プロセス ID
%u	実効ユーザー ID
%g	実効グループ ID
%f	実効可能ファイル名
%n	システムノード名。uname -n 出力と同じ
%m	マシン名。uname -m 出力と同じ
%t	time(2) システムコールの 10 進数値
%%	リテラル %

たとえば、グローバルコアファイルパスが次のように設定されているとします。

```
/var/core/core.%f.%p
```

このとき、PID 12345 の sendmail プロセスが異常終了すると、次のコアファイルが生成されます。

```
/var/core/core.sendmail.12345
```

コアファイル名パターンの設定

コアファイル名パターンを、グローバルにでも、プロセス別にでも設定できます。さらに、システムをリブートしてもこれらの設定値が保存されるようにするかどうかを指定することもできます。

たとえば、次の `coreadm` コマンドでは、`init` プロセスによって起動されるすべてのプロセスのグローバルコアファイルパターンを設定します。このパターンは、システムをリブートしても有効です。

```
$ coreadm -i /var/core/core.%f.%p
```

設定したグローバルコアの値は `/etc/coreadm.conf` ファイルに格納されます。つまり、これらの設定値はシステムをリブートしても保存されます。

次の `coreadm` コマンドでは、すべてのプロセスのプロセス別コアファイル名パターンを設定します。

```
$ coreadm -p /var/core/core.%f.%p $$
```

`$$` 記号は、動作しているシェルのプロセス ID を表す可変部分です。プロセス別コアファイル名パターンは、そのプロセス自身のすべての子プロセスに継承されます。

グローバルまたはプロセス別コアファイル名パターンが設定されると、`coreadm -e` コマンドで有効にしなければなりません。詳細は、次の手順を参照してください。

ユーザーの `$HOME/.profile` または `$HOME/.login` ファイルに上記のコマンドを指定すると、ユーザーのログインセッションで実行されるすべてのプロセスのコアファイル名パターンを設定できます。

setuid プログラムがコアファイルを生成するように設定する

`coreadm` コマンドを使用して、`setuid` プログラムがコアファイルを生成するかどうかを設定します。またこの設定は、すべてのシステムプロセスに適用するか、プロセス別に適用するかも選択できます。

- グローバル `setuid` オプションを有効にすると、グローバルコアファイルパスでは、システムのすべての `setuid` プログラムがコアファイルを生成します。

- プロセス別 `setuid` オプションを有効にすると、プロセス別コアファイルパスでは、特定の `setuid` プロセスがコアファイルを生成します。

デフォルトでは、両方のフラグが無効となっています。セキュリティ上の理由により、グローバルコアファイルパスは、/ で始まる完全パス名でなければなりません。スーパーユーザーがプロセス別コアファイルを無効にすると、各ユーザーはコアファイルを取得できなくなります。

`setuid` コアファイルは、スーパーユーザー専用の読み取り/書き込みアクセス権をもつスーパーユーザーによって所有されます。`setuid` コアファイルを生成したプロセスが通常ユーザーに所有されていた場合でも、通常ユーザーがこのファイルにアクセスすることはできません。

詳細は、`coreadm(1M)` のマニュアルページを参照してください。

▼ 現在のコアダンプ構成を表示するには

現在のコアダンプ構成を表示するには、`coreadm` コマンドをオプションなしで使用します。

```
$ coreadm
      global core file pattern: /var/core/core.%f.%p
      init core file pattern: core
      global core dumps: enabled
      per-process core dumps: enabled
      global setid core dumps: enabled
      per-process setid core dumps: disabled
      global core dump logging: disabled
```

▼ コアファイル名パターンを設定するには

1. プロセス別コアファイルを設定するのか、グローバルコアファイルを設定するのかを決めます。次の **a**、**b** のどちらかを選択します。

- a. プロセス別コアファイル名パターンを設定します。

```
# coreadm -p $HOME/corefiles/%f.%p $$
```

- b. グローバルコアファイル名パターンを設定します。

スーパーユーザーになります。

```
# coreadm -g /var/core/core.%f.%p
```

▼ コアファイル名パターンを表示するには

現在のプロセスのコアファイル設定値を照会するには、次の `coreadm` コマンドを使用します。\$\$ 記号は、動作しているシェルのプロセス ID を表す可変部分です。

```
$ coreadm $$  
278:    core.%f.%p
```

スーパーユーザーは、`coreadm process ID` を使用して、任意のユーザーのコアファイル設定値を照会できます。通常ユーザーは、自身のプロセスのコアファイル設定値しか照会できません。

▼ プロセス別コアファイルパスを有効にするには

1. スーパーユーザーになります。
2. プロセス別コアファイルパスを有効にします。

```
# coreadm -e process
```

3. 現在のプロセスコアファイルパスを表示して、構成を確認します。

```
$ coreadm $$  
1180:  /home/kryten/corefiles/%f.%p
```

▼ グローバルコアファイルパスを有効にするには

1. スーパーユーザーになります。
2. グローバルコアファイルパスを有効にします。

```
# coreadm -e global -g /var/core/core.%f.%p
```

3. 現在のプロセスコアファイルパスを表示して、構成を確認します。

```
# coreadm
  global core file pattern: /var/core/core.%f.%p
  init core file pattern: core
  global core dumps: enabled
  per-process core dumps: enabled
  global setid core dumps: disabled
  per-process setid core dumps: disabled
  global core dump logging: disabled
```

コアファイルで発生する問題の解決

エラーメッセージ

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

原因:

setuid コアファイルを指定する古いパラメータが `/etc/system` ファイルにあります。

解決方法:

`/etc/system` ファイルから `allow_setid_core=1` を削除します。次に、`coreadm` コマンドでグローバル setuid コアファイルパスを有効にします。

リモートシステム用の新しいコンソール機能

この機能は、Solaris 7-5/99 ソフトウェアリリースで追加されました。

Solaris 7-5/99 ソフトウェアリリースの新しいコンソール機能により、Solaris オペレーティング環境が更新されます。この情報は、『Solaris のシステム管理 (第 2 巻)』の「ソフトウェアの問題解決の概要」と『Solaris 移行ガイド』での Solaris システムにおける問題解決の説明を補足するものです。

次の新しいコンソール機能により、リモートシステムでの問題解決機能が改善されます。

- `consadm` コマンドにより、シリアルデバイスを補助 (または、リモート) コンソールとして選択できます。システム管理者は `consadm` コマンドを使用して 1 つまたは複数のシリアルポートを構成し、リダイレクトされたコンソールメッセージを表示したり、システムの実行レベルを変更するときに `sulogin` セッションを実行することができます。この機能を使用すると、モデムが接続されたシリアルポートにダイヤルインしてコンソールメッセージを監視し、`init` 状態の変更に対応できます。詳細は、`sulogin(1M)` のマニュアルページと次の手順の説明を参照してください。

補助コンソールとして構成されたポートを使用してシステムにログインすることもできますが、本来補助コンソールはデフォルトのコンソールにも表示される情報を表示する出力デバイスです。ブートスクリプトや他のアプリケーションがデフォルトのコンソールから読み込んだりここに書き込んだりする場合、書き込み出力はすべての補助コンソールに表示されます。しかし、入力にはデフォルトのコンソールからだけ読み取られます。補助コンソールは、ブートスクリプトに入力を行う場合には使用できません。対話式のログインセッション時に `consadm` コマンドを使用する方法については、46ページの「対話式ログインセッション時の `consadm` コマンドの使用」を参照してください。

- 新しいコンソール出力はカーネルと `syslog` メッセージから構成され、新しい疑似デバイス `/dev/sysmsg` に書き込まれます。また、`rc` スクリプトの起動メッセージは `/dev/msglog` に書き込まれます。以前は、すべてのメッセージは `/dev/console` に書き込まれていました。

補助コンソールにメッセージをリダイレクトさせたい場合、コンソール出力を `/dev/console` に送るスクリプトにおいては出力先を `/dev/msglog` に変更する必要があります。また、`/dev/console` を参照するソースコードにおいては `syslog()` または `strlog()` を使用するように明示的に変更しなければなりません。

- `consadm` コマンドは、補助コンソールデバイスを監視するデーモンを実行します。補助コンソールとして指定された表示デバイスが切断された (つまり、ハンダアップしたか、キャリア信号を失った) 場合、その表示デバイスは補助コンソールデバイスリストから削除され、無効になります。1 つまたは複数の補助コンソールを有効にしても、デフォルトのコンソールへのメッセージ出力は無効になりません。メッセージは `/dev/console` に引き続き表示されます。

実行レベル変更時の補助コンソールメッセージの使用

実行レベル変更時に補助コンソールメッセージ機能を使用する場合、次のことに注意してください。

- システムブート時に実行される rc スクリプトにおいてユーザーの入力を求める場合、補助コンソールからは入力できません。デフォルトのコンソールから入力しなければなりません。
- `sulogin` プログラム (`init` によって呼び出され、実行レベル変更時にスーパーユーザーのパスワード入力を求める) は、デフォルトのコンソールデバイスだけでなく、各補助デバイスにもスーパーユーザーのパスワードを求めるプロンプトを送信するように変更されました。
- システムがシングルユーザーモードであり、`consadm` コマンドで1つまたは複数の補助コンソールを有効にしている場合、コンソールログインセッションはスーパーユーザーの正しいパスワードを `sulogin` プロンプトに渡した最初のデバイスで動作します。あるコンソールデバイスから正しいパスワードを受信すると、`sulogin` は他のすべてのコンソールデバイスからの入力を無効にします。
- コンソールの1つがシングルユーザー特権を持った場合、メッセージがデフォルトのコンソールと他の補助コンソールに表示されます。このメッセージは、どのデバイスがスーパーユーザーの正しいパスワードを受け取ってコンソールになったかを示します。シングルユーザーシェルを実行している補助コンソールでキャリア信号が失われた場合、次の2つの動作のいずれかが行われる可能性があります。
 - 補助コンソールが実行レベル 1 で動作するシステムを示している場合、システムはデフォルトの実行レベルに進みます。
 - 補助コンソールが実行レベル S で動作するシステムを示している場合、システムは、`init s` または `shutdown` コマンドをシェルから入力したデバイス上に「ENTER RUN LEVEL (0-6, s or S):」というメッセージを表示します。そのデバイスにキャリア信号がない場合、再度キャリアを確立し、正しい実行レベルを入力しなければなりません。`init` または `shutdown` コマンドは実行レベルのプロンプトを表示し直しません。
- シリアルポートを使用してシステムにログインし、`init` または `shutdown` コマンドで実行レベルを変更しようとした場合、そのデバイスが補助コンソールかどうかにかかわらず、ログインセッションは失われます。この状態は、補助コンソール機能のない Solaris リリースと同じです。

- `consadm` コマンドを使用してあるデバイスを補助コンソールとして選択すると、システムをリブートするまで、あるいは補助コンソールの選択を解除するまで、そのデバイスは補助コンソールのままです。ただし、`consadm` コマンドには、システムリブート時にデバイスを補助コンソールとして設定するオプションがあります。設定手順については、次を参照してください。

対話式ログインセッション時の `consadm` コマンドの使用

シリアルポートに接続された端末を使用してシステムにログインし、対話式ログインセッションを実行し、`consadm` コマンドで端末からのコンソールメッセージを表示したい場合、次の動作に注意してください。

- 補助コンソールが有効のままで端末を対話式ログインセッション用に使用する場合、コンソールメッセージは `/dev/sysmsg` または `/dev/msglog` デバイスに送信されます。
- 端末上でコマンドを入力する場合は、入力是对話式セッションに送信されます。デフォルトのコンソール (`/dev/console`) には送信されません。
- `init` コマンドを実行して実行レベルを変更する場合、リモートコンソールソフトウェアは対話式セッションを終了し、`sulogin` プログラムを実行します。この時点で、端末からの入力だけが受け入れられ、コンソールデバイスから入力されたように扱われます。これによって、45ページの「実行レベル変更時の補助コンソールメッセージの使用」に記述されているように、`sulogin` プログラムにパスワードを入力できるようになります。

次に、正しいパスワードを (補助) 端末上で入力すると、補助コンソールは対話式 `sulogin` セッションを実行して、デフォルトのコンソールと他の競合する補助コンソールをロックアウトします。つまり、端末が本質的にシステムコンソールとして機能することを意味します。

- この時点から実行レベルを 3 に変更したり、別の実行レベルに変更できます。実行レベルを変更すると、`sulogin` は再びすべてのコンソールデバイス上で動作します。終了するか、システムを実行レベル 3 で起動することを指定した場合、すべての補助コンソールは入力機能を失います。つまり、コンソールメッセージ用の表示デバイスに戻ります。

システムが起動するときは、デフォルトのコンソールデバイス上で `rc` スクリプトに情報を提供しなければなりません。システムが起動した後、`login` プログラムがシリアルポート上で動作するため、別の対話式セッションにログインできるようになります。デバイスを補助コンソールとして指定した場合、コンソール

メッセージは端末上に引き続き表示されます。しかし、端末からのすべての入力は対話式セッションに送信されます。

▼ 補助 (リモート) コンソールを有効にするには

`consadm` コマンドで補助コンソールを追加するまで、`consadm` デーモンはポートの監視を開始しません。セキュリティ上、コンソールメッセージがリダイレクトされるのはキャリア信号が失われるか、あるいは補助コンソールデバイスが選択解除されるまでのみです。つまり、`consadm` コマンドを正常に使用するためには、ポート上でキャリアが確立されていなければならないことを意味します。

補助コンソールを有効にする方法の詳細は、`consadm(1M)` のマニュアルページを参照してください。

1. スーパーユーザーとしてシステムにログインします。
2. 補助コンソールを有効にします。

```
# consadm -a devicename
```

3. 現在の接続が補助コンソールであることを確認します。

```
# consadm
```

例 — 補助 (リモート) コンソールを有効にする

```
# consadm -a /dev/term/a
# consadm
/dev/term/a
```

▼ 補助 (リモート) コンソールのリストを表示するには

1. スーパーユーザーとしてシステムにログインします。
2. 次の手順のいずれかを選択します。
 - a. 補助コンソールのリストを表示します。

```
# consadm
/dev/term/a
```

- b. 固定されている補助コンソールのリストを表示します。

```
# consadm -p
/dev/term/b
```

▼ システムリブート時に補助 (リモート) コンソールを有効にするには

1. スーパーユーザーとしてシステムにログインします。
2. システムリブート時に補助コンソールを有効にします。

```
# consadm -a -p devicename
```

固定補助コンソールのリストにデバイスが追加されます。

3. 固定補助コンソールのリストにデバイスが追加されていることを確認します。

```
# consadm
```

例 — システムリブート時に補助 (リモート) コンソールを有効にする

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ 補助 (リモート) コンソールを無効にするには

1. スーパーユーザーとしてシステムにログインします。
2. 次の手順のいずれかを選択します。
 - a. 補助コンソールを無効にします。

```
# consadm -d devicename
```

- b. 補助コンソールを無効にして、固定補助コンソールのリストから削除します。

```
# consadm -p -d devicename
```

3. 補助コンソールが無効になっていることを確認します。

```
# consadm
```

例 — 補助 (リモート) コンソールを無効にする

```
# consadm -d /dev/term/a
# consadm
```

改良されたシステムブートとエラーメッセージ形式

この機能は、Solaris 7-3/99 ソフトウェアリリースで追加されたものです。

この情報は、『Solaris のシステム管理 (第 2 巻)』の「システムのメッセージの表示」と、『Solaris 移行ガイド』にあるシステムブートとエラーメッセージに関する追加説明です。

Solaris 7-3/99 リリースではシステムブートとエラーメッセージ形式が改良され、syslog ロギング機能が生成するメッセージに、数値による識別子、モジュール名、およびタイムスタンプが追加されるようになりました。さらに、システムのパニックやリブートの後に以前は失われていたメッセージが、保存されるようになりました。

新しいメッセージ形式を有効または無効にするには、log.conf ファイルで msgid プロパティを設定します。新しいメッセージ形式はデフォルトでは有効になっていません。新しいメッセージ形式の詳細は、log(7D) のマニュアルページと次の手順を参照してください。

システムエラーロギングの一般情報については、『Solaris のシステム管理 (第 2 巻)』を参照してください。

システムブートとエラーメッセージ形式の変更

log.conf ファイルで msgid が 0 に設定されている場合は、メッセージ形式に変更はありません。msgid が 1 に設定されている場合は、メッセージ形式に次の 2 つの変更があります。

1. メッセージテキストの前に、次のようなメッセージ ID が付きます。

```
[ID msgid facility.priority]
```

たとえば、次のようになります。

```
[ID 123456 kern.notice]
```

msgid 識別子については、msgid(1M) のマニュアルページを参照してください。facility 識別子と priority 識別子については、syslog.conf(4) のマニュアルページを参照してください。

2. メッセージの原因がカーネルにある場合は、「unix」だけではなく、カーネルモジュール名が表示されます。

以前のメッセージ形式は、次のようになっていました。

```
Oct 1 14:07:24 mars unix: alloc: /: file system full
```

新しいメッセージ形式では、次のようになります。

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

▼ システムメッセージ ID を有効にするには

1. スーパーユーザーになります。
2. システムメッセージ ID を有効にするには、`/platform/`uname -i`/kernel/drv/log.conf` ファイルに次の行を追加します。このファイルがない場合は、`/kernel/drv/log.conf` ファイルに `msgid` プロパティを追加します。

```
msgid=1
```

3. ファイルを保存して閉じます。
4. 次のコマンドでシステムをリブートします。

```
# init 6
```

注 - システムをリブートせずにシステムメッセージ ID を有効にするには、次の `adb` コマンドを使用します。

```
# echo log_msgid/W1 | adb -kw
```

▼ システムメッセージ ID を無効にするには

1. スーパーユーザーになります。

2. システムメッセージ ID を無効にするに

は、`/platform/`uname -i`/kernel/drv/log.conf` ファイルの `msgid` 行を次のように変更します。このファイルがない場合は、`/kernel/drv/log.conf` ファイルの `msgid` プロパティを変更します。

```
msgid=0
```

3. ファイルを保存して閉じます。

4. 次のコマンドでシステムをリブートします。

```
# init 6
```

注 - システムをリブートせずにシステムメッセージ ID を無効にするには、次の `adb` コマンドを使用します。

```
# echo log_msgid/W0 | adb -kw
```

システム管理の注意事項

この章では、次の新しいシステム管理の情報について説明します。

- 53ページの「BIND 8.1 への移行」
- 53ページの「AnswerBook2 サーバーソフトウェアの更新」
- 54ページの「SPARC: Power Management ソフトウェアでの cron ジョブの実行」

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 7-11/99 のマニュアルページには、「*Solaris 7 Reference Manual Collection*」には記載されていない新しい情報が提供されています。

BIND 8.1 への移行

Solaris 7-8/99 ソフトウェアリリースで、『Solaris ネーミングの設定と構成』と『Solaris ネーミングの管理』の「BIND 4.9.x から BIND 8.1 への移行」セクションが更新されました。シェルスクリプト名とパス名が `/etc/bin/named-boot2conf` から `/usr/sbin/named-bootconf` に変更されました。

AnswerBook2 サーバーソフトウェアの更新

この機能は、Solaris 7-8/99 ソフトウェアリリースで更新されました。

AnswerBook2 バージョン 1.4.1 サーバーソフトウェアでは、より向上した性能、(グラフィカルではなく) テキストによるナビゲーションインタフェース、およびコレクションごとに情報を表示して検索する機能が提供されます。

AnswerBook2 インタフェースの変更の詳細は、『AnswerBook2 を使ったオンライン文書の表示』を参照してください。

SPARC: Power Management ソフトウェアでの cron ジョブの実行

この機能は、Solaris 7-3/99 ソフトウェアリリースで追加されたものです。

この情報は、『電源管理システム ユーザーマニュアル』の「電源管理システムの使用」に記載されている cron ジョブに関する情報よりも優先されます。

Solaris 7 リリース以降、Power Management ソフトウェアによりシステムが一時停止している時間内に実行するようスケジュールされている cron ジョブは、次のように処理されます。

- 1 番目の cron ジョブ (システムが一時停止している時間内で、一番先に実行するようにスケジュールされていた cron ジョブ) は、システムが再開すると同時に起動されます。
- 2 番目以降の cron ジョブ (システムが一時停止している時間内で、1 番目の cron ジョブより後で実行するようにスケジュールされていた cron ジョブ) は、まったく起動されません。

通常、cron コマンドは、次に実行するジョブを識別して、そのジョブを実行するまでに待機する時間間隔を計算します。システムが再開すると、cron コマンドは、最初のジョブが起動されるはずだった時刻までの時間間隔を計算しますが、この時間間隔は負の値になるため、すでに待ち行列に入っているこの cron ジョブをすぐに実行します。

次に、cron はジョブの待ち行列をチェックし直します。cron は実行時刻がまだ過ぎていないジョブだけを対象とするため、この時点でまだ待ち行列に残っているジョブのうちシステムが一時停止していた間に実行時刻を迎えているジョブは実行されません。