

SPARC プラットフォームでの **Oracle®** **Solaris** のブートおよびシャットダウン

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

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

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

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

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

目次

はじめに	7
1 SPARC ベースシステムのブートおよびシャットダウン (概要)	13
システムのブートおよびシャットダウンの新機能	14
管理用に提供された <code>driver.conf</code> ファイル	14
SPARC プラットフォームでの高速リブート	14
SPARC ベースシステムのブートおよびシャットダウン (トピックマップ)	15
システムのブートに関するガイドライン	16
システムをブートする理由	16
サービス管理機構とブート	17
SMF 使用時の動作に関する変更	18
実行レベルの動作	18
システムがマルチユーザー状態 (実行レベル 3) にブートされたときの処理	20
実行レベルとマイルストーンの使い分け	20
Oracle Solaris ブートアーキテクチャーの概要	20
SPARC ブートプロセスの説明	21
SPARC のブートフェーズ	22
2 指定された状態への SPARC ベースシステムのブート (手順)	25
指定された状態への SPARC ベースシステムのブート (タスクマップ)	25
指定された状態への SPARC ベースシステムのブート	26
システムの現在の実行レベルの確認	26
マルチユーザー状態 (実行レベル 3) への SPARC ベースシステムのブート	27
シングルユーザー状態 (実行レベル S) への SPARC ベースシステムのブート	28
SPARC ベースシステムの対話式でのブート	29

3 システムのシャットダウン(手順)	33
システムのシャットダウン(タスクマップ)	33
システムのシャットダウンの概要	34
システムのシャットダウンに関するガイドライン	34
システムシャットダウンコマンド	35
システムのシャットダウン	36
▼システムにログインしているユーザーを確認する方法	36
▼ shutdown コマンドを使用してシステムをシャットダウンする方法	37
▼ init コマンドを使用してシステムをシャットダウンする方法	39
システムデバイスの電源の切断	40
 4 SPARC ベースシステムのリブート(手順)	41
SPARC ベースシステムのリブート(タスクマップ)	41
SPARC ベースシステムのリブート	42
▼ init コマンドを使用してシステムをリブートする方法	43
▼ reboot コマンドを使用してシステムをリブートする方法	44
SPARC ベースシステムのリブート処理の高速化	44
▼ SPARC システムの高速リブートを開始する方法	44
高速リブート機能のデフォルト動作の変更	45
高速リブートが有効化済みのシステムの標準リブートの開始	45
 5 SPARC ベースシステムのネットワークからのブート(手順)	47
SPARC ベースシステムのネットワークからのブート(作業マップ)	47
SPARC システムをネットワークからブートする	48
SPARC ネットワークブートプロセス	48
SPARC ベースシステムをネットワークからブートするための要件	49
OpenBoot PROM へのネットワークブート引数の設定	49
DHCP を使用して自動ブートするための NVRAM 別名の設定	51
▼ SPARC ベースシステムをネットワークからブートする方法	51
 6 SPARC ベースシステムでのブートパラメータの変更(手順)	53
SPARC ベースシステムでのブートパラメータの変更(タスクマップ)	54
SPARC ベースシステムでのブートパラメータの変更	55
▼ システムの PROM リビジョン番号を確認する方法	55

▼ システム上のデバイスを確認する方法	55
▼ デフォルトのブートデバイスを調べる方法	57
▼ ブート PROM を使用してデフォルトのブートデバイスを変更する方法	58
▼ eeprom ユーティリティを使用してデフォルトのブートデバイスを変更する方 法	59
▼ ブート PROM を使用してデフォルトのブートファイルを変更する方法	60
▼ eeprom ユーティリティを使用してデフォルトのブートファイルを変更する方 法	60
7 SPARC プラットフォームでの ZFS ブート環境の作成と管理、およびこのブート環境から のブート (手順)	61
ZFS ブート環境の作成と管理、およびこのブート環境からのブート (タスク マップ)	62
ブート環境の作成と管理	63
▼ 新しいブート環境を作成する方法	63
▼ ブート環境のスナップショットの作成方法	65
▼ 既存のスナップショットからブート環境を作成する方法	65
▼ 新しく作成されたブート環境をアクティブにする方法	65
▼ 使用可能なブート環境、スナップショット、およびデータセットの一覧を表示す る方法	66
▼ ブート環境を破棄する方法	67
SPARC プラットフォームの ZFS ブート環境からのブート	68
▼ SPARC: ブートシーケンス中に使用可能なブート環境の一覧を表示する方法	69
▼ ZFS ブート環境またはルートファイルシステムからブートする方法	70
8 SPARC ベースシステムのブート可能状態の維持 (手順)	73
SPARC ベースシステムのブート可能状態の維持 (タスクマップ)	73
Oracle Solaris ブートアーカイブの説明	74
SPARC ブートアーカイブの場所と内容に関する情報の取得	74
ブートアーカイブ SMF サービスの管理	76
boot-archive サービスが実行中かどうかの判定	76
▼ boot-archive SMF サービスを有効または無効にする方法	77
ブートアーカイブの整合性の維持	77
▼ ブートアーカイブを手動で更新することによってブートアーカイブの自動更新障 害を解決する方法	78

9	SPARC ベースシステムのブートのトラブルシューティング (手順)	79
	SPARC ベースシステムのブートのトラブルシューティング (タスクマップ)	80
	復旧を目的とした SPARC ベースシステムのシャットダウンおよびブート	81
	システムの復旧目的の停止とブート	81
	SPARC ベースシステムのクラッシュダンプとリブートの強制実行	85
	▼ カーネルデバッガ (kmdb) を有効にしてシステムをブートする方法	87
	索引	89

はじめに

『SPARC プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』は、Oracle Solaris システム管理に関する重要な情報を提供するドキュメントセットの一部です。このマニュアルには、SPARC プラットフォームについての情報が記載されています。

本書は、読者が次のタスクを終了済みであることを前提としています。

- Oracle Solaris 11 をインストール済み
- 使用する予定のすべてのネットワークソフトウェアを設定済み

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

サポートされるシステムについては、[Oracle Solaris OS: Hardware Compatibility Lists](#)を参照してください。

対象読者

このマニュアルは、Oracle Solaris 11 リリースを実行している 1 つまたは複数のシステムの管理を行うユーザーを対象にしています。本書を使用するには、UNIX のシステム管理について 1 - 2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Solaris システム管理マニュアルセットの構成

システム管理ガイドセットに含まれる各ガイドとその内容は、次のとおりです。

マニュアルのタイトル	トピック
『SPARC プラットフォームでの Oracle Solaris のブートおよびシャットダウン』	システムのブートおよびシャットダウン、ブートサービスの管理、ブート動作の変更、ZFS からのブート、ブートアーカイブの管理、および SPARC プラットフォーム上でのブートのトラブルシューティング
『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』	システムのブートおよびシャットダウン、ブートサービスの管理、ブート動作の変更、ZFS からのブート、ブートアーカイブの管理、および x86 プラットフォーム上でのブートのトラブルシューティング
『Oracle Solaris の管理: 一般的なタスク』	Oracle Solaris コマンドの使用、システムのブートとシャットダウン、ユーザーアカウントとグループの管理、サービスの管理、ハードウェア障害、システム情報、システムリソース、およびシステム性能、ソフトウェアの管理、印刷、コンソールと端末、およびシステムとソフトウェアの問題の障害追跡
『Oracle Solaris の管理: デバイスとファイルシステム』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Oracle Solaris の管理: IP サービス』	TCP/IP ネットワーク管理、IPv4 および IPv6 アドレスの管理、DHCP、IPsec、IKE、IP フィルタ、および IPQoS
『Oracle Solaris Administration: Naming and Directory Services』	DNS、NIS、および LDAP ネームサービスおよびディレクトリサービス (NIS から LDAP への移行を含む)
『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』	WiFi ワイヤレスを含む自動および手動の IP インタフェース構成、ブリッジ、VLAN、集積体、LLDP、および IPMP の管理、仮想 NIC とリソース管理
『Oracle Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と autofs)、メール、SLP、および PPP
『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』	リソース管理機能 (利用可能なシステムリソースをアプリケーションが使用する方法を制御できる)、Oracle Solaris ゾーンソフトウェア区分技術 (オペレーティングシステムのサービスを仮想化してアプリケーション実行用の隔離された環境を作成する)、および Oracle Solaris 10 ゾーン (Oracle Solaris 11 カーネル上で実行される Oracle Solaris 10 環境をホストする)
『Oracle Solaris の管理: セキュリティーサービス』	監査、デバイス管理、ファイルセキュリティー、BART、Kerberos サービス、PAM、暗号化フレームワーク、キー管理フレームワーク、特権、RBAC、SASL、Secure Shell、およびウイルススキャン。

マニュアルのタイトル	トピック
『Oracle Solaris Administration: SMB and Windows Interoperability』	SMB クライアントが SMB 共有を使用できるように Oracle Solaris システムを構成できる SMB サービス、SMB 共有にアクセスできる SMB クライアント、およびユーザーとグループの識別情報を Oracle Solaris システムと Windows システムの間でマップできるネイティブな識別情報マッピングサービス。
『Oracle Solaris の管理: ZFS ファイルシステム』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Oracle Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Trusted Extensions 構成と管理』	Trusted Extensions に固有のシステムのインストール、構成、および管理
『Oracle Solaris 11 セキュリティーガイドライン』	Oracle Solaris システムのセキュリティー保護、およびゾーン、ZFS、Trusted Extensions などのセキュリティー機能の使用シナリオ
『Oracle Solaris 10 から Oracle Solaris 11 への移行』	インストール、デバイス、ディスク、およびファイルシステムの管理、ソフトウェア管理、ネットワーク接続、システム管理、セキュリティー、仮想化、デスクトップ機能、ユーザーアカウント管理、およびユーザー環境といった分野における、システム管理の情報および Oracle Solaris 10 から Oracle Solaris 11 への移行例を提供します

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <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。
「 」	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。	sun% grep '^#define \ XV_VERSION_STRING'

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

- C シェル
machine_name% **command y|n** [*filename*]
- C シェルのスーパーユーザー
machine_name# **command y|n** [*filename*]
- Bash シェル、Korn シェル、および Bourne シェル
\$ **command y|n** [*filename*]
- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー
command y|n [*filename*]

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

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

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

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

一般規則

このドキュメントでは次の規則が使用されています。

- このドキュメント中の手順を実行したり、例(コマンド入力、コードなど)を使用する場合には、二重引用符(")、左一重引用符(')、右一重引用符(')をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。
- root パスには通常、/usr/sbin、/usr/bin、および/etc ディレクトリが含まれているため、このマニュアルの手順ではこれらのディレクトリ内のコマンドを絶対パス名なしで表記します。ただし、それ以外のあまり一般的でないディレクトリにあるコマンドについては、このドキュメント中の例では絶対パスで表記します。

SPARC ベースシステムのブートおよびシャットダウン(概要)

Oracle Solaris は、データベースや Web サービスなどのエンタープライズサービスが可能ながざりいつでも利用できるようにするために、停止することなく動作するように設計されています。この章では、SPARC ベースのシステムのシャットダウンとブートのガイドラインについて説明します。

注- このマニュアルでは主に、サーバーおよびワークステーション上の単一の Oracle Solaris インスタンスのブートおよびシャットダウンに焦点を合わせています。サービスプロセッサを持つシステムや、複数の物理ドメインを持つシステム上での Oracle Solaris のブートおよびシャットダウンについての情報は、このドキュメントでは詳細には記載されていません。詳細は、<http://www.oracle.com/technetwork/indexes/documentation/index.html> にある特定のハードウェアについての製品ドキュメントを参照してください。

この章の内容は次のとおりです。

- 14 ページの「システムのブートおよびシャットダウンの新機能」
- 15 ページの「SPARC ベースシステムのブートおよびシャットダウン(トピックマップ)」
- 16 ページの「システムのブートに関するガイドライン」
- 17 ページの「サービス管理機構とブート」
- 18 ページの「実行レベルの動作」
- 20 ページの「Oracle Solaris ブートアーキテクチャーの概要」

x86 ベースシステムのブートについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』を参照してください。

システムのブートおよびシャットダウンの新機能

Oracle Solaris 11 リリースで追加されたブート機能は次のとおりです。

管理用に提供された **driver.conf** ファイル

/kernel および /platform ディレクトリ内にあるベンダー提供の元のファイルを変更することなく、ドライバ構成ファイル (driver.conf) によって管理上のローカルな変更を補うことができます。この拡張機能により、システムのアップグレード中にローカルな構成を保持しやすくなります。新しい /etc/driver/drv ディレクトリに driver.conf ファイルを追加することにより、ドライバ構成にローカルな変更を加えることができるようになりました。ブート時に、システムは /etc/driver/drv にそのドライバの構成ファイルがないかチェックします。見つかった場合、システムはベンダー提供の構成と、管理用に提供された変更を自動的にマージします。

マージされたプロパティを表示するには、prtconf コマンドに新しい -u オプションを付けて使用します。-u オプションによって、指定されたドライバについての元のプロパティ値と更新されたプロパティ値の両方を表示できます。詳細は、[prtconf\(1M\)](#) のマニュアルページを参照してください。手順については、『Oracle Solaris の管理: 一般的なタスク』の「[デバイスのデフォルトのプロパティ値とカスタマイズされたプロパティ値を表示する方法](#)」を参照してください。

注 - /kernel および /platform ディレクトリにあるベンダー提供の driver.conf ファイルを編集しないでください。ドライバの構成を補完する必要がある場合は、対応する driver.conf ファイルをローカルの /etc/driver/drv ディレクトリに追加してから、そのファイルをカスタマイズする方法を推奨します。手順については、『Oracle Solaris の管理: デバイスとファイルシステム』の第 5 章「[デバイスの管理 \(概要と手順\)](#)」を参照してください。

次の補足情報も参照してください。

- [driver.conf\(4\)](#)
- [driver\(4\)](#)
- [『Writing Device Drivers』](#)
- [ddi_prop_exists\(9F\)](#)
- [ddi_prop_lookup\(9F\)](#)

SPARC プラットフォームでの高速リブート

SPARC プラットフォーム上で高速リブートを統合すると、reboot コマンドで -f オプションが使用できるため、ある程度の POST 検査を省略することでブート処理を高速化できます。

高速リブート機能は Oracle Solaris のサービス管理機構 (SMF) の機能によって管理され、ブート構成サービス `svc:/system/boot-config` によって実装されます。
`boot-config` サービスは、デフォルトのブート構成パラメータを設定または変更する手段を提供します。`config/fastreboot_default` プロパティを `true` に設定すると、システムは高速リブートを自動的に実行するため、`reboot -f` コマンドを使用する必要はありません。SPARC プラットフォームでは、このプロパティの値は `false` に設定されています。タスク関連の情報については、[44 ページの「SPARC ベースシステムのリブート処理の高速化」](#) を参照してください。

注-SPARC 上の高速リブート動作は特定のシステムにのみ適用できます。`sun4v` システムでは、リブートは実際は POST を伴わないハイパーバイザリブートであるため、高速リブートは不要です。

SPARC ベースシステムのブートおよびシャットダウン(トピックマップ)

次の参照情報を使用して、このドキュメント内のさまざまなブート関連のトピックについての手順を見つけてください。

表 1-1 SPARC ベースシステムのブートおよびシャットダウン:トピックマップ

タスク	参照先
SPARC ベースシステムを指定された状態にします (実行レベルのブート)。	第 2 章「指定された状態への SPARC ベースシステムのブート (手順)」
SPARC ベースシステムをシャットダウンします。	第 3 章「システムのシャットダウン (手順)」
SPARC ベースシステムをリブートします。	第 4 章「SPARC ベースシステムのリブート (手順)」
SPARC システムをネットワークからブートします。	第 5 章「SPARC ベースシステムのネットワークからのブート (手順)」
SPARC ベースシステムのデフォルトのブート動作を変更します。	第 6 章「SPARC ベースシステムでのブートパラメータの変更 (手順)」
SPARC ベースシステムの ZFS ブート環境、スナップショット、またはデータセットからブートします。	第 7 章「SPARC プラットフォームでの ZFS ブート環境の作成と管理、およびこのブート環境からのブート (手順)」
ブート管理インタフェース (<code>bootadm</code>) を使用することによって、SPARC ベースシステムをブート可能に維持します。	第 8 章「SPARC ベースシステムのブート可能状態の維持 (手順)」

表 1-1 SPARC ベースシステムのブートおよびシャットダウン:トピックマップ (続き)

タスク	参照先
SPARC ベースシステムのブートのトラブルシューティングを行います。	第 9 章「SPARC ベースシステムのブートのトラブルシューティング(手順)」

システムのブートに関するガイドライン

システムをブートするときは、次の点に注意してください。

- SPARC システムをシャットダウンしたあとでブートするには、PROM レベルで boot コマンドを使用します。
- 電源を切断したあとに再投入すればシステムをリブートできます。



注意- この方法は正常な停止とは見なされていません。このシャットダウン方法は、緊急の場合にだけ使用してください。システムサービスやプロセスが突然終了するため、ファイルシステムに損傷を与えることがあります。この種の損傷の修復作業には手間がかかり、さまざまなユーザーファイルやシステムファイルをバックアップコピーから復元する必要がある可能性があります。

システムをブートする理由

次の表に、システム管理タスクとそれに伴って必要となるブートオプションを示します。

表 1-2 システムをブートする理由

システムブートの理由	適切なブートオプション	参照先
停電のためシステムの電源を切断します	システムの電源を再投入します	第 3 章「システムのシャットダウン(手順)」
/etc/system ファイル内のカーネルパラメータを変更します	システムをマルチユーザー状態にリブートします (SMB または NFS リソースを共有する実行レベル 3)	27 ページの「システムをマルチユーザー状態 (実行レベル 3) にブートする方法」
ファイルシステムを保守します (システムデータのバックアップや復元など)	シングルユーザー状態 (実行レベル S) で Ctrl + D キーを押してシステムをマルチユーザー状態 (実行レベル 3) に戻します	28 ページの「システムをシングルユーザー状態 (実行レベル S) にブートする方法」
/etc/system などのシステム構成ファイルを修復します	対話式ブート	30 ページの「システムを対話式でブートする方法」

表 1-2 システムをブートする理由 (続き)

システムブートの理由	適切なブートオプション	参照先
システムにハードウェアを追加します (または、システムからハードウェアを削除する)	再ブート (デバイスがホットプラグ対応でない場合は、デバイスを追加または削除したあとにシステムの電源を入れます)	『Oracle Solaris の管理: デバイスとファイルシステム』の「ZFS ファイルシステム用のディスクの設定 (タスクマップ)」
システムをハング状態から回復させ、クラッシュダンプを強制します。	回復ブート	86 ページの「クラッシュダンプを強制してシステムをリブートする方法」
カーネルデバッガ (kmdb) を使用してシステムをブートし、システムの障害を調査します	kmdb をブートします	87 ページの「カーネルデバッガ (kmdb) を有効にしてシステムをブートする方法」

サービス管理機構とブート

SMF は、従来の UNIX の起動スクリプト、init 実行レベル、および構成ファイルを補強するインフラストラクチャーを提供します。SMF の導入により、ブートプロセスで作成されるメッセージが少なくなりました。デフォルトでは、サービスの起動時にメッセージは表示されません。ブートメッセージによって提供されていた情報は、`/var/svc/log` にある各サービス用のログファイルで提供されるようになりました。ブートの問題の診断には `svcs` コマンドが役立ちます。ブートプロセス中に各サービスが開始されたときにメッセージを生成するには、`boot` コマンドで `-v` オプションを使用します。

システムがブートされるとき、ブート先のマイルストーンを選択したり、記録されるエラーメッセージのレベルを選択したりできます。例:

- 次のコマンドを使用すると、ブート先の特定のマイルストーンを選択できます。

```
ok boot -m milestone=milestone
```

デフォルトのマイルストーンは、有効になっているすべてのサービスを起動する `all` です。ほかの有用なマイルストーンは `none` で、これは `init`、`svc.startd`、および `svc.configd` のみを開始します。このマイルストーンは、サービスを手動で開始できる非常に有用なデバッグ環境を提供します。`none` マイルストーンの使用手順については、[84 ページの「どのサービスも起動しないでシステムをブートする方法」](#)を参照してください。

実行レベルと同等な `single-user`、`multi-user`、および `multi-user-server` も使用できますが、一般的には使用されません。特に `multi-user-server` マイルストーンは、そのマイルストーンに依存しないサービスを開始しないため、重要なサービスを含まないことがあります。

- 次のコマンドを使用すると、`svc.startd` のログレベルを選択できます。

```
ok boot -m logging_level
```

選択できるログレベルは `quiet`、`verbose`、および `debug` です。ログレベルに関する具体的な情報については、『[Oracle Solaris の管理: 一般的なタスク](#)』の「[SMF サービスのエラーログ](#)」を参照してください。

SMF 使用時の動作に関する変更

SMF が提供する機能のほとんどが、ユーザーの目に触れることなく実行されるため、ユーザーは通常これらの機能に気が付きません。それ以外の機能には新しいコマンドでアクセスします。

もっともよく目にする動作の変更の一覧を次に示します。

- ブートプロセスで生成されるメッセージが少なくなりました。デフォルトでは、サービスの起動時にメッセージは表示されません。ブートメッセージによって提供されていた情報は、`/var/svc/log` にある各サービス用のログファイルで提供されるようになりました。ブートの問題の診断には `svcs` コマンドが役立ちます。なお、`boot` コマンドで `-v` オプションを使用すれば、ブートプロセス中に各サービスが起動されるたびにメッセージが生成されます。
- サービスは可能なかぎり自動的に再起動されるため、プロセスの終了が失敗したように見えることがあります。サービスに障害があれば保守モードに切り替わりますが、通常、サービスのプロセスが終了すると、そのサービスは再起動されます。SMF サービスが実行されないようにするには、`svcadm` コマンドを使用してそのプロセスを停止する必要があります。
- `/etc/init.d` および `/etc/rc*.d` 内の多くのスクリプトが削除されました。サービスの有効化および無効化に、これらのスクリプトはもう必要ありません。`/etc/inittab` のエントリも削除され、サービスの管理に SMF が使用できるようになりました。ISV によって提供されるスクリプトおよび `inittab` エントリ、あるいはローカルで開発されたそれらは、従来どおり機能します。各サービスは、ブートプロセス内のまったく同じ時点では開始されない可能性があります。SMF サービスの前に開始されることはありません。

実行レベルの動作

システムの「実行レベル」(「`init` 状態」とも呼ばれる)は、ユーザーが使用できるサービスとリソースを定義します。システムが一度に持つことのできる実行レベルは 1 つだけです。

Oracle Solaris には、次の表で説明する 8 つの実行レベルがあります。デフォルトの実行レベル 3 は、`/etc/inittab` ファイルに指定されています。

表 1-3 Oracle Solaris の実行レベル

実行レベル	init 状態	タイプ	目的
0	電源切断状態	電源切断	オペレーティングシステムをシャットダウンして、システムの電源を安全に落とせるようにします。
s または S	シングルユーザー状態	シングルユーザー	一部のファイルシステムがマウントされ使用可能な状態で、シングルユーザーとして動作します。
1	システム管理状態	シングルユーザー	すべての使用可能なファイルシステムにアクセスします。ユーザーログインは使用できません。
2	マルチユーザー状態	マルチユーザー	通常の運用に使用します。複数のユーザーがシステムとすべてのファイルシステムにアクセスできます。NFS サーバーデーモンを除く、すべてのデーモンが動作します。
3	NFS リソースを共有したマルチユーザーレベル	マルチユーザー	NFS リソースを共有する通常の運用に使用します。これがデフォルトの実行レベルです。
4	マルチユーザー状態 (予備)	マルチユーザー	デフォルトでは構成されていませんが、顧客用に使用できます。
5	電源切断状態	電源切断	オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにします。可能であれば、この機能をサポートしているシステムでは電源を自動的に切断します。
6	リブート状態	リブート	システムをシャットダウンして実行レベル 0 にしたあと、NFS リソースを共有するマルチユーザーレベル (または <code>inittab</code> ファイルに指定されたデフォルトの実行レベル) でリブートします。

また、`svcadm` コマンドを使用してシステムの実行レベルを変更することもできます。その場合は、実行するときのマイルストーンを選択してください。次の表に、各マイルストーンに対応する実行レベルを示します。

表 1-4 実行レベルと SMF マイルストーン

実行レベル	SMF マイルストーンの FMRI
S	<code>milestone/single-user:default</code>

表 1-4 実行レベルと SMF マイルストーン (続き)

実行レベル	SMF マイルストーンの FMRI
2	milestone/multi-user:default
3	milestone/multi-user-server:default

システムがマルチユーザー状態 (実行レベル 3) にブートされたときの処理

1. `init` プロセスが開始され、環境変数を設定するために
`svc:/system/environment:init` SMF サービス内に定義されているプロパティを読み取ります。デフォルトでは、`TIMEZONE` 変数だけが設定されます。
2. `init` は `inittab` ファイルを読み取り、次の処理を行います。
 - a. `action` フィールドが `sysinit` になっているすべてのプロセスエントリを実行して、ユーザーがシステムにログインする前に特殊な初期化がすべて行われるようにします。
 - b. 起動アクティビティを `svc.startd` に渡します。

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

実行レベルとマイルストーンの使い分け

一般に、マイルストーンまたは実行レベルを変更する手順はめったに実行されません。必要な場合、実行レベルを変更するために `init` コマンドを使用するとマイルストーンも変更され、これは使用するのに適切なコマンドです。 `init` コマンドはまた、システムをシャットダウンするのにも適しています。

ただし、`none` マイルストーンを使用してシステムをブートすると、起動時の問題をデバッグするときに非常に役立つ場合があります。 `none` マイルストーンと同等の実行レベルはありません。詳細は、[84 ページの「どのサービスも起動しないでシステムをブートする方法」](#)を参照してください。

Oracle Solaris ブートアーキテクチャーの概要

Oracle Solaris SPARC ブートアーキテクチャーには次の基本特性が含まれています。

- ブートアーカイブの使用
ブートアーカイブとは、システムのブートに必要なすべてのファイルが含まれている RAM ディスクイメージのことです。

- ブート管理インタフェースを使用した **Oracle Solaris** ブートアーカイブの整合性の維持

`bootadm` コマンドは、ブートアーカイブの更新と検証を細部にわたって処理します。インストールまたはアップグレードの間、`bootadm` コマンドは初期ブートアーカイブを作成します。通常システムシャットダウンの処理中に、シャットダウンプロセスではブートアーカイブの内容がルートファイルシステムと比較されます。システムにドライバや構成ファイルなどの更新がある場合、ブートアーカイブがこれらの変更を含むように再構築されることによって、リブート時にブートアーカイブとルートファイルシステムの同期がとられます。`bootadm` コマンドを使用して、ブートアーカイブを手動で更新することができます。手順については、77 ページの「ブートアーカイブの整合性の維持」を参照してください。

注 - 一部の `bootadm` コマンドオプションは SPARC プラットフォームに適用されません。

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

- インストール中のルートファイルシステムとしての **RAM** ディスクイメージの使用

このプロセスは、SPARC および x86 の両方のプラットフォームで同じです。RAM ディスクイメージは、ブートアーカイブから生成されたあと、ブートデバイスからシステムに転送されます。

注 - SPARC プラットフォームでは、OpenBoot PROM を引き続き使用してブートデバイスにアクセスし、ブートアーカイブをシステムのメモリーに転送します。

ソフトウェアをインストールする場合、RAM ディスクイメージは、インストールプロセス全体に使用されるルートファイルシステムになります。RAM ディスクイメージを使用するとブートプロセスが速くなります。これは、Oracle Solaris とドライバおよび必要なアプリケーションが、リムーバブルメディアから 1 回読み込まれてメモリーに配置されるためです。その後システムは、RAM ディスクに基づいてインストールプロセスを実行します。RAM ディスクファイルシステムのタイプには、HSFS (High Sierra File System) を指定できます。

SPARC ブートプロセスの説明

このセクションでは Oracle Solaris SPARC プラットフォーム上の基本的なブートプロセスについて説明します。サービスプロセッサを持つシステムや、複数の物理ドメインを持つシステムなど、特定のハードウェアタイプ上のブートプロセスについての詳細は、<http://www.oracle.com/technetwork/indexes/documentation/index.html> にある特定ハードウェアについての製品ドキュメントを参照してください。

スタンドアロンプログラムの読み込みと実行のプロセスをブートストラップと呼びます。通常、スタンドアロンプログラムはオペレーティングシステムカーネルです。ただし、カーネル以外の任意のスタンドアロンプログラムをブートできます。

SPARC プラットフォームでは、ブートストラッププロセスは次の基本フェーズで構成されます。

- システムをオンに設定したあと、システムファームウェア (PROM) は電源投入時自己診断 (POST) を実行します。
- 診断テストが正常に完了したあと、マシンのファームウェアが使用する不揮発性記憶領域に適切なフラグが設定されていれば、ファームウェアは自動ブートを試みます。
- 第2レベルのプログラムは、ディスクからブートする場合はファイルシステム固有のブートブロック、ネットワーク経由または自動インストーラ (AI) ユーティリティを使用してブートする場合は `inetboot` と `wanboot` のいずれかです。

ネットワークブートプロセスは次のとおりです。

- 最初に、クライアントは二次ブーターをロードするために必要な IP アドレスおよびその他のパラメータを取得します。
- 次に、二次ブーターがブートデバイスからブートアーカイブをロードします。

SPARC ベースシステムをネットワークからブートする場合の詳細については、[第5章「SPARC ベースシステムのネットワークからのブート\(手順\)」](#)を参照してください。

SPARC のブートフェーズ

Oracle Solaris 10 リリース以降では、SPARC プラットフォームのブートプロセスは、x86 プラットフォームとの共通点を増やすために変更されて拡張されました。

次の4つのブートフェーズは、互いに独立しています。

1. オープンブート **PROM** フェーズ

SPARC プラットフォーム上でのブートプロセスのオープンブート PROM (OBP) フェーズは変更されていません。

ディスクデバイスの場合、ファームウェアドライバは通常 OBP ラベルパッケージの「読み込み」メソッドを使用します。このメソッドは、ディスクの先頭にある VTOC ラベルを解析して、指定されたパーティションを見つけます。そして、パーティションのセクター 1 - 15 がシステムのメモリーに読み込まれます。この領域は一般にブートブロックと呼ばれ、通常はファイルシステムリーダーが格納されています。

2. ブーターフェーズ

このフェーズの間に、ブートアーカイブが読み取られ、実行されます。ブートファイルシステムの形式を認識しておく必要があるのは、ブートプロセスのこのフェーズだけです。ブートローダーとブートアーカイブの転送に使用されるプロトコルには、ローカルディスクアクセス、NFS、およびHTTPがあります。

3. RAM ディスクフェーズ

RAM ディスクは、Oracle Solaris のインスタンスのブートに必要なカーネルモジュールおよびその他のコンポーネントで構成されるブートアーカイブです。

4. カーネルフェーズ

カーネルフェーズは、ブートプロセスの最終段階です。このフェーズの間に、Oracle Solaris が初期化され、ブートアーカイブから構築された RAM ディスク上に最小ルートファイルシステムがマウントされます。インストールなどの一部の環境では、RAM ディスクはルートファイルシステムとして使用され、マウントされたままになります。RAM ディスクには、指定されたルートデバイス上にルートファイルシステムをマウントするために十分な1組のカーネルファイルおよびドライバが格納されます。

そして、カーネルはブートアーカイブから主モジュールの残りを抽出し、それ自体を初期化し、実際のルートファイルシステムをマウントして、ブートアーカイブを破棄します。

指定された状態への SPARC ベースシステムのブート (手順)

この章では、SPARC ベースシステムを、実行レベルとしても知られるさまざまなシステム状態にブートするためのタスク関連情報を提供します。

この章の内容は次のとおりです。

- 25 ページの「指定された状態への SPARC ベースシステムのブート (タスクマップ)」
- 26 ページの「指定された状態への SPARC ベースシステムのブート」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン (概要)」を参照してください。

指定された状態への x86 ベースシステムのブートについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 2 章「指定された状態への x86 ベースシステムのブート (タスク)」を参照してください。

指定された状態への **SPARC** ベースシステムのブート (タスクマップ)

表 2-1 指定された状態への SPARC ベースシステムのブート:タスクマップ

タスク	説明	参照先
システムの現在の実行レベルを確認します。	who コマンドと -r オプションを使用して、システムの現在の実行レベルを確認します。	26 ページの「システムの現在の実行レベルの確認」

表 2-1 指定された状態への SPARC ベースシステムのブート:タスクマップ (続き)

タスク	説明	参照先
SPARC ベースシステムをマルチユーザー状態にブートします。	このブート方法は、シャットダウン後またはシステムのハードウェア保守タスクの実行後にシステムをマルチユーザー状態 (実行レベル 3) に戻す場合に使用します。	27 ページの「マルチユーザー状態 (実行レベル 3) への SPARC ベースシステムのブート」
SPARC ベースシステムをシングルユーザー状態にブートします。	このブート方法は、ファイルシステムのバックアップなど、システムの保守タスクを実行する場合に使用します。	28 ページの「シングルユーザー状態 (実行レベル S) への SPARC ベースシステムのブート」
SPARC システムを対話式でブートします。	このブート方法は、テストのためにシステムファイルまたはカーネルを一時的に変更したあとで使用します。	29 ページの「SPARC ベースシステムの対話式でのブート」

指定された状態への SPARC ベースシステムのブート

次の手順では、SPARC ベースシステムを、ok PROM プロンプトから指定された状態にブートする (実行レベルのブート とも呼ばれる) 方法について説明します。これらの手順では、特に指示がない限り、システムが正常な状態でシャットダウンしていることを前提とします。

システムの現在の実行レベルの確認

実行中のシステムで現在の実行レベルを確認するには、`who -r` コマンドを使用します。

例 2-1 システムの実行レベルを確認する

`who -r` コマンドの出力には、システムの現在の実行レベルと、以前の実行レベルに関する情報が表示されます。

```
$ who -r
.      run-level 3   Dec 13 10:10  3   0 S
$
```

<code>who -r</code> コマンドの出力	説明
<code>run-level 3</code>	現在の実行レベル
<code>Dec 13 10:10</code>	実行レベルが最後に変更された日時

例 2-1 システムの実行レベルを確認する (続き)

who -r コマンドの出力	説明
3	現在の実行レベル
0	最後にリブートしてからシステムがこの実行レベルになった回数
5	以前の実行レベル

マルチユーザー状態 (実行レベル 3) への SPARC ベースシステムのブート

システムの電源を切ってから入れ直すと、マルチユーザーのブートシーケンスが開始されます。

who -r コマンドを使って、システムが指定した実行レベルになっていることを確認します。26 ページの「システムの現在の実行レベルの確認」を参照してください。

- ▼ システムをマルチユーザー状態 (実行レベル 3) にブートする方法
- この手順を使用して、現時点で実行レベル 0 になっている SPARC ベースシステムを実行レベル 3 にブートします。

- 1 システムを **ok PROM** プロンプトにします。
- 2 システムを実行レベル 3 でブートします。

ok boot

自動ブート処理によって、一連の起動メッセージが表示され、システムが実行レベル 3 になります。詳細は、**boot(1M)** のマニュアルページを参照してください。

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

hostname console login:

例 2-2 マルチユーザー状態 (実行レベル 3) へのシステムのブート

次の例は、システムを実行レベル 3 でブートしたときに表示されるメッセージを示しています。

```
ok boot
Probing system devices
Probing memory
```

```
ChassisSerialNumber FN62030249
Probing I/O buses

.
.
.
.
OpenBoot 4.30.4.a, 8192 MB memory installed, Serial #51944031.
Ethernet address 0:3:ba:18:9a:5f, Host ID: 83189a5f.
Rebooting with command: boot
Boot device: /pci@1c,600000/scsi@2/disk@0,0:a File and args:
SunOS Release 5.11 Version fips_checksum nightly 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
DEBUG enabled
misc/forthdebug (455673 bytes) loaded
Hardware watchdog enabled
Hostname: portia-123
NIS domain name is solaris.us.oracle.com

portia-123 console login: NIS domain name is solaris.us.oracle.com
```

シングルユーザー状態 (実行レベル S) への SPARC ベースシステムのブート

システムのシングルユーザー状態へのブートは、ファイルシステムのバックアップや、システムに関するその他の問題のトラブルシューティングなどのシステム保守に使用されます。

▼ システムをシングルユーザー状態 (実行レベル S) にブートする方法

- 1 システムを **ok PROM** プロンプトにします。
- 2 システムを実行レベル **S** でブートします。
`ok boot -s`
- 3 次のメッセージが表示されたら、**root** のパスワードを入力します。
`SINGLE USER MODE`
`Root password for system maintenance (control-d to bypass): xxxxxx`
- 4 システムが実行レベル **S** になっていることを確認します。
`# who -r`
- 5 実行レベル **S** への変更が必要な保守タスクを実行します。
- 6 システム保守タスクが完了したら、**Ctrl+D** キーを押してシステムをマルチユーザー状態にします。

例 2-3 SPARC: シングルユーザー状態 (実行レベル S) へのシステムのブート

次の例は、システムを実行レベル S でブートしたときに表示されるメッセージを示しています。

```
ok boot -s
SC Alert: Host System has Reset
Enter #. to return to ALOM.
cpu Device: pci
Device: ebus
/ebus@800: serial
Device: pci
/pci@780: Device 0 Nothing there
/pci@7c0: Device 0 pci
/pci@7c0/pci@0: Device 4 network network
/pci@7c0/pci@0: Device 8 pci
/pci@7c0/pci@0/pci@8: Device 1 network network
/pci@7c0/pci@0/pci@8: Device 2 scsi tape disk

Sun Fire(TM) T1000, No Keyboard
Copyright 2008 ... All rights reserved.
OpenBoot 4.30.0.build_12***PROTOTYPE BUILD***, 2000 MB memory available,
Serial #69312178.
Ethernet address 0:14:4f:21:9e:b2, Host ID: 84219eb2.

Boot device: /pci@7c0/pci@0/pci@8/scsi@2/disk@0,0:a File and args:
zfs-file-system
Loading: /platform/SUNW,Sun-Fire-T1000/boot_archive
ramdisk-root hsfs-file-system
Loading: /platform/SUNW,Sun-Fire-T1000/kernel/sparcv9/unix
SunOS Release 5.11 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.0.build_12***PROTOTYPE BUILD***, 2000 MB memory available,
Serial #69312178.
Ethernet address 0:14:4f:21:9e:b2, Host ID: 84219eb2.

Boot device: /pci@7c0/pci@0/pci@8/scsi@2/disk@0,0:a File and args:
zfs-file-system
Loading: /platform/SUNW,Sun-Fire-T1000/boot_archive
ramdisk-root hsfs-file-system
Loading: /platform/SUNW,Sun-Fire-T1000/kernel/sparcv9/unix
SunOS Release 5.11 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
os-io Hostname: t1000

t1000 console login:
```

SPARC ベースシステムの対話式でのブート

対話式でのシステムのブートは、ブートプロセス中に代替カーネルまたは `/etc/system` ファイルを指定する必要がある場合に役立ちます。システムを対話式でブートするには、次の手順を使用します。

▼ システムを対話式でブートする方法

ブート環境が1つしかない SPARC ベースシステムをブートするときに、代替の `/etc/system` ファイルを指定する場合は、`boot-a` コマンドを使用して、システムを対話式でブートできます。または、代替ブート環境を作成してブートすることにより、`/etc/system` ファイルの問題を解決できます。[68 ページの「SPARC プラットフォームの ZFS ブート環境からのブート」](#)を参照してください。

- 1 `/etc/system` ファイルと `boot/solaris/filelist.ramdisk` ファイルのバックアップコピーを作成します。例:

```
# cp /etc/system /etc/system.bak
# cp /boot/solaris/filelist.ramdisk /boot/solaris/filelist.ramdisk.orig
```

- 2 `etc/system.bak` ファイル名を `/boot/solaris/filelist.ramdisk` ファイルに追加します。

```
# echo "etc/system.bak" >> /boot/solaris/filelist.ramdisk
```

- 3 ブートアーカイブを更新します。

```
# bootadm update-archive -v
```

- 4 システムを **ok PROM** プロンプトにします。

- 5 システムを対話式でブートします。

```
ok boot -a
```

- 6 次のように、システムプロンプトに応答します。

- a. 代替のシステムファイルを指定して、**Return** キーを押します。例:

```
Name of system file [etc/system]: /etc/system.bak
```

- b. ルートファイルシステムを指定して、**Return** キーを押します。

- c. プロンプトが表示されたら、ルートデバイスの物理名を指定して、**Return** キーを押します。

情報を指定せずに **Return** キーを押すと、システムのデフォルトが受け入れられます。

- 7 情報を求めるプロンプトがシステムで表示されない場合は、`boot -a` コマンドを正しく入力しているかどうか確認してください。

例 2-4 システムを対話式でブートする

次の例では、利用できるデフォルトの選択例 ([] で囲まれた部分) を示します。`boot -a` コマンドを使用して代替ファイルシステムをブートする方法と例については、[30 ページの「システムを対話式でブートする方法」](#)を参照してください。

```
ok boot -a
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and args: -a
Name of system file [/etc/system]:
SunOS Release 5.11 Version ... 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Retire store [/etc/devices/retire_store] (/dev/null to bypass):
root filesystem type [zfs]:
Enter physical name of root device
[/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a]:
Hostname: system1
Mar 11 17:15:20 svc.startd[9]: svc:/system/filesystem/local:default: \
  Method "/lib/svc/method/fs-local" failed with exit status 95.
system1 console login: NIS domain name is solaris.us.oracle.com
NIS domain name is solaris.us.oracle.com

system1 console login:
```


システムのシャットダウン(手順)

この章では、システムのシャットダウンについての概要とタスク関連の情報を提供します。SPARC ベースシステムのシャットダウン手順は、x86 ベースシステムのシャットダウンの手順と同じです。ただし、出力例が異なる場合があります。

この章の内容は次のとおりです。

- 33 ページの「システムのシャットダウン(タスクマップ)」
- 34 ページの「システムのシャットダウンの概要」
- 34 ページの「システムのシャットダウンに関するガイドライン」
- 36 ページの「システムのシャットダウン」
- 40 ページの「システムデバイスの電源の切断」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン(概要)」を参照してください。

x86 ベースシステムのブートおよびシャットダウンについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』を参照してください。

システムのシャットダウン(タスクマップ)

表 3-1 システムのシャットダウン:タスクマップ

タスク	説明	参照先
システムにログインしているユーザーを確認します。	システムが複数ユーザーによって使用されているサーバーの場合は、 <code>who</code> コマンドを使用して、システムにログインしているユーザーを確認します。	36 ページの「システムにログインしているユーザーを確認する方法」

表 3-1 システムのシャットダウン:タスクマップ (続き)

タスク	説明	参照先
shutdown コマンドを使用してシステムをシャットダウンします。	適切なオプションで shutdown コマンドを使用してシステムをシャットダウンします。サーバーのシャットダウンには、この方法が推奨されます。	37 ページの「shutdown コマンドを使用してシステムをシャットダウンする方法」
init コマンドを使用してシステムをシャットダウンします。	init コマンドを使用し、適切な実行レベルを指定して、システムをシャットダウンします。	39 ページの「init コマンドを使用してシステムをシャットダウンする方法」

システムのシャットダウンの概要

Oracle Solaris は、電子メールやネットワークソフトウェアをいつでも利用できるように、停止することなく動作するように設計されています。しかし、システム管理タスクを行う場合や緊急事態が発生した場合は、システムをシャットダウンして安全に電源を切断できる状態にする必要があります。場合によっては、システムを一部のシステムサービスしか利用できない中間の実行レベルまで移行する必要があります。

次のような場合がそれに該当します。

- ハードウェアを追加または削除する
- 予定された停電に備える
- ファイルシステムの保守を行う (バックアップなど)

システムの電源管理機能を使用する方法については、[poweradm\(1M\)](#) のマニュアルページを参照してください。

システムのシャットダウンに関するガイドライン

システムをシャットダウンするときは、次の点に注意してください。

- システムをシャットダウンするには、shutdown または init コマンドを使用してください。どちらのコマンドも、すべてのシステムプロセスとサービスを正常に終了させてから、システムを正常にシャットダウンします。
- shutdown および init コマンドを使用するには、root の役割である必要があります。
- shutdown および init コマンドはどちらも実行レベルを引数に指定します。

もっともよく使用される実行レベルは次の3つです。

- **実行レベル 3** – すべてのシステムリソースを使用でき、ユーザーもログインできます。デフォルトでは、システムをブートすると実行レベル 3 になります。通常の運用で使用されます。この実行レベルは、NFS リソースを共有するマルチユーザー状態とも呼ばれます。
- **実行レベル 6** – システムをシャットダウンして実行レベル 0 にしたあと、SMB または NFS リソースを共有するマルチユーザーレベル (または `inittab` ファイルでデフォルトに指定された任意の実行レベル) にシステムをリブートします。
- **実行レベル 0** – オペレーティングシステムがシャットダウンされ、安全に電源が切断されます。システムの設置場所を変更したり、ハードウェアを追加または削除したりする場合は、システムを実行レベル 0 にする必要があります。

実行レベルについては、[18 ページの「実行レベルの動作」](#)で詳細に説明されています。

システムシャットダウンコマンド

`shutdown` および `init` コマンドが、システムのシャットダウンに使用される主なコマンドです。どちらのコマンドも、システムを正常にシャットダウンします。そのため、ファイルシステムに対するすべての変更がディスクに書き込まれ、さらにすべてのシステムサービス、プロセス、およびオペレーティングシステムが正常に終了します。

システムをオフにしてオンにすることは、正常な状態のシャットダウンではありません。これは、システムサービスが突然終了するためです。しかし、緊急時には、これらの方法が必要となる場合もあります。

次の表に、各種シャットダウンコマンドとその用途を要約します。

表 3-2 シャットダウンコマンド

コマンド	説明	どのような場合に使用するか
<code>shutdown</code>	<code>init</code> プログラムを呼び出してシステムをシャットダウンする実行可能ファイル。デフォルトでは、システムは実行レベル S に移行します。	このコマンドは、実行レベル 3 で動作しているサーバーをシャットダウンするために使用します。
<code>init</code>	すべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能ファイル。	このコマンドは高速なシステムシャットダウンを提供します。このコマンドは、ほかのユーザーが影響を受けないスタンドアロンシステムのシャットダウンに適しています。

表 3-2 シャットダウンコマンド (続き)

コマンド	説明	どのような場合に使用するか
reboot	ディスクを同期させ、ブート命令を <code>uadmin</code> システムコールに渡す実行可能ファイル。このシステムコールによってプロセッサが停止します。	<code>init</code> コマンドを使用するのが望ましい方法です。
halt、poweroff	ディスクを同期させ、プロセッサを停止する実行可能ファイル。	すべてのプロセスがシャットダウンされるわけではなく、また残りのファイルシステムのマウントも解除されないため、お勧めしません。正常なシャットダウンを行わずにサービスを停止する操作は、緊急時またはほとんどのサービスがすでに停止している場合に限って行うべきです。

システムのシャットダウン

次の手順と例は、`shutdown` および `init` コマンドを使用してシステムをシャットダウンする方法について説明したものです。

▼ システムにログインしているユーザーを確認する方法

マルチユーザーのタイムシェアリングシステムとして使用されている Oracle Solaris システムでは、システムをシャットダウンする前に、システムにログインしているユーザーがいるかどうかの確認が必要になることがあります。このような場合は次の手順を使用します。

- システムにログインしているユーザーを確認するには、次のように `who` コマンドを使用します。

```
$ who
holly      console      May  7 07:30
kryten     pts/0         May  7 07:35  (starlite)
lister     pts/1         May  7 07:40  (bluemidget)
```

- 1 列目のデータはログインしているユーザーのユーザー名を示します。
- 2 列目のデータはログインしているユーザーの端末回線を示します。
- 3 列目のデータはユーザーがログインした日時を示します。
- 4 列目のデータ (存在する場合) は、ユーザーがリモートシステムからログインしているときのホスト名を示します。

▼ shutdown コマンドを使用してシステムをシャットダウンする方法

- 1 **root** の役割になります。
- 2 マルチユーザーサーバーのシャットダウンの場合は、システムにログインしているユーザーがいるかどうかを確認します。

```
# who
```

注- この手順は条件付きで、システムがマルチユーザータイムシェアリングシステムである場合のみ必要で、新しい Oracle Solaris サーバーおよびプロセッサをシャットダウンするときには通常使用されません。

- 3 システムをシャットダウンします。

```
# shutdown -i init-state -g grace-period -y
```

-i init-state システムをデフォルトの S とは異なる init 状態にします。0、1、2、5、6 のいずれかを指定できます。

実行レベル 0 および 5 は、システムのシャットダウン用に予約された状態です。実行レベル 6 はシステムをリブートします。実行レベル 2 はマルチユーザーオペレーティング状態として使用できます。

-g grace-period システムがシャットダウンするまでの時間 (秒) を示します。デフォルト値は 60 秒です。

-y ユーザーの介入なしにシャットダウンを継続します。それ以外の場合、シャットダウンを継続するかどうか 60 秒後に尋ねられます。

詳細は、[shutdown\(1M\)](#) のマニュアルページを参照してください。

- 4 シャットダウンを継続するかどうか尋ねられたら、**y** を入力します。

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

-y オプションを指定した場合、このプロンプトは表示されません。

- 5 プロンプトが表示されたら、**root** パスワードを入力します。

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

- 6 システム管理タスクの実行が完了したら、**Ctrl+D** を押して、デフォルトのシステム実行レベルに戻ります。
- 7 次の表を使用して、システムが **shutdown** コマンドで指定した実行レベルに移行したことを確認します。

指定した実行レベル	SPARC システムのプロンプト
S (シングルユーザー状態)	#
0 (電源切断可能)	ok または>
実行レベル 3 (リモートリソースを共有するマルチユーザー状態)	hostname console login:

例 3-1 shutdown コマンドを使用してマルチユーザーサーバーをシングルユーザー状態 (実行レベル S) にする

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

```
# who
root  console      Jun 14 15:49      (:0)

# shutdown -g180 -y

Shutdown started.      Mon Jun 14 15:46:16...

Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 3 minutes .
.
.
Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 30 seconds .
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
.
.
Jun 14 15:49:00 venus syslogd: going down on signal 15
Killing user processes: done.

Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
.
.
.
```

例 3-2 shutdown コマンドを使用してシステムをシャットダウン状態 (実行レベル 0) にする

次の例では、shutdown コマンドを使用して、追加の確認を求めずに SPARC ベースシステムを 5 分後に実行レベル 0 にしています。

```
# shutdown
Shutdown started.    Thu Jun 17 12:40:25...

Broadcast Message from root (console) on pretend Thu Jun 17 12:40:25...
The system pretend will be shut down in 5 minutes
.
.
.
Changing to init state 0 - please wait
#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type  help  for more information
ok
```

参照 システムをシャットダウンした理由にかかわらず、すべてのファイルリソースが利用可能かつユーザーがログイン可能な、実行レベル3にシステムが戻ることを想定しているでしょう。システムをマルチユーザー状態に戻す手順については、[27 ページの「マルチユーザー状態 \(実行レベル 3\) への SPARC ベースシステムのブート」](#)を参照してください。

▼ **init** コマンドを使用してシステムをシャットダウンする方法

スタンドアロンシステムをシャットダウンする必要がある場合は、次の手順を実行します。

- 1 **root** の役割になります。
- 2 システムをシャットダウンします。

```
# init 5
```

詳細は、[init\(1M\)](#) のマニュアルページを参照してください。

例 3-3 **init** コマンドを使用してシステムをシャットダウン状態 (実行レベル 0) にする

この例では、**init** コマンドを使用して、システムの電源を安全に切断できるレベルにします。

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

参照 システムをシャットダウンした理由にかかわらず、すべてのファイルリソースが利用可能かつユーザーがログイン可能な、実行レベル3にシステムが戻ることを想定しているでしょう。システムをマルチユーザー状態に戻す手順については、[マルチユーザー状態 \(実行レベル3\) への SPARC ベースシステムのブート](#)を参照してください。

システムデバイスの電源の切断

次のようなときは、システムデバイスの電源を切断する必要がある場合があります。

- ハードウェアを置換または追加する。
- システムをある場所から別の場所に移動する。
- 予定された停電や自然災害 (接近中の雷雨など) に備える。

デバイスの電源の切断については、<http://www.oracle.com/technetwork/indexes/documentation/index.html>にある製品ドキュメント内の指定されたハードウェアに対する手順を参照してください。

SPARC ベースシステムのリブート (手順)

この章では、Oracle Solaris の高速リブート機能に関する情報を含め、SPARC ベースシステムをリブートするためのさまざまな方法について説明します。

この章の内容は次のとおりです。

- 41 ページの「SPARC ベースシステムのリブート (タスクマップ)」
- 42 ページの「SPARC ベースシステムのリブート」
- 44 ページの「SPARC ベースシステムのリブート処理の高速化」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン (概要)」を参照してください。

x86 ベースシステムのリブートについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 4 章「x86 ベースシステムのリブート (タスク)」を参照してください。

SPARC ベースシステムのリブート (タスクマップ)

表 4-1 SPARC ベースシステムのリブート: タスクマップ

タスク	説明	参照先
init コマンドを使用して SPARC ベースシステムをリブートします。	init コマンドを使用して実行レベルの移行を開始します。init コマンドを使用してシステムをリブートする場合は、マルチユーザーのシステム状態として実行レベル 2、3、および 4 を使用できません。	43 ページの「init コマンドを使用してシステムをリブートする方法」

表 4-1 SPARC ベースシステムのリブート:タスクマップ (続き)

タスク	説明	参照先
reboot コマンドを使用して SPARC ベースシステムをリブートします。	reboot コマンドを使用してカーネルを再起動し、システムをマルチユーザー状態にします。	44 ページの「reboot コマンドを使用してシステムをリブートする方法」
SPARC システムの高速リブートをブートします。	高速リブート機能が有効でない場合は、reboot コマンドに <code>-f</code> オプションを指定して使用して、SPARC ベースシステムの高速リブートを開始します。 高速リブート機能が有効化済みの場合は、reboot コマンドまたは <code>init 6</code> コマンドを使用すると、SPARC ベースシステムの高速リブートを自動的に開始できます。	44 ページの「SPARC システムの高速リブートを開始する方法」
高速リブートを SPARC ベースシステムでのデフォルト動作にします。	SPARC ベースシステムで高速リブート機能はサポートされていますが、デフォルトで無効化されています。boot-config サービスを構成すると、SPARC システムの高速リブートをデフォルトで実行できます。	45 ページの「高速リブート機能のデフォルト動作の変更」
高速リブートが有効化済みのシステムの標準リブートを開始します。	高速リブート機能が有効になっているシステムの標準リブートを実行するには、reboot コマンドを <code>-p</code> オプションとともに使用します。	45 ページの「高速リブートが有効化済みのシステムの標準リブートの開始」

SPARC ベースシステムのリブート

システムをリブートするには、`init` コマンドまたは `reboot` コマンドを使用できます。

システムは常に定義済み実行レベルの 1 つで動作します。実行レベルは `init` プロセスによって維持されるため、実行レベルは「init 状態」と呼ばれることもあります。`init` コマンドを使用すると、実行レベルの移行を開始できます。`init` コマンドを使用してシステムをリブートする場合は、マルチユーザーのシステム状態として実行レベル 2、3、および 4 を使用できます。

`reboot` コマンドはカーネルを再起動します。カーネルは PROM モニターによってメモリーに読み込まれ、読み込まれたカーネルに制御が渡されます。`root` ユーザーはいつでも `reboot` コマンドを使用できますが、サーバーのリブートなどの特定の状況

では、システムにログインしているすべてのユーザーにすぐにサービスが失われることを警告するために、通常は最初に `shutdown` コマンドが使用されます。詳細は、[第3章「システムのシャットダウン\(手順\)」](#)を参照してください。

▼ **init** コマンドを使用してシステムをリブートする方法

`init` コマンドは、システム上のすべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能シェルスクリプトです。

- 1 **root** の役割になります。
- 2 システムをリブートします。
 - `/etc/inittab` ファイルの **initdefault** エントリによって定義された状態にシステムをリブートするには、次のコマンドを入力します。


```
# init 6
```
 - システムをマルチユーザー状態にリブートするには、次のコマンドを入力します。


```
# init 2
```

例 4-1 `init` コマンドを使用してシステムをシングルユーザー状態(実行レベル S)にする

この例では、`init` コマンドを使用して、システムをシングルユーザー状態(実行レベル S)にしています。

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

SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
#
```

▼ **reboot** コマンドを使用してシステムをリブートする方法

- 1 **root** の役割になります。
- 2 システムをリブートします。

```
# reboot
```

SPARC ベースシステムのリブート処理の高速化

Oracle Solaris の高速リブート機能が SPARC プラットフォーム上でサポートされるようになりました。つまり、**reboot** コマンドの **-f** オプションを使用して一部の POST 検査を省略することでブートプロセスを高速化できます。

SPARC ベースシステムの高速リブート機能は、x86 ベースシステムとは異なる方法で動作します。x86 ベースシステムでは、高速リブートはデフォルトです。一方、SPARC ベースシステムでは、この動作は有効になっていますが、高速リブートを開始するには **reboot** コマンドの **-f** オプションを使用する必要があります。また、SPARC 上の高速リブートは特定の SPARC ベースシステムにのみ適用されます。sun4v システムでは、リブートは POST を伴わないハイパーバイザリブートであるため、高速リブートは不要です。

高速リブート機能は SMF を介して管理され、ブート構成サービス、`svc:/system/boot-config` によって実装されます。`boot-config` サービスは、デフォルトのブート構成プロパティを設定または変更する手段を提供します。`config/fastreboot_default` プロパティを `true` に設定すると、`reboot -f` コマンドを使用する必要なく、システムは高速リブートを自動的に実行します。SPARC プラットフォームでは、このプロパティの値はデフォルトで `false` に設定されています。

高速リブートを SPARC ベースシステムのデフォルト動作にするには、`svccfg` および `svcadm` コマンドを使用します。手順については、[45 ページの「高速リブート機能のデフォルト動作の変更」](#)を参照してください。

注-SPARC ベースシステムでは、`boot-config` サービスに、`action_authorization` および `value_authorization` として `solaris.system.shutdown` 権限も必要です。

▼ **SPARC** システムの高速リブートを開始する方法

`boot-config` サービスの `config/fastreboot_default` プロパティが `false` に設定されている場合 (デフォルトの動作) は、次の手順を使用して SPARC ベースシステムの

高速リブートを開始します。システムのリブート時に高速リブートが自動的に実行されるように、高速リブート機能のデフォルト動作を変更するには、[45 ページ](#)の「[高速リブート機能のデフォルト動作の変更](#)」を参照してください。

- 1 **root** の役割になります。
- 2 次のコマンドを入力してシステムの高速リブートを開始します。

```
# reboot -f
```

高速リブート機能のデフォルト動作の変更

boot-config サービスの config/fastreboot_default プロパティを使用すると、reboot コマンドまたは init 6 コマンドのいずれかが使用されている場合のシステムの自動高速リブートが有効になります。config/fastreboot_default プロパティを true に設定すると、システムは高速リブートを自動的に実行するため、reboot -f コマンドを使用する必要はありません。デフォルトでは、SPARC システム上でこのプロパティの値は false に設定されています。

例 4-2 SPARC:boot-config サービスのプロパティの構成

boot-config サービスの一部であるプロパティを構成するには、svccfg および svcadm コマンドを使用します。

SPARC ベースシステム上でプロパティの値を true に設定するには、次のコマンドを入力します。

```
# svccfg -s "system/boot-config:default" setprop config/fastreboot_default=true
# svcadm refresh svc:/system/boot-config:default
```

プロパティの値を true に設定すると、特定の POST 検査をバイパスすることでリブートプロセスが高速化されます。このプロパティを true に設定すると、reboot コマンドで -f オプションを使用して高速リブートをブートする必要はありません。

SMF を介したブート構成サービスの管理については、[svcadm\(1M\)](#) および [svccfg\(1M\)](#) のマニュアルページを参照してください。

高速リブートが有効化済みのシステムの標準リブートの開始

boot-config サービスのプロパティを再構成せずに、Oracle Solaris の高速リブート機能を有効化した SPARC ベースシステムをリブートするには、次のように reboot コマンドの -p オプションを使用します。

```
# reboot -p
```


SPARC ベースシステムのネットワークからのブート(手順)

この章では、SPARC ベースシステムのネットワークからのブートに関する概要情報、ガイドライン、および作業について説明します。

この章の内容は次のとおりです。

- 47 ページの「SPARC ベースシステムのネットワークからのブート(作業マップ)」
- 48 ページの「SPARC システムをネットワークからブートする」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン(概要)」を参照してください。

x86 ベースシステムのネットワークからのブートについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 5 章「x86 ベースシステムのネットワークからのブート(作業)」を参照してください。

SPARC ベースシステムのネットワークからのブート(作業マップ)

表 5-1 SPARC ベースシステムのネットワークからのブート:作業マップ

タスク	説明	参照先
OBP PROM にネットワークブート引数を追加して、広域ネットワーク (WAN) ブートを可能にします。	eeprom ユーティリティの network-boot-arguments パラメータを設定することによって、WAN ブートを実行するときに使用するネットワークブートプロトコルについての情報を保存します。	50 ページの「OpenBoot PROM でのネットワークブート引数の指定方法」

表 5-1 SPARC ベースシステムのネットワークからのブート:作業マップ (続き)

タスク	説明	参照先
NVRAM 別名を設定して、DHCP ネットワークプロトコルを使用することによって SPARC ベースシステムを自動的にブートします。	NVRAM 別名を設定することによって、システムリブート後に使用するネットワークブートプロトコルの情報を保存します。	51 ページの「 DHCP を使用して自動ブートするための NVRAM 別名の設定 」
SPARC システムをネットワークからブートします。	予備的な作業を実行したあと、boot コマンドを使用して、SPARC ベースシステムをネットワークからブートします。	51 ページの「 SPARC ベースシステムをネットワークからブートする方法 」

SPARC システムをネットワークからブートする

次の理由により、システムのネットワークからのブートが必要になることがあります。

- Oracle Solaris をインストールするため
- 復旧のため

Oracle Solaris で使用されるネットワーク構成ブート方法は動的ホスト構成プロトコル (DHCP) です。

この Oracle Solaris リリースでの DHCP の動作に関する一般的な情報と、DHCP サーバーを設定するための具体的な情報については、『[Oracle Solaris の管理: IP サービス](#)』のパート II 「[DHCP](#)」を参照してください。

SPARC ネットワークブートプロセス

ネットワークデバイスの場合、ローカルエリアネットワーク (LAN) 上でブートするプロセスと WAN 上でブートするプロセスは多少異なっています。どちらのネットワークブートシナリオでも、PROM によってブーター(この場合 `inetboot`)がブートサーバーまたはインストールサーバーからダウンロードされます。

LAN 上でブートするとき、ファームウェアは DHCP を使用して、ブートサーバーとインストールサーバーのいずれかを検出します。次に、Trivial File Transfer Protocol (TFTP) が使用されて、ブーター(この場合 `inetboot`)がダウンロードされます。

WAN 上でブートするときは、ファームウェアは DHCP または NVRAM プロパティを使用して、システムをネットワークからブートするために必要なインストールサーバー、ルーター、およびプロキシを検出します。ブーターのダウンロードに使用されるプロトコルは HTTP です。さらに、定義済みの非公開鍵を使ってブーターの署名が確認される場合もあります。

SPARC ベースシステムをネットワークからブートするための要件

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。スタンドアロンのシステムがローカルディスクからブートできない場合、そのシステムを復旧目的でネットワークからブートしなければならないことがあります。

- Oracle Solaris を復旧目的でインストールするために SPARC ベースシステムのネットワークブートを実行するには、DHCP サーバーが必要です。

クライアントのネットワークインタフェースの構成に必要な情報は、DHCP サーバーから提供されます。自動インストーラ (AI) サーバーを設定している場合は、そのサーバーも DHCP サーバーにすることができます。または、別の DHCP サーバーを設定できます。詳細については、『[Oracle Solaris の管理: IP サービス](#)』のパート II 「DHCP」を参照してください。

- また、tftp サービスを提供するブートサーバーも必要です。

OpenBoot PROM へのネットワークブート引数の設定

EEPROM ユーティリティの `network-boot-arguments` パラメータを使用すれば、WAN ブートを実行するときに PROM によって使用される構成パラメータを設定できます。PROM に設定したネットワークブート引数は、あらゆるデフォルト値よりも優先されます。DHCP を使用する場合、これらの引数も、所定のパラメータについて DHCP サーバーによって提供された構成情報よりも優先されます。

ネットワークからブートするように Oracle Solaris システムを手動で構成する場合、システムがブートするために必要なすべての情報をクライアントシステムに提供する必要があります。

PROM が必要な情報には次のものが含まれます。

- ブートクライアントの IP アドレス
- ブートファイルの名前
- ブートファイルイメージを提供するサーバーの IP アドレス

さらに、使用するデフォルトルーターのサブネットマスクおよび IP アドレスを提供することが必要な場合もあります。

ネットワークブートで使用する構文は次のとおりです。

`[protocol,] [key=value,]*`

`protocol` 使用するアドレス検索プロトコルを指定します。

key=value 構成パラメータを属性ペアとして指定します。

次の表に、`network-boot-arguments` パラメータに指定できる構成パラメータを示します。

パラメータ	説明
<code>tftp-server</code>	TFTP サーバーの IP アドレス。
ファイル	TFTP を使ってダウンロードするファイル、または WAN ブートの URL。
<code>host-ip</code>	クライアントの IP アドレス (ドット区切り 10 進表記)。
<code>router-ip</code>	デフォルトのルーターの IP アドレス (ドット区切り 10 進表記)。
<code>subnet-mask</code>	サブネットマスク (ドット区切り 10 進表記)
<code>client-id</code>	DHCP クライアント識別子
<code>hostname</code>	DHCP トランザクションに使用するホスト名
<code>http-proxy</code>	HTTP プロキシサーバー指定 (<i>IPADDR[:PORT]</i>)
<code>tftp-retries</code>	TFTP の最大リトライ回数
<code>dhcp-retries</code>	DHCP の最大リトライ回数

▼ **OpenBoot PROM** でのネットワークブート引数の指定方法

始める前に システムをネットワークからブートするために必要なすべての予備的作業を完了します。詳細は、[49 ページの「SPARC ベースシステムをネットワークからブートするための要件」](#)を参照してください。

- 1 ネットワークからブートするシステム上で、`root` の役割になります。
- 2 `network-boot-arguments` パラメータに適切な値を指定します。

`# eeprom network-boot-arguments="protocol,hostname=hostname"`
たとえば、ブートプロトコルとして DHCP を使用し、ホスト名 `mssystem.example.com` を使用するには、`network-boot-arguments` パラメータの値を次のように設定します。

`# eeprom network-boot-arguments="DHCP,hostname=mssystem.example.com"`

- 3 システムを `ok PROM` プロンプトにします。
`# init 0`

4 ネットワークからシステムをブートします。

```
ok boot net
```

注 - この方法で `network-boot-arguments` パラメータを指定した場合、PROM コマンド行で引数を指定する必要はありません。指定すると、指定済みの `network-boot-arguments` パラメータに設定されたほかの値は無視されます。

DHCP を使用して自動ブートするための NVRAM 別名の設定

Oracle Solaris 11 では、Oracle Solaris をインストールするためにネットワークからブートするとき、使用されるネットワーク構成ブート方法は DHCP です。DHCP を使用してシステムをネットワークからブートするには、DHCP ブートサーバーをネットワーク上で利用できる必要があります。

`boot` コマンドを実行するとき、DHCP プロトコルを使用した SPARC ベースシステムブートを指定できます。あるいは、NVRAM 別名を設定することによって、システムをリブートしてもこの情報を PROM レベルで保存できます。

次の `nvalias` コマンドの例では、デフォルトで DHCP を使ってブートするように、ネットワークデバイスの別名を設定します。

```
ok nvalias net /pci@1f,4000/network@1,1:dhcp
```

結果として、`boot net` と入力するだけで、システムは DHCP を使用してブートします。



注意 - `nvalias` コマンドと `nvunalias` コマンドの構文を十分理解するまでは、`nvalias` コマンドを使用して NVRAMRC ファイルを変更しないでください。

▼ SPARC ベースシステムをネットワークからブートする方法

- 始める前に
- DHCP 構成を設定するための前提条件となるタスクをすべて実行します。
49 ページの「[SPARC ベースシステムをネットワークからブートするための要件](#)」を参照してください。
 - Oracle Solaris をインストールするためにシステムをネットワーク経由でブートする場合は、最初に AI クライアントイメージをダウンロードし、そのイメージに基づいてインストールサービスを作成します。手順については、『[Oracle Solaris 11 システムのインストール](#)』のパート III「インストールサーバーを使用したインストール」を参照してください。

- 1 **root** の役割になります。
- 2 必要な場合、システムを **ok PROM** プロンプトにします。
init 0
- 3 「**install**」フラグを付けずにネットワークからシステムをブートします。
ok boot net:dhcp

注- デフォルトで DHCP を使用してブートするように PROM 設定を変更した場合、次に示すように **boot net** と指定するだけでかまいません。

ok boot net

SPARC ベースシステムでのブートパラメータの変更(手順)

この章では、SPARC ベースシステムのデフォルトのブート動作を変更するためのタスク関連の情報を提供します。

この章の内容は次のとおりです。

- 54 ページの「SPARC ベースシステムでのブートパラメータの変更(タスクマップ)」
- 55 ページの「SPARC ベースシステムでのブートパラメータの変更」

Oracle Integrated Lights Out Manager (ILOM) サービスプロセッサ上の SPARC ブートモードプロパティを構成する必要がある場合は、<http://download.oracle.com/docs/cd/E19166-01/E20792/z40003d6165586.html#scrolltoc> にあるハードウェアのドキュメントを参照してください。

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン(概要)」を参照してください。

x86 ベースシステムのブートパラメータの変更については、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 6 章「x86 ベースシステム上のブートパラメータの変更(タスク)」を参照してください。

SPARC ベースシステムでのブートパラメータの変更(タスクマップ)

表 6-1 SPARC ベースシステムでのブートパラメータの変更: タスクマップ

タスク	説明	参照先
SPARC ベースシステムの PROM リビジョン番号を確認します。	システムの PROM リビジョン番号を表示するには、 <code>ok PROM</code> プロンプトで <code>banner</code> コマンドを使用します。	55 ページの「システムの PROM リビジョン番号を確認する方法」
ブート可能な SPARC ベースシステム上のデバイスを確認します。	ブート PROM を使用してブート動作を変更する前に、システム上のデバイスを確認します。	55 ページの「システム上のデバイスを確認する方法」
SPARC ベースシステムの現在のブートデバイスを表示します。	この手順を使用して、システムのブート元となる現在のデフォルトのブートデバイスを調べます。	57 ページの「デフォルトのブートデバイスを調べる方法」
SPARC ベースシステムのデフォルトのブートデバイスを変更します。	デフォルトのブートデバイスを変更するには、次のいずれかの方法を使用します。 ■ <code>ok PROM</code> プロンプトで <code>boot-device</code> パラメータを変更する。 ■ <code>eeeprom</code> コマンドを使用して <code>boot-device</code> パラメータを変更する。	58 ページの「ブート PROM を使用してデフォルトのブートデバイスを変更する方法」 59 ページの「 <code>eeeprom</code> ユーティリティを使用してデフォルトのブートデバイスを変更する方法」
SPARC ベースシステムのデフォルトのブートファイルまたはカーネルを変更します。	システムがブートするデフォルトのカーネルを変更するには、次のいずれかの方法を使用します。 ■ ブート PROM を使用して <code>boot-file</code> パラメータを変更する。 ■ <code>eeeprom</code> コマンドを使用して <code>boot-file</code> パラメータを変更する。	60 ページの「ブート PROM を使用してデフォルトのブートファイルを変更する方法」 60 ページの「 <code>eeeprom</code> ユーティリティを使用してデフォルトのブートファイルを変更する方法」

SPARC ベースシステムでのブートパラメータの変更

ブート PROM は、SPARC ベースシステムをブートし、ブートパラメータを変更するために使用されます。たとえば、ブート元のデバイスをリセットしたり、デフォルトのブートファイルまたはカーネルを変更したり、ハードウェア診断を実行してからシステムをマルチユーザー状態にしたりすることが必要な場合もあります。

次のいずれかのタスクを行う必要がある場合は、デフォルトのブートデバイスを変更する必要があります。

- 新しいドライブを永久または一時的にシステムに追加します
- ネットワークブート方法を変更します
- スタンドアロンシステムを一時的にネットワークからブートします

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

▼ システムの **PROM** リビジョン番号を確認する方法

- 1 システムを **ok PROM** プロンプトにします。

詳細については、[39 ページ](#)の「[init コマンドを使用してシステムをシャットダウンする方法](#)」を参照してください。

- 2 システムの **PROM** リビジョン番号を表示するには、**banner** コマンドを使用します。

```
ok banner
```

▼ システム上のデバイスを確認する方法

システム上のデバイスを確認してブート元となる適切なデバイスを調べる必要があります。

始める前に probe コマンドを使用してシステムに接続されているデバイスを安全に確認するためには、次のことを行なっておく必要があります。

- PROM の auto-boot? パラメータを false に変更します。

```
ok setenv auto-boot? false
```

- reset-all コマンドを発行して、システムのレジスタをクリアします。

```
ok reset-all
```

sifting probe コマンドを使用すると、システム上で利用可能な probe コマンドを表示できます。

```
ok sifting probe
```

システムのレジスタをクリアしないで `probe` コマンドを実行すると、次のメッセージが表示されます。

```
ok probe-scsi
This command may hang the system if a Stop-A or halt command
has been executed. Please type reset-all to reset the system
before executing this command.
Do you wish to continue? (y/n) n
```

- 1 システム上のデバイスを確認します。

```
ok probe-device
```

- 2 (省略可能) 電源障害後や `reset` コマンドの使用後にシステムをリブートするようするには、`auto-boot?` パラメータを `true` にリセットします。

```
ok setenv auto-boot? true
auto-boot? = true
```

- 3 システムをマルチユーザー状態にブートします。

```
ok reset-all
```

例 6-1 システム上のデバイスを確認する

次の例は、システムに接続されているデバイスの確認方法を示しています。

```
ok setenv auto-boot? false
auto-boot? = false
ok reset-all
SC Alert: Host System has Reset
```

```
Sun Fire T200, No Keyboard
```

```
.
.
.
```

```
OpenBoot 4.30.4.a, 16256 MB memory available, Serial #69069018.
Ethernet address 0:14:4f:1d:e8:da, Host ID: 841de8da.
```

```
ok probe-ide
Device 0 ( Primary Master )
Removable ATAPI Model: MATSHITACD-RW CW-8124
```

```
Device 1 ( Primary Slave )
Not Present
```

```
Device 2 ( Secondary Master )
Not Present
```

```
Device 3 ( Secondary Slave )
Not Present
```

```
ok setenv auto-boot? true
auto-boot? = true
```

別の方法として、`devalias` コマンドを使うと、システムに接続されている「可能性のある」デバイスの別名と関連パスを確認できます。例:

```
ok devalias
ttya                /pci@7c0/pci@0/pci@1/pci@0/isa@2/serial@0,3f8
nvram               /virtual-devices/nvram@3
net3                /pci@7c0/pci@0/pci@2/network@0,1
net2                /pci@7c0/pci@0/pci@2/network@0
net1                /pci@780/pci@0/pci@1/network@0,1
net0                /pci@780/pci@0/pci@1/network@0
net                 /pci@780/pci@0/pci@1/network@0
ide                 /pci@7c0/pci@0/pci@1/pci@0/ide@8
cdrom               /pci@7c0/pci@0/pci@1/pci@0/ide@8/cdrom@0,0:f
disk3               /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@3
disk2               /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@2
disk1               /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1
disk0               /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
disk                /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
scsi                /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
virtual-console     /virtual-devices/console@1
name                aliases
```

▼ デフォルトのブートデバイスを調べる方法

- 1 システムを **ok PROM** プロンプトにします。

詳細については、[39 ページの「init コマンドを使用してシステムをシャットダウンする方法」](#)を参照してください。

- 2 デフォルトのブートデバイスを確認します。

```
ok printenv boot-device
```

`boot-device` ブートするデバイスを設定するパラメータを示します。

詳細は、[printenv\(1B\)](#) のマニュアルページを参照してください。

次のような形式で、デフォルトの `boot-device` が表示されます。

```
boot-device = /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a
```

`boot-device` パラメータがネットワークブートデバイスを指定する場合、出力は次のようになります。

```
boot-device = /sbus@1f,0/SUNW,fas@e,88000000/sd@a,0:a \
/sbus@1f,0/SUNW,fas@e,88000000/sd@0,0:a disk net
```

▼ ブート **PROM** を使用してデフォルトのブートデバイスを変更する方法

始める前に デフォルトのブートデバイスをほかのデバイスに変更するには、まずシステム上のデバイスを確認しておく必要があります。システム上のデバイスを確認する方法については、[55 ページの「システム上のデバイスを確認する方法」](#)を参照してください。

- 1 システムを **ok PROM** プロンプトにします。

```
# init 0
```

- 2 **boot-device** パラメータの値を変更します。

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

`device[n]` **boot-device** の値 (disk または network など) を設定します。*n* にはディスク番号を指定できます。ディスク番号を確認する必要がある場合は、`probe` コマンドのいずれかを使用します。

- 3 デフォルトのブートデバイスが変更されていることを確認します。

```
ok printenv boot-device
```

- 4 新しい **boot-device** 値を保存します。

```
ok reset-all
```

新しい **boot-device** 値が PROM に書き込まれます。

例 6-2 ブート PROM を使用してデフォルトのブートデバイスを変更する

この例では、デフォルトのブートデバイスをディスクに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device /pci@1f,4000/scsi@3/disk@1,0
boot-device = /pci@1f,4000/scsi@3/disk@1,0
ok printenv boot-device
boot-device /pci@1f,4000/scsi@3/disk@1,0
ok boot
Resetting ...

screen not found.
Can't open input device.
```

```
Keyboard not present. Using ttys for input and output.
.
.
.
Rebooting with command: boot disk1
Boot device: /pci@1f,4000/scsi@3/disk@1,0 File and args:
```

この例では、デフォルトのブートデバイスをネットワークに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device = net
ok printenv boot-device
boot-device net disk
ok reset
.
.
.
Boot device: net File and args:

pluto console login:
```

▼ **eeprom** ユーティリティーを使用してデフォルトのブートデバイスを変更する方法

- 1 **root** の役割になります。
- 2 ブートする代替デバイスを指定します。
- 3 新しいブートパラメータが設定されていることを確認します。

```
# eeprom boot-device
```

この出力には、**boot-device** パラメータの新しい **eeprom** 値が表示されるはずです。

▼ ブート **PROM** を使用してデフォルトのブートファイルを変更する方法

- 1 システムを実行レベル **0** にします。

```
# init 0
```

ok PROM プロンプトが表示されます。詳細は、[init\(1M\)](#) のマニュアルページを参照してください。

- 2 **boot-file** プロパティを代替のブートファイルまたはカーネルに設定します。

```
ok setenv boot-file boot-file
```

- 3 デフォルトのブートファイルまたはカーネルが変更されていることを確認します。

```
ok printenv boot-file
```

- 4 新しい **boot-file** 値を保存します。

```
ok reset-all
```

新しい **boot-file** 値が PROM に書き込まれます。

▼ **eeprom** ユーティリティーを使用してデフォルトのブートファイルを変更する方法

- 1 **root** の役割になります。

- 2 ブートする代替ブートファイルまたはカーネルを指定します。

```
# eeprom boot-file new boot-file
```

例:

```
# eeprom boot-file=kernel.name/sparcv9/unix
```

- 3 デフォルトのブートファイルが変更されていることを確認します。

```
# eeprom boot-file
```

この出力には、指定したパラメータの新しい **eeprom** 値が表示されるはずです。

SPARC プラットフォームでの ZFS ブート環境の作成と管理、およびこのブート環境からのブート(手順)

この章では、SPARC ベースシステム上で ZFS ブート環境 (*BE* と呼ばれる) の作成と管理、およびこのブート環境からのブートを行う方法について説明します。

この章の内容は次のとおりです。

- 62 ページの「ZFS ブート環境の作成と管理、およびこのブート環境からのブート (タスクマップ)」
- 63 ページの「ブート環境の作成と管理」
- 68 ページの「SPARC プラットフォームの ZFS ブート環境からのブート」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン(概要)」を参照してください。

x86 ベースシステムでの ZFS ブート環境からのブートについては、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 7 章「x86 プラットフォーム上での ZFS ブート環境の作成、管理、およびこのブート環境からのブート (タスク)」を参照してください。

ブート環境の管理については、『Oracle Solaris 11 ブート環境の作成と管理』を参照してください。

ZFS ブート環境の作成と管理、およびこのブート環境からのブート(タスクマップ)

表 7-1 ZFS ブート環境の作成と管理、およびこのブート環境からのブート: タスクマップ

タスク	説明	参照先
ブート環境、スナップショット、およびデータセットの一覧を表示します。	ブート環境、スナップショット、およびデータセットの一覧を表示するには、 <code>beadm list</code> コマンドを使用します。	66 ページの「使用可能なブート環境、スナップショット、およびデータセットの一覧を表示する方法」
新しいブート環境を作成します。	<code>beadm create</code> コマンドを使用して新しいブート環境を作成します。	63 ページの「新しいブート環境を作成する方法」
ブート環境のスナップショットを作成します。	<code>beadm create beName@snapshot</code> コマンドを使用して既存のブート環境のスナップショットを作成します。	65 ページの「ブート環境のスナップショットの作成方法」
既存のスナップショットからブート環境を作成します。	<code>beadm</code> コマンドを使用して既存のスナップショットから新しいブート環境を作成します。	65 ページの「既存のスナップショットからブート環境を作成する方法」
新しく作成されたブート環境をアクティブにします。	<code>beadm activate</code> コマンドを使用して、新しく作成されたブート環境をアクティブにします。	65 ページの「新しく作成されたブート環境をアクティブにする方法」
ブートシーケンス中に SPARC ベースシステム上のブート環境とデータセットの一覧を表示します。	システム上のブート環境の一覧をブートシーケンス中に表示するには、 <code>boot</code> コマンドで <code>-L</code> オプションを指定します。	69 ページの「SPARC: ブートシーケンス中に使用可能なブート環境の一覧を表示する方法」
ブート環境を破棄します。	<code>beadm destroy</code> コマンドを使用してブート環境を破棄します。	67 ページの「ブート環境を破棄する方法」
SPARC ベースシステム上の指定されたブート環境、データセット、またはルートファイルシステムからブートします。	<code>boot -z</code> オプションを使用して、指定された ZFS ブート環境、スナップショット、またはデータセットをブートします。 注-このオプションは、ZFS プールを含むブートデバイスでのみサポートされています。	70 ページの「ZFS ブート環境またはルートファイルシステムからブートする方法」

ブート環境の作成と管理

次のタスクでは、`beadm` ユーティリティを使用して、ブート環境、スナップショット、およびデータセットを作成および管理する方法について説明します。

- ブート環境 (BE) とは、ブート用に指定された ZFS ファイルシステムのことで、ブート環境は基本的に、Oracle Solaris OS イメージのブート可能インスタンスに、そのイメージにインストールされたその他のソフトウェアパッケージを加えたものです。1つのシステム上に複数のブート環境を維持できます。各ブート環境には異なる OS バージョンをインストールできます。Oracle Solaris をインストールすると、インストール中に新しいブート環境が自動的に作成されます。
- スナップショットとは、特定の時点で取得されるデータセットまたはブート環境の読み取り専用イメージです。スナップショットはブート可能でないことに注意してください。ただし、特定のスナップショットに基づくブート環境を作成したあと、その新しいブート環境をアクティブにすることで、そのブート環境を次のシステムリブート時のデフォルトブート環境とすることができます。
- データセットは、ZFS ファイルシステム、クローン、スナップショット、またはボリュームを識別するために使用される一般的な用語です。
- 共有データセットとは、`/export` のように、アクティブなブート環境とアクティブでないブート環境の両方において同じマウントポイントを持つユーザー定義ディレクトリのことです。共有データセットは、各ブート環境のルートデータセット領域の外側にあります。
- ブート環境のクリティカルデータセットは、そのブート環境のルートデータセット領域に含まれています。

`beadm` コマンドの詳細は、[beadm\(1M\)](#) のマニュアルページを参照してください。ブート環境の管理に関する詳細は、『[Oracle Solaris 11 ブート環境の作成と管理](#)』を参照してください。大域または非大域ゾーン環境での `beadm` コマンドの使用についての詳細は、『[Oracle Solaris 11 ブート環境の作成と管理](#)』の第2章「[beadm でのゾーンのサポート](#)」を参照してください。

▼ 新しいブート環境を作成する方法

- 1 `root` の役割になります。
- 2 `beadm create` コマンドを使用してブート環境を作成します。

```
# beadm create beName
```

`beName` は、新しいブート環境の名前を表す変数です。この新しいブート環境はアクティブではありません。

注 - `beadm create` コマンドでは、部分的なブート環境は作成されません。新しい完全なブート環境が正常に作成されるか、コマンドが失敗するかのいずれかです。

3 (省略可能) 新しいブート環境をマウントします。

```
# beadm mount beName mountpoint
```

マウントポイントのディレクトリが存在しない場合、`beadm` コマンドはそのディレクトリを作成してから、そのディレクトリにブート環境をマウントします。ブート環境がすでにマウントされている場合、`beadm mount` コマンドは失敗し、ブート環境は新しい場所に再マウントされません。

ブート環境はマウントされますが、アクティブでない状態です。マウント済みのアクティブでないブート環境をアップグレードできます。また、システムをリブートする前にブート環境をマウント解除することを忘れないでください。

4 (省略可能) 新しいブート環境からブートするには、最初にブート環境をアクティブにします。

```
# beadm activate beName
```

`beName` は、アクティブにするブート環境の名前を表す変数です。リブート時に、新しくアクティブになったブート環境は、GRUB メニューに表示されるデフォルトのブートエントリになります。

例 7-1 共有データセットを持つ複製されたブート環境を作成する

次の例は、BE2 という名前の新しく作成したブート環境内のデータセットを示しています。この例では、元のブート環境は BE1 です。新しいブート環境 BE2 には、BE1 から複製された個々のデータセットが含まれています。BE1 に `/opt` などの従来のファイルシステム用の個々のデータセットが含まれている場合は、それらのデータセットも複製されます。

```
# beadm create BE2
# beadm list -a BE2
BE/Dataset/Snapshot Active Mountpoint Space Policy Created
-----
BE2
  rpool/ROOT/BE2      -      -      42.0K static 2011-04-07 10:56
```

前の出力に示すように、ストレージプールの名前は `rpool` です。このプールは、初期インストールまたはアップグレードによってあらかじめ設定されているため、すでにシステム上に存在します。ROOT は、同様に初期インストールまたはアップグレードによってあらかじめ作成された特殊なデータセットです。ROOT は、ブート環境のルートで使用するために排他的に予約されています。

▼ ブート環境のスナップショットの作成方法

- 1 **root** の役割になります。
- 2 ブート環境のスナップショットを作成します。

```
# beadm create beName@snapshot
```

スナップショット名の例として、次のようなものがあります。

- BE@0312200.12:15pm
- BE2@backup
- BE1@march132008

▼ 既存のスナップショットからブート環境を作成する方法

- 1 **root** の役割になります。
- 2 次のコマンドを入力して、スナップショットから新しいブート環境を作成します。

```
# beadm create -e BENAME@snapshotdescription beName
```

BENAME@snapshotdescription を既存のスナップショットの名前に、*beName* を新しいブート環境のカスタム名にそれぞれ置き換えます。

例:

```
# beadm create -e BE1@now BE2
```

このコマンドは、BE1@now という名前の既存のスナップショットから BE2 という名前の新しいブート環境を作成します。このあと、このブート環境をアクティブにすることができます。手順については、[65 ページの「新しく作成されたブート環境をアクティブにする方法」](#)を参照してください。

▼ 新しく作成されたブート環境をアクティブにする方法

新しく作成されたブート環境をアクティブにすることによって、リブート時にそのブート環境がデフォルトのブート環境となりブートされます。アクティブにできるブート環境は常に1つだけであることに注意してください。

- 1 **root** の役割になります。

- 2 次のコマンドを使用して、新しく作成されたブート環境をアクティブにします。

```
# beadm activate beName
```

beName は、アクティブにするブート環境を表す変数です。

次の事項に注意してください。

- `beadm activate beName` コマンドは、`bootfs` ブート可能プールプロパティを、アクティブにするブート環境の `ROOT` データセットの値に設定することによってブート環境をアクティブにします。
- `beadm activate` コマンドは、新しくアクティブにしたブート環境を `menu.lst` ファイルにデフォルトとして設定します。

- 3 システムをリブートします。

これで新しくアクティブにしたブート環境が SPARC ブートメニューのデフォルトエントリになります。

▼ 使用可能なブート環境、スナップショット、およびデータセットの一覧を表示する方法

`beadm` コマンドを使用して作成された使用可能なブート環境、スナップショット、およびデータセットを表示するには、`beadm list` コマンドを使用します。

- 1 **root** の役割になります。
- 2 **beadm** コマンドを使用して作成されたシステム上の使用可能なデータセットをすべて表示するには、次のコマンドを入力します。

```
# beadm list option
```

- a ブート環境に関する利用可能なすべての情報を一覧表示します。このオプションには、従属データセットおよびスナップショットが含まれます。
- d ブート環境のデータセットに関する情報を一覧表示します。
- s ブート環境のスナップショットに関する情報を一覧表示します。このオプションは、-d オプションと組み合わせて使用されます。
- H 表示からヘッダー情報を省略します。このオプションを選択すると、スクリプトやほかのプログラムの表示をより簡単に解析できるようになります。

- 3 特定のブート環境について使用可能なデータセットを一覧表示するには、**beadm list** コマンド構文にブート環境名を含めます。

たとえば、**oracle-solaris** ブート環境内で使用可能なデータセットをすべて表示するには、次のようなコマンドを入力します。

```
# beadm list -a oracle-solaris
BE/Dataset/Snapshot    Active Mountpoint Space  Policy Created
-----
oracle-solaris
  rpool/ROOT/solaris -          14.33M static 2011-01-20 07:45
```

例7-2 スナップショットの仕様を表示する

次の **beadm list** の例では、**-s** オプションを指定します。これにより、現在のイメージに存在するすべてのスナップショットの情報が表示されます。**zfs** コマンドを使用することによって、これらのスナップショットの状況も表示できます。

次の結果の例では、各スナップショットのタイトルに、スナップショットが作成された日時を示すタイムスタンプが含まれています。

```
# beadm list -s test-2
```

結果の例が表示されます。

```
BE/Snapshot    Space Policy Created
-----
test-2
test-2@2010-04-12-22:29:27 264.02M static 2010-04-12 16:29
test-2@2010-06-02-20:28:51 32.50M static 2010-06-02 14:28
test-2@2010-06-03-16:51:01 16.66M static 2010-06-03 10:51
test-2@2010-07-13-22:01:56 25.93M static 2010-07-13 16:01
test-2@2010-07-21-17:15:15 26.00M static 2010-07-21 11:15
test-2@2010-07-25-19:07:03 13.75M static 2010-07-25 13:07
test-2@2010-07-25-20:33:41 12.32M static 2010-07-25 14:33
test-2@2010-07-25-20:41:23 30.60M static 2010-07-25 14:41
test-2@2010-08-06-15:53:15 8.92M static 2010-08-06 09:53
test-2@2010-08-06-16:00:37 8.92M static 2010-08-06 10:00
test-2@2010-08-09-16:06:11 193.72M static 2010-08-09 10:06
test-2@2010-08-09-20:28:59 102.69M static 2010-08-09 14:28
test-2@install 205.10M static 2010-03-16 19:04
```

▼ ブート環境を破棄する方法

システム上で使用可能なディスク容量を増やす場合は、**beadm** コマンドを使用すると既存のブート環境を破棄(削除)できます。

次の事項に注意してください。

- 現在ブートされているブート環境を破棄することはできません。
- `beadm destroy` コマンドを実行すると、破棄されたブート環境のエントリが SPARC ブートメニューから自動的に削除されます。
- `beadm destroy` コマンドで破棄されるのは、ブート環境のクリティカルデータセットまたは非共有データセットだけです。共有データセットは、ブート環境のルートデータセット領域の外側にあり、ブート環境が破棄される際に影響を受けません。

1 **root** の役割になります。

2 ブート環境を破棄するには、次のコマンドを入力します。

beadm destroy *beName*

ブート環境を破棄する前に、確認のプロンプトが表示されます。

`beadm destroy` *beName* で指定されたブート環境を破棄します。

-F 確認を要求せずに、ブート環境を強制的に破棄します。

-f ブート環境がマウントされている場合でも強制的に破棄します。

SPARC プラットフォームの ZFS ブート環境からのブート

`boot` コマンドの次の 2 つのオプションは、SPARC ベースシステムでの ZFS ルートファイルシステムからのブートをサポートします。

-L ZFS プール内の使用可能なブート環境の一覧を表示します。

注 - `boot -L` コマンドは、コマンド行からではなく OBP から実行されます。

-Z *dataset* 指定の ZFS ブート環境のルートファイルシステムをブートします。

ZFS ルートファイルシステムからシステムをブートする場合は、最初に OBP から -L オプションを指定して `boot` コマンドを実行し、システム上の使用可能なブート環境のリストを表示します。次に、-Z オプションを使用して指定のブート環境をブートします。

詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

▼ SPARC: ブートシーケンス中に使用可能なブート環境の一覧を表示する方法

SPARC ベースシステムでは、`menu.lst` ファイルに次の 2 つのコマンドが含まれています。

- `title` – ブート環境のタイトルを指定する
- `bootfs` – ブート環境のフルネームを指定する

次の手順で説明されているように、ZFS プール内のブート環境の一覧を表示するには、`boot -L` コマンドを使用します。このコマンドを実行すると、指定の ZFS ルートプール内の使用可能なブート環境の一覧が表示され、システムをブートするための方法も示されます。

- 1 **root** の役割になります。
- 2 システムを **ok PROM** プロンプトにします。
`# init 0`
- 3 ZFS プール内の使用可能なブート環境の一覧を表示します。
`ok boot device-specifier -L`
device-specifier は単一のルートファイルシステムではなくストレージプールを識別します。
- 4 表示されているいずれかのエントリをブートするには、そのエントリに対応する番号を入力します。
- 5 画面に表示される手順に従い、指定されたブート環境をブートします。
手順については、[70 ページの「ZFS ブート環境またはルートファイルシステムからブートする方法」](#)を参照してください。

例 7-3 boot-L コマンドを使用して使用可能なブート環境の一覧を表示する

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 94 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
Program terminated
ok boot -L
.
.
.
Boot device: /pci@1f,0/pci@1/scsi@8/disk@0,0 File and args: -L
zfs-file-system
Loading: /platformsun4v/bootlst
```

```
1.s10s_nbu6wos
2.zfs2BE
Select environment to boot: [ 1 - 2 ]: 2

to boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs2BE
```

参照 詳細は、『Oracle Solaris の管理: ZFS ファイルシステム』の第 5 章「ZFS ルートプールのコンポーネントの管理」を参照してください。

▼ ZFS ブート環境またはルートファイルシステムからブートする方法

ZFS からブートするときは、*device-specifier* によって単一のルートファイルシステムではなくストレージプールが識別されます。ストレージプールには、複数のブート環境、データセット、またはルートファイルシステムが含まれていることがあります。そのため、ZFS からブートするときは、ブートデバイスによってデフォルトと見なされるプール内のルートファイルシステムも識別する必要があります。デフォルトのブートデバイスはプールの `bootfs` プロパティによって識別されます。この手順では、ZFS ブート環境を指定してシステムをブートする方法を示します。使用可能なすべてのブートオプションの詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

注 - Oracle Solaris 11 では、ZFS ルートファイルシステムがデフォルトでブートされます。この手順を使用して、ブートする ZFS ルートファイルシステムを指定してください。

詳細は、[zpool\(1M\)](#) のマニュアルページを参照してください。

- 1 **root** の役割になります。
- 2 システムを **ok PROM** プロンプトにします。
`# init 0`
- 3 (省略可能) **boot** コマンドに **-L** オプションを使用して、使用可能なブート環境の一覧を表示します。
手順については、[69 ページの「SPARC: ブートシーケンス中に使用可能なブート環境の一覧を表示する方法」](#)を参照してください。

- 4 指定のエントリをブートするには、そのエントリの番号を入力し、**Return** キーを押します。

Select environment to boot: [1 - 2]:

- 5 システムをブートするには、画面に表示された指示に従います。

To boot the selected entry, invoke:

```
boot [<root-device>] -Z rpool/ROOT/boot-environment
```

```
ok boot -Z rpool/ROOT/boot-environment
```

例:

```
# boot -Z rpool/ROOT/zfs2BE
```

- 6 システムをブートしたあと、アクティブなブート環境を確認します。

```
# prtconf -vp | grep whoami
```

- 7 (省略可能)アクティブなブート環境のブートパスを表示するには、次のコマンドを入力します。

```
# prtconf -vp | grep bootpath
```

- 8 (省略可能)正しいブート環境がブートされたかどうかを確認するには、次のコマンドを入力します。

```
# df -lk
```

例 7-4 ZFS ブート環境からブートする

次の例は、boot -z コマンドを使用して SPARC システムで ZFS ブート環境をブートする方法を示しています。

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 79 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
Program terminated
ok boot -Z rpool/ROOT/zfs2BEe
Resetting
LOM event: =44d+21h38m12s host reset
g ...

rProcessor Speed = 648 MHz
Baud rate is 9600
8 Data bits, 1 stop bits, no parity (configured from lom)

.
.
.
Environment monitoring: disabled
```

```
Executng last command: boot -Z rpool/ROOT/zfs2BE
Boot device: /pci@1f,0/pci@1/scsi@8/disk@0,0 File and args: -Z rpool/ROOT/zfs2Be
zfs-file-system
.
.
.
Hostname: mallory
NIS domainname is ...
Reading ZFS config: done.
Mounting ZFS filesystems: (6/6)

mallory console login:
```

参照 ZFS ルートファイルシステムからのブートについての詳細は、[『Oracle Solaris の管理: ZFS ファイルシステム』](#)の「ZFS ルートファイルシステムからのブート」を参照してください。

SPARC ベースシステムのブート可能状態の維持 (手順)

この章では、ブート管理インタフェース (bootadm) を使用することによって、SPARC ベースシステムをブート可能に維持する方法について説明します。ブートアーカイブについての情報を表示するための手順、ブートアーカイブの整合性を維持するための手順、およびブートアーカイブの問題をトラブルシューティングする手順についても説明します。

この章の内容は次のとおりです。

- 73 ページの「SPARC ベースシステムのブート可能状態の維持 (タスクマップ)」
- 74 ページの「Oracle Solaris ブートアーカイブの説明」
- 76 ページの「ブートアーカイブ SMF サービスの管理」
- 77 ページの「ブートアーカイブの整合性の維持」

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン (概要)」を参照してください。

x86 ベースシステムのブート可能状態の維持については、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の第 8 章「x86 ベースシステムのブート可能状態の維持 (タスク)」を参照してください。

SPARC ベースシステムのブート可能状態の維持 (タスクマップ)

表 8-1 SPARC ベースシステムのブート可能状態の維持: タスクマップ

タスク	説明	参照先
bootadm コマンドを使用してブートアーカイブの内容を一覧表示します。	ブートアーカイブの内容を一覧表示する場合は、bootadm list-archive コマンドを使用します。	75 ページの「ブートアーカイブの内容を一覧表示する方法」

表 8-1 SPARC ベースシステムのブート可能状態の維持: タスクマップ (続き)

タスク	説明	参照先
boot-archive サービスを管理します。	boot-archive サービスは SMF に よって制御されます。svcadm コマ ンドを使用して、サービスを有効化ま たは無効化します。boot-archive サービスが実行されているかどうか を確認するには、svcs コマンドを使 用します。	76 ページの「ブートアーカイ ブ SMF サービスの管理」
bootadm コマンドを使用し てブートアーカイブを手動 で更新することに よって、ブートアーカイブ 更新障害を解決します。	SPARC ベースシステム上のブート アーカイブ更新障害を手動で解決す るには、この手順を使用します。	78 ページの「ブートアーカイ ブを手動で更新することに よってブートアーカイブの自 動更新障害を解決する方法」

Oracle Solaris ブートアーカイブの説明

Oracle Solaris をインストールするとき、bootadm コマンドによってシステム上にブートアーカイブが作成されます。ブートアーカイブはルートファイルシステムのサブセットです。このブートアーカイブには、すべてのカーネルモジュール、driver.conf ファイル、およびいくつかの構成ファイルが含まれています。これらのファイルは、/etc ディレクトリにあります。ブートアーカイブ内のファイルは、ルートファイルシステムがマウントされる前にカーネルによって読み取られます。ルートファイルシステムがマウントされると、ブートアーカイブはカーネルによってメモリーから破棄されます。次に、ファイル I/O がルートデバイスに実行されます。

また、bootadm コマンドは、ブートアーカイブの更新と検証を細部にわたって処理します。通常のシステムシャットダウンの処理中に、シャットダウンプロセスではブートアーカイブの内容がルートファイルシステムと比較されます。システムにドライバや構成ファイルなどの更新がある場合、ブートアーカイブがこれらの変更を含むように再構築されることによって、リブート時にブートアーカイブとルートファイルシステムの同期がとられます。

SPARC ブートアーカイブの場所と内容に関する情報の取得

SPARC ブートアーカイブ内のファイルは /platform ディレクトリにあります。ブートアーカイブの内容は、次の手順に記載されているように bootadm list-archive コマンドを使用すると一覧表示できます。ブートアーカイブ内のファイルが更新された場合、アーカイブを再構築する必要があります。変更を有効にするには、次にシステムがリブートする前にアーカイブを再構築する必要があります。

▼ ブートアーカイブの内容を一覧表示する方法

- 1 **root** の役割になります。
- 2 ブートアーカイブに含まれるファイルとディレクトリを一覧表示するには、次のように入力します。

```
# bootadm list-archive
```

例 8-1 SPARC ブートアーカイブの内容の一覧表示

次の例は、SPARC ベースシステムのブートアーカイブの内容を示しています。

```
root@tsystem:~# bootadm list-archive
platform/SUNW,A70/kernel
platform/SUNW,Netra-210/kernel
platform/SUNW,Netra-240/kernel
platform/SUNW,Netra-440/kernel
platform/SUNW,Netra-CP2300/kernel
platform/SUNW,Netra-CP3010/kernel
platform/SUNW,Netra-CP3060/kernel
platform/SUNW,Netra-CP3260/kernel
platform/SUNW,Netra-T12/kernel
platform/SUNW,Netra-T2000/kernel
platform/SUNW,Netra-T4/kernel
platform/SUNW,Netra-T5220/kernel
platform/SUNW,Netra-T5440/kernel
platform/SUNW,SPARC-Enterprise-T1000/kernel
platform/SUNW,SPARC-Enterprise-T2000/kernel
platform/SUNW,SPARC-Enterprise-T5120/kernel
platform/SUNW,SPARC-Enterprise-T5220/kernel
platform/SUNW,SPARC-Enterprise/kernel
platform/SUNW,Serverblade1/kernel
platform/SUNW,Sun-Blade-100/kernel
platform/SUNW,Sun-Blade-1000/kernel
platform/SUNW,Sun-Blade-1500/kernel
platform/SUNW,Sun-Blade-2500/kernel
platform/SUNW,Sun-Blade-T6300/kernel
platform/SUNW,Sun-Blade-T6320/kernel
platform/SUNW,Sun-Blade-T6340/kernel
platform/SUNW,Sun-Fire-15000/kernel
platform/SUNW,Sun-Fire-280R/kernel
platform/SUNW,Sun-Fire-480R/kernel
platform/SUNW,Sun-Fire-880/kernel
platform/SUNW,Sun-Fire-T1000/kernel
platform/SUNW,Sun-Fire-T200/kernel
platform/SUNW,Sun-Fire-V210/kernel
platform/SUNW,Sun-Fire-V215/kernel
platform/SUNW,Sun-Fire-V240/kernel
platform/SUNW,Sun-Fire-V245/kernel
platform/SUNW,Sun-Fire-V250/kernel
platform/SUNW,Sun-Fire-V440/kernel
platform/SUNW,Sun-Fire-V445/kernel
platform/SUNW,Sun-Fire-V490/kernel
platform/SUNW,Sun-Fire-V890/kernel
```

```
platform/SUNW,Sun-Fire/kernel
platform/SUNW,T5140/kernel
platform/SUNW,T5240/kernel
platform/SUNW,T5440/kernel
platform/SUNW,USBRDT-5240/kernel
platform/SUNW,Ultra-250/kernel
platform/SUNW,Ultra-4/kernel
platform/SUNW,Ultra-5_10/kernel
platform/SUNW,Ultra-80/kernel
platform/SUNW,Ultra-Enterprise-10000/kernel
platform/SUNW,Ultra-Enterprise/kernel
platform/SUNW,UltraAX-i2/kernel
platform/SUNW,UltraSPARC-IIe-NetraCT-40/kernel
platform/SUNW,UltraSPARC-IIe-NetraCT-60/kernel
platform/SUNW,UltraSPARC-IIi-Netract/kernel
platform/sun4u-us3/kernel
platform/sun4v/kernel
etc/cluster/nodeid
etc/dacf.conf
etc/driver
etc/mach
kernel
root@tsystem:~#
```

ブートアーカイブ SMF サービスの管理

boot-archive サービスは SMF によって制御されます。このサービスインスタンスは、`svc:/system/boot-archive:default` です。サービスを有効または無効にするには、`svcadm` コマンドを使用します。

boot-archive サービスが実行中かどうかの判定

boot-archive サービスが無効な場合は、システムリブート時にブートアーカイブの自動復旧が実行されないことがあります。その結果、ブートアーカイブが同期されないか破壊され、システムがブートされなくなる可能性があります。

boot-archive サービスが実行されているかどうかを確認するには、`svcs` コマンドを使用します。

```
$ svcs boot-archive
STATE          STIME          FMRI
online         Mar_31         svc:/system/boot-archive:default
```

この例で、`svcs` コマンドの出力は boot-archive サービスがオンラインであることを示しています。

詳細は、[svcadm\(1M\)](#) および [svcs\(1\)](#) のマニュアルページを参照してください。

▼ boot-archive SMF サービスを有効または無効にする方法

- 1 **root** の役割になります。

詳細は、『[Oracle Solaris の管理: セキュリティーサービス](#)』の「[管理権限を取得する方法](#)」を参照してください。

- 2 **boot-archive** サービスを有効または無効にするには、次のように入力します。

```
# svcadm enable | disable system/boot-archive
```

- 3 **boot-archive** サービスの状態を確認するには、次のように入力します。

```
# svcs boot-archive
```

サービスが実行されている場合は、オンラインサービスの状態が表示されます。

```
STATE          STIME      FMRI
online         9:02:38   svc:/system/boot-archive:default
```

サービスが実行されていない場合は、サービスがオフラインになっていることが出力に表示されます。

注意事項 ブートアーカイブ更新障害のトラブルシューティングの詳細については、[77 ページ](#)の「[ブートアーカイブの整合性の維持](#)」を参照してください。

ブートアーカイブの整合性の維持

ブート管理インタフェース **bootadm** によって、ブートアーカイブを維持するための次のタスクを実行できます。

- システムのブートアーカイブに含まれるファイルとディレクトリを一覧表示する。
- システム上の現在のブートアーカイブを手動で更新する。

このコマンドの構文は次のとおりです。

```
bootadm [subcommand] [-option] [-R altroot]
```

bootadm コマンドの詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。

▼ ブートアーカイブを手動で更新することによって ブートアーカイブの自動更新障害を解決する方法

システムのブートプロセス中に次のような警告メッセージが表示される場合は、適切な操作を実行します。

```
WARNING: Automatic update of the boot archive failed.  
Update the archives using 'bootadm update-archive'  
command and then reboot the system from the same device that  
was previously booted.
```

次の手順は、`bootadm` コマンドを使用して期限切れのブートアーカイブを手動で更新する方法を示しています。

注-ブートアーカイブを手動で更新する場合も同じ手順を使用できます。

- 1 **root** の役割になります。
- 2 ブートアーカイブを更新するには、次のコマンドを入力します。

```
# bootadm update-archive
```

注-代替ルートファイルシステムにあるブートアーカイブを更新するには、次のコマンドを入力します。

```
# bootadm update-archive -R /a
```

`-R altroot` `update-archive` サブコマンドに適用する代替ルートパスを指定します。



注意 - `-R` オプションを使用して非大域ゾーンのルートファイルシステムを参照してはいけません。そのようにすると、大域ゾーンのファイルシステムが損傷したり、大域ゾーンのセキュリティが低下したり、非大域ゾーンのファイルシステムが損傷したりする場合があります。[zones\(5\)](#) のマニュアルページを参照してください。

- 3 システムをリブートします。

```
# reboot
```

SPARC ベースシステムのブートのトラブルシューティング (手順)

ここでは、SPARC ベースシステム上での Oracle Solaris インスタンスのブートに関するトラブルシューティングの手順を示します。

この章の内容は次のとおりです。

- 80 ページの「SPARC ベースシステムのブートのトラブルシューティング (タスクマップ)」
- 81 ページの「復旧を目的とした SPARC ベースシステムのシャットダウンおよびブート」

復旧の目的での Oracle Solaris の停止と起動について (サービスプロセッサを実行している場合)、および Oracle ILOM サービスプロセッサを制御する手順については、<http://download.oracle.com/docs/cd/E19166-01/E20792/z400130a9112.html#scrolltoc> にあるハードウェアのドキュメントを参照してください。

SPARC ベースシステムのブートに関する概要情報は、第 1 章「SPARC ベースシステムのブートおよびシャットダウン (概要)」を参照してください。

Oracle Solaris ブートアーカイブの問題を解決する方法については、77 ページの「ブートアーカイブの整合性の維持」を参照してください。

SPARC ベースシステムのブートのトラブルシューティング(タスクマップ)

表 9-1 SPARC ベースシステムのブートのトラブルシューティング:タスクマップ

タスク	説明	参照先
復旧を目的として SPARC システムを停止します。	破損したファイルが原因で SPARC ベースシステムがブートされなくなっている場合は、最初にシステムを停止して復旧を試みます。	81 ページの「復旧を目的としてシステムを停止する方法」
SPARC ベースシステムをシングルユーザーモードでブートして、不正な root シェルまたは誤ったパスワードの入力などの軽微なブートの問題を解決します。	システムをシングルユーザーモードでブートして、不明な root パスワードまたは同様の問題を解決します。	83 ページの「シングルユーザーモードでブートして、不正な root シェルまたはパスワードの問題を解決する方法」
SPARC ベースシステムをメディアからブートして、不明な root パスワードの問題を解決します。	メディアからシステムをブートし、ルートプールをインポートおよびマウントして、問題を修正します。	83 ページの「メディアからブートして、不明な root パスワードを解決する方法」
どのサービスも起動しないで SPARC ベースシステムをブートします。	ブートプロセス中にシステムがハングアップする場合は、問題をトラブルシューティングするためにどのサービスも開始せずにシステムをブートします。	84 ページの「どのサービスも起動しないでシステムをブートする方法」
SPARC ベースシステムのクラッシュダンプとリブートを強制的に実行します。	トラブルシューティングの手段として、SPARC ベースシステムのクラッシュダンプとリブートを強制的に実行します。	86 ページの「クラッシュダンプを強制してシステムをリブートする方法」
カーネルデバッグ (kmdb) を有効にして、SPARC ベースシステムをブートします。	カーネルと対話してシステムの問題をトラブルシューティングするために、カーネルデバッグを有効にして SPARC ベースシステムをブートします。	87 ページの「カーネルデバッグ (kmdb) を有効にしてシステムをブートする方法」

復旧を目的とした **SPARC** ベースシステムのシャットダウンおよびブート

次の場合は、ブートおよびシステムのその他の問題を分析またはトラブルシューティングするために、システムをまずシャットダウンする必要があります。

- システムのブート時にエラーメッセージのトラブルシューティングを行います。
- システムを停止して復旧を試みます。
- 復旧を目的としてシステムをブートします。
- 強制的にクラッシュダンプを実行し、システムをリブートします。
- `kmdb` コマンドを使用してカーネルデバッガでシステムをブートします。

このあとの手順では、復旧の目的で SPARC ベースシステムを安全にシャットダウンしてからブートする方法について説明します。

システムの復旧目的の停止とブート

復旧の目的でシステムをブートする必要がある場合があります。一般的なエラーおよび復旧のシナリオには次のものが含まれます。

- シングルユーザーモードでシステムをブートして、`/etc/passwd` ファイルの `root` シェルエントリの修正や、NIS サーバーの変更などの軽微な問題を解決します。
- インストールメディアまたはネットワーク上のインストールサーバーからブートして、システムのブートを妨げている問題から回復したり、紛失した `root` パスワードを回復したりします。ルートプールのインポート、BE のマウント、および問題の修正を行うことにより、ブート構成の問題を解決します。

SPARC システムでは、`boot net:dhcp` コマンドは、Oracle Solaris 10 リリースで使用される `boot net` コマンドに代わるものです。

▼ 復旧を目的としてシステムを停止する方法

- 1 `shutdown` または `init 0` コマンドを使用して、システムを **ok PROM** プロンプトにします。
- 2 ファイルシステムを同期させます。
- 3 該当する `boot` コマンドを入力して、ブートプロセスを起動します。
詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。
- 4 システムが指定した実行レベルになっていることを確認します。

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

例 9-1 サービスプロセッサの電源をオフにする

Oracle ILOM サービスプロセッサ上で Oracle Solaris 11 を実行中の場合、オペレーティングシステムをシャットダウンしたあと、システムコンソールプロンプトからサービスプロセッサプロンプトに切り替える必要があります。その場所から、次の例に示すようにサービスプロセッサを停止できます。

```
# shutdown -g0 -i0 -y
# svc.startd: The system is coming down. Please wait.
svc.startd: 91 system services are now being stopped.
Jun 12 19:46:57 wgs41-58 syslogd: going down on signal 15
svc.stard: The system is down.
syncing file systems...done
Program terminated
r)ebboot o)k prompt, h)alt?
# o
```

```
ok #.
->
```

```
-> stop /SYS
Are you sure you want to stop /SYS (y/n)? y
Stopping /SYS
```

```
->
```

即時シャットダウンを実行する必要がある場合は、`stop -force -script /SYS` コマンドを使用します。このコマンドを入力する前に、すべてのデータが保存されていることを確認してください。

例 9-2 サービスプロセッサの電源をオンにする

次の例は、サーバーの電源をオンにする方法を示しています。最初に Oracle ILOM にログインする必要があります。<http://download.oracle.com/docs/cd/E19166-01/E20792/z40002fe1296006.html#scrolltoc> を参照してください。

モジュラーシステムを使用している場合は、目的のサーバーモジュールにログインしていることを確認します。

```
-> start /SYS
Are you sure you want to start /SYS (y/n) ? y
Starting /SYS
```

```
->
```

確認を求めるプロンプトを表示しない場合は、`start -script /SYS` コマンドを使用します。

▼ シングルユーザーモードでブートして、不正な **root** シェルまたはパスワードの問題を解決する方法

- 1 システムを **ok PROM** プロンプトにします。81 ページの「復旧を目的としてシステムを停止する方法」を参照してください。
- 2 システムをシングルユーザーモードでブートします。
`ok boot -s`
- 3 `/etc/passwd` ファイルのシェルエントリを修正します。
`# vi /etc/password`
- 4 **Ctrl-d** を押してシステムをリブートします。

▼ メディアからブートして、不明な **root** パスワードを解決する方法

- 1 **Oracle Solaris** のメディアからシステムをブートします。
 - テキストインストール-インストールメディアまたはネットワークからブートして、テキストインストール画面から「オプション3 シェル」を選択します。
 - 自動インストール-次のコマンドを使用して、シェルに出ることができるインストールメニューから直接ブートします。
`ok boot net:dhcp`
- 2 シェルプロンプトで、ルートプールをインポートします。
`# zpool import -f rpool`
- 3 ブート環境のマウントポイントを作成します。
`# mkdir /a`
- 4 ブート環境をマウントします。
`# beadm mount solaris-instance|bename /a`
- 5 **TERM** タイプを設定します。
`# TERM=vt100`
`# export TERM`
- 6 不明なパスワードエントリを慎重に削除します。
`# cd /a/etc`
`# vi shadow`
`# cd /`

注- この手順のあとにディレクトリを変更する必要があります。

- 7 ブートアーカイブを更新します。

```
# bootadm update-archive -R /a
```

- 8 ブート環境をマウント解除します。

```
# beadm umount be-name
```

- 9 システムを停止します。

```
# halt
```

- 10 シングルユーザーモードでシステムをリブートし、**root** パスワードの入力を求められたら、**Return** キーを押します。例:

```
ok boot -s
Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a File and args: -s
SunOS Release 5.11 Version 11.0 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights
reserved.
Booting to milestone "milestone/single-user:default".
Hostname: tardis.central
Requesting System Maintenance Mode
SINGLE USER MODE
Enter user name for system maintenance (control-d to bypass): root
Enter root password (control-d to bypass): <Press return>
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode
```

- 11 **root** パスワードをリセットします。

```
root@system:~# passwd -r files root
New Password: xxxxxx
Re-enter new Password: xxxxxx
passwd: password successfully changed for root
```

- 12 **Ctrl-d** を押してシステムをリブートします。

参照 システムのデフォルト OS がブートできないために、代替の ZFS データセットからブートする必要がある場合、詳細なトラブルシューティング情報について [68 ページ](#) の「SPARC プラットフォームの ZFS ブート環境からのブート」を参照してください。

▼ どのサービスも起動しないでシステムをブートする方法

サービスの起動に関する問題が発生した場合、ブートプロセス中にシステムがハングアップすることがあります。この手順は、そうした問題のトラブルシューティングを行う方法を示しています。

- 1 どのサービスも起動しないでブートします。
このコマンドを実行すると、`svc.startd` デーモンはすべてのサービスを一時的に無効にし、コンソール上で `sulogin` を起動します。
`ok boot -m milestone=none`
- 2 システムにログインし、`root` の役割になります。
- 3 すべてのサービスを有効にします。
`# svcadm milestone all`
- 4 ブートプロセスがハングアップした場合、実行されていないサービスと、ブートプロセスがハングアップしている場所を判定します。
`# svcs -a`
- 5 `/var/svc/log` のログファイル内でエラーメッセージの有無を確認します。
- 6 問題が解決したら、すべてのサービスが起動していることを確認します。
 - a. 必要なサービスがすべてオンラインになっていることを確認します。
`# svcs -x`
 - b. `console-login` サービスの依存関係に問題がないことを確認します。
このコマンドを使えば、コンソール上の `login` プロセスが実行されるかどうかを確認できます。
`# svcs -l system/console-login:default`
- 7 通常のブートプロセスを継続します。

SPARC ベースシステムのクラッシュダンプとリブートの強制実行

場合によっては、トラブルシューティングの目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できます。

システムクラッシュダンプについての詳細は、『[Oracle Solaris の管理: 一般的なタスク](#)』の第 17 章「システムクラッシュ情報の管理(手順)」を参照してください。

▼ クラッシュダンプを強制してシステムをリブートする方法

システムのクラッシュダンプを強制実行するには、次の手順を実行します。この手順のあとに紹介する例は、`halt -d` コマンドを使用してシステムのクラッシュダンプを強制実行する方法を示しています。このコマンドの実行後は、手動でシステムをリブートする必要があります。

- 1 システムを **ok PROM** プロンプトにします。
- 2 ファイルシステムを同期させ、クラッシュダンプを書き出します。

```
> n
ok sync
```

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

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

```
hostname console login:
```

例 9-3 SPARC: `halt -d` コマンドを使用してシステムのクラッシュダンプとリブートを強制実行する

この例は、`halt -d` コマンドと `boot` コマンドを使用して、システムのクラッシュダンプとリブートを強制実行する方法を示しています。

```
# halt -d
Jul 21 14:13:37 jupiter halt: halted by root

panic[cpu0]/thread=30001193b20: forced crash dump initiated at user request

000002a1008f7860 genunix:kadmin+438 (b4, 0, 0, 0, 5, 0)
%l0-3: 0000000000000000 0000000000000000 0000000000000004 0000000000000004
%l4-7: 000000000000003cc 0000000000000010 0000000000000004 0000000000000004
000002a1008f7920 genunix:uadmin+110 (5, 0, 0, 6d7000, ff00, 4)
%l0-3: 0000030002216938 0000000000000000 0000000000000001 0000004237922872
%l4-7: 000000423791e770 0000000000004102 0000030000449308 0000000000000005

syncing file systems... 1 1 done
dumping to /dev/dsk/c0t0d0s1, offset 107413504, content: kernel
100% done: 5339 pages dumped, compression ratio 2.68, dump succeeded
Program terminated
ok boot
Resetting ...

.
.
Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: kernel/sparcv9/unix
```

```

configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: jupiter
The system is coming up. Please wait.
NIS domain name is example.com
.
.
.
System dump time: Wed Jul 21 14:13:41 2010
Jul 21 14:15:23 jupiter savecore: saving system crash dump
in /var/crash/jupiter/*.0
Constructing namelist /var/crash/jupiter/unix.0
Constructing corefile /var/crash/jupiter/vmcore.0
100% done: 5339 of 5339 pages saved
.
.
.

```

▼ カーネルデバッグ (kmdb) を有効にしてシステムをブートする方法

この手順では、カーネルデバッグ (kmdb) をロードする方法を示します。

注-システムを対話式でデバッグする時間がない場合は、reboot コマンドと、-d オプションを指定した halt コマンドを使用します。-d オプションを指定して halt コマンドを実行した場合、実行後にシステムを手動でリブートする必要があります。ただし、reboot コマンドを使用した場合、システムは自動的にブートします。詳細は、[reboot\(1M\)](#) のマニュアルページを参照してください。

- 1 システムを停止します。これにより、**ok** プロンプトが表示されます。
システムを正常に停止するには、halt コマンドを使用します。
- 2 **boot kmdb** または **boot -k** のどちらかを入力して、カーネルデバッグの読み込みを要求します。**Return** キーを押します。
- 3 カーネルデバッグにアクセスします。
デバッグに入るための方法は、システムへのアクセスに使用するコンソールのタイプによって異なります。
 - ローカル接続されたキーボードを使用している場合は、キーボードの種類に応じて **Stop-A** または **L1-A** を押します。
 - シリアルコンソールを使用している場合は、そのシリアルコンソールタイプに適した方法を使用してブレイクを送信します。

カーネルデバッガをはじめて入力したときに、ウェルカムメッセージが表示されます。

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,8000000/esp@4,8800000/sd@3,0
.
.
.
```

例 9-4 SPARC: カーネルデバッガ (kadb) を有効にしてシステムをブートする

```
ok boot kadb
Resetting...

Executing last command: boot kadb -d
Boot device: /pci@1f,0/ide@d/disk@0,0:a File and args: kadb -d
Loading kadb...
```

索引

B

banner コマンド (PROM), 55
boot-archive サービスの管理, 76-77
boot-file パラメータ, 変更, 60
bootadm コマンド, ブートアーカイブの管理に使用, 76-77

H

halt コマンド, 36

I

init コマンド, 説明, 35
init 状態, 「実行レベル」を参照

L

-L ブートオプション, ZFS ルートファイルシステムのブート, 68-72

O

Oracle Solaris ブート動作, 管理方法, 53-60

P

poweroff コマンド, 36

R

reboot コマンド, 36

S

shutdown コマンド
 サーバーのシャットダウン (方法), 37
 説明, 35
sync コマンド, 86
sync コマンドによるファイルシステムの同期化, 86

W

who コマンド, 26

Z

ZFS ブート, ブートオプション, 68-72
ZFS ブート環境の作成と管理、およびこのブート環境からのブート, (タスクマップ), 62
ZFS ルートファイルシステム, からのブート, 68-72
ZFS ルートファイルシステムからのブート, 68-72
-z ブートオプション, ZFS ルートファイルシステムのブート, 68-72

か

確認する, システムの実行レベル, 26
管理, ブート動作, 53-60

こ

高速リブート, 開始する方法, 44-45

し

システムシャットダウンコマンド, 35
システム状態
 NFS によるマルチユーザー
 ブート, 27
 シングルユーザー
 ブート, 28
システムの高速リブートの開始, 方法, 44-45
システムのシャットダウン
 shutdown および init コマンドによる正常
 な, 35
 ガイドライン, 34-36
 (タスクマップ), 33-34
 復旧の目的, 80
システムのブート, ガイドライン, 16-17
システムのブート可能状態の維持, 73-74
システムのリブート, (タスクマップ), 41-42
実行レベル
 0 (電源切断レベル), 19
 1 (シングルユーザーレベル), 19
 2 (マルチユーザーレベル), 19
 3 (NFS によるマルチユーザー)
 システムの実行レベルが移行したときの処
 理, 20
 3 (NFS を持つマルチユーザー), 19
 6 (リブートレベル), 19
 s または S (シングルユーザー状態)
 ブート, 28
 s または S (シングルユーザーレベル), 19
 定義, 18
 デフォルトの実行レベル, 18
実行レベル 3
 NFS によるマルチユーザー
 ブート, 27

指定された状態へのシステムのブート, (タスク
マップ), 25-26
シングルユーザーレベル, 「実行レベル s または
S」を参照

せ

正常なシャットダウン, 35

ね

ネットワークからのシステムのブート, (作業
マップ), 47-48

は

パラメータ, boot-file の変更, 60

ふ

ブート
 実行レベル S, 28
 対話式 (方法), 30
 トラブルシューティングの方法, 80
ブートアーカイブ
 管理, 73-78
ブートオプション
 -L
 ZFS ルートファイルシステム, 68-72
 -Z
 ZFS ルートファイルシステム, 68-72
ブート動作, 管理, 53-60
ブートのトラブルシューティング
 (タスクマップ), 80
 方法, 80
ブートパラメータの変更, 60
 (タスクマップ), 54
復旧のためのシャットダウン, ブートのトラブル
 シューティング, 80

ま

マルチユーザーレベル, 「実行レベル 3」を参照

