

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

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 x86 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
ビットマップコンソールのサポート	15
ブートおよびシャットダウンアニメーション	15
高速リブート	15
x86: 32 ビットカーネルのサポートの削除	16
x86 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
x86 ベースシステムをブートするためのガイドライン	17
システムをブートする理由	17
サービス管理機構とブート	18
SMF 使用時のブート動作に関する変更	19
実行レベルの動作	20
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	21
実行レベルとマイルストーンの使い分け	21
Oracle Solaris ブートアーキテクチャーの概要	22
x86 ブートプロセスの動作	22
GRUB ベースのブート	23
GRUB のコンポーネント	24
GRUB メニューの目的と機能	24
GRUB デバイス命名規則	25
x86 および GRUB ブートの用語	25
2 指定された状態への x86 ベースシステムのブート (タスク)	27
指定された状態への x86 ベースシステムのブート (タスクマップ)	27

指定された状態への x86 ベースシステムのブート	28
システムの現在の実行レベルの確認	28
マルチユーザー状態 (実行レベル 3) への x86 ベースシステムのブート	29
シングルユーザー状態 (実行レベル S) への x86 ベースシステムのブート	30
x86 ベースシステムの対話式ブート	31
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	41
システムデバイスの電源の切断	41
4 x86 ベースシステムのリブート (タスク)	43
x86 ベースシステムのリブート (タスクマップ)	43
x86 ベースシステムのリブート	44
▼ init コマンドを使用してシステムをリブートする方法	45
▼ reboot コマンドを使用してシステムをリブートする方法	46
x86 ベースシステム上のリブートプロセスの高速化	46
▼ BIOS をバイパスしてシステムをリブートする方法	47
新しくアクティブにしたブート環境または代替ブート環境へのシステムのリブートの開始	47
高速リブート機能のデフォルト動作の変更	49
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	50
5 x86 ベースシステムのネットワークからのブート (作業)	51
x86 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)	51
x86 システムをネットワークからブートする	52
x86 ネットワークブートプロセス	52
x86 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	53
▼ x86 ベースシステムをネットワークからブートする方法	54

6	x86 ベースシステム上のブートパラメータの変更 (タスク)	55
	x86 ベースシステム上のブートパラメータの変更 (タスクマップ)	55
	x86 ベースシステム上のブートパラメータの変更	57
	eeprom コマンドを使用したブートパラメータの表示および設定	58
	▼ eeprom コマンドを使用してブートパラメータを変更する方法	58
	ブート時のブートパラメータの変更	59
	ビットマップコンソールのサポート	61
	シャットダウンアニメーションの無効化	63
	menu.lst ファイルの編集によるブートエントリとパラメータの変更	64
	bootadm コマンドを使用したブートエントリのパラメータの表示および設定	67
7	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)	69
	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)	70
	ブート環境の作成と管理	71
	▼ 新しいブート環境を作成する方法	71
	▼ ブート環境のスナップショットの作成方法	73
	▼ 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	73
	▼ 新しく作成されたブート環境をアクティブにする方法	73
	▼ 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	74
	▼ ブート環境を破棄する方法	75
	x86 プラットフォーム上の ZFS ブート環境またはルートファイルシステムからのブート	76
8	x86 ベースシステムのブート可能状態の維持 (タスク)	79
	x86 ベースシステムのブート可能状態の維持 (タスクマップ)	79
	Oracle Solaris ブートアーカイブの説明	80
	x86 ブートアーカイブの場所と内容に関する情報の取得	81
	▼ ブートアーカイブの内容を一覧表示する方法	81
	ブートアーカイブ SMF サービスの管理	81
	boot-archive サービスが実行中かどうかを確認する	81
	▼ boot-archive SMF サービスを有効または無効にする方法	82
	ブートアーカイブの整合性の維持	82
	▼ auto-reboot-safe プロパティを使用してブートアーカイブ自動更新障害を消去	

する方法	83
▼ 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	83
9 x86 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)	85
x86 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)	86
復旧を目的とした x86 ベースシステムのシャットダウンおよびブート	87
システム復旧のための停止とブート	87
クラッシュダンプを強制してシステムをリブートする	91
▼ カーネルデバッグ (kmdb) を有効にした状態でシステムをブートする方法	93
x86 プラットフォーム上での高速リブートに関する問題のトラブルシューティング	94
発生する可能性のある早期システムパニックのデバッグ	94
x86 プラットフォーム上で高速リブートが動作できなくなる可能性のある状態のトラブルシューティング	94
 索引	 97

はじめに

『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』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主に、x86 ベースシステムのブートに関する情報が含まれています。ただし、一部の情報は、x86 プラットフォームと SPARC プラットフォームの両方に関連します。

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ー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。

x86 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ぎ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、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の概略の情報とガイドラインを提供します。

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」
- 16 ページの「x86 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)」
- 17 ページの「x86 ベースシステムをブートするためのガイドライン」
- 18 ページの「サービス管理機構とブート」
- 20 ページの「実行レベルの動作」
- 22 ページの「Oracle Solaris ブートアーキテクチャーの概要」
- 23 ページの「GRUB ベースのブート」

SPARC 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、『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』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。

- [14 ページの「管理用に提供された driver.conf ファイル」](#)
- [15 ページの「ビットマップコンソールのサポート」](#)
- [15 ページの「ブートおよびシャットダウンアニメーション」](#)
- [16 ページの「x86: 32 ビットカーネルのサポートの削除」](#)

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って、ドライバ構成にローカルな変更を提供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ーを表示するには、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ー値と変更されたプロパティー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ー値とカスタマイズされたプロパティー値を表示する方法](#)」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\(9F\)](#)

ビットマップコンソールのサポート

Oracle Solaris 11 は、x86 ベースのシステムで、古い VGA (Video Graphics Array) 640-480 16 色コンソールよりも高い解像度と色数をサポートします。このサポートは、従来の BIOS および VESA (Video Electronics Standards Association) オプション読み取り専用メモリー (ROM) を使用するシステムを対象に提供されます。サポートは、物理または仮想コンソールとしてグラフィックカードまたはフレームバッファが使用されるときに限られることに注意してください。シリアルコンソールの動作への影響はありません。

詳細は、[61 ページ](#)の「ビットマップコンソールのサポート」を参照してください。

ブートおよびシャットダウンアニメーション

ブートプロセス中にシステム上で表示される進捗状況インジケータは、次の場合に自動的に割り込まれます:

- カーネルデバッグに移行した。
- システムパニックが発生した。
- 入力が必要な Oracle Solaris サービスのサービス管理機能 (SMF) がブートプロセスに割り込んだ。
- GDM (GNOME Desktop Manager) のログイン画面が表示された。

システムをブートするときに `console=graphics` オプションが指定され、かつ X.org サーバーによってシャットダウンがトリガーされた場合は、シャットダウンプロセス中に進捗ステータスインジケータが表示されます。`svc:/system/boot-config` SMF サービスの新しい `splash-shutdown` プロパティを `false` に設定することにより、進捗状況インジケータを非表示にすることができます。手順については、[63 ページ](#)の「シャットダウンアニメーションの無効化」を参照してください。

高速リブート

高速リブートでは、カーネルをメモリー内に読み込んでからそのカーネルに切り替えるインカーネルブートローダーが実装されます。ファームウェアおよびブートローダーのプロセスがバイパスされ、システムを数秒以内でリブートできるようになります。

高速リブート機能は SMF によって管理され、ブート構成サービス `svc:/system/boot-config` を通して実装されます。`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。

ん。このプロパティの値は、x86 プラットフォームではtrueに設定されています。SPARC プラットフォーム上での高速リブートのデフォルト動作を変更する方法を含むタスク関連の情報については、[46 ページの「x86 ベースシステム上のリブートプロセスの高速化」](#)を参照してください。

x86: 32 ビットカーネルのサポートの削除

Oracle Solaris 11 では、x86 プラットフォームでの 32 ビットカーネルのサポートが削除されました。その結果、32 ビットの x86 プラットフォーム上では Oracle Solaris 11 をブートできません。32 ビットハードウェアを含むシステムは、64 ビットハードウェアにアップグレードするか、または引き続き Oracle Solaris 10 を実行する必要があります。

注- このサポートの削除は、32 ビットアプリケーションには影響しません。x86 プラットフォーム上での 32 ビットアプリケーションに対するサポートは同じままです。

x86 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 x86 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
x86 ベースシステムを指定された状態にします (実行レベルブート)。	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)」
x86 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)」
x86 ベースシステムをリブートします。	第 4 章「x86 ベースシステムのリブート(タスク)」
x86 システムをネットワークからブートします。	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)」
x86 ベースシステム上のデフォルトのブート動作を変更します。	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)」
x86 ベースシステム上で ZFS BE、スナップショット、またはデータセットの作成と管理、およびこれらの環境からのブートを行います。	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)」

表 1-1 x86 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
ブート管理インタフェース (bootadm) を使用することによって x86 ベースシステムをブート可能に維持します。	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)」
x86 ベースシステムのブートをトラブルシューティングします。	第 9 章「x86 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)」

x86 ベースシステムをブートするためのガイドライン

システムをブートするときは、次のガイドラインに従ってください。

- シャットダウンした x86 ベースシステムは、GRUB メニューでオペレーティングシステムを選択することによってブートします。オペレーティングシステムが選択されていない場合、システムは、menu.lst ファイルで指定されているデフォルトのオペレーティングシステムをブートします。
- 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 - このシャットダウン方法をサポートしている リリースが稼働する x86 システムを使用するのではないかぎり、この方法は正常なシャットダウンとは言えません。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

次の表に示す理由により、x86 ベースシステムのブートが必要になることがあります。また、システム管理タスクと、これらのタスクを完了するために使用される対応するブートオプションについても説明します。

表 1-2 システムのブート

システムリブートの理由	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 (NFS リソースを共有する実行レベル 3)	29 ページの「マルチユーザー状態 (実行レベル 3) への x86 ベースシステムのブート」

表 1-2 システムのブート (続き)

システムリブートの理由	適切なブートオプション	参照先
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	30 ページの「シングルユーザー状態 (実行レベル S) への x86 ベースシステムのブート」
/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	31 ページの「x86 ベースシステムの対話式ブート」
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	92 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	93 ページの「カーネルデバッグ (kmdb) を有効にした状態で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 svcsvcs 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 マイルストーンを使用する方法については、『Oracle Solaris の管理: 一般的なタスク』の「どのサービスも起動しないでブートする方法」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 使用時のブート動作に関する変更

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。
- サービスは可能な限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	milestone/single-user:default
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ーを読み取ります。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ーを `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、『[Oracle Solaris の管理: 一般的なタスク](#)』の「[どのサービスも起動しないでブートする方法](#)」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 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。詳細は、[80 ページの「Oracle Solaris ブートアーカイブの説明」](#)を参照してください。

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

`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。`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、[82 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。

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

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。

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。この目的で RAM ディスクイメージを使用すると、リムーバブルメディアからシステムをブートする必要がなくなります。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。

x86 ブートプロセスの動作

このセクションでは Oracle Solaris x86 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> にある特定のハードウェアの製品ドキュメントを参照してください。

x86 ベースシステムの電源を入れると、BIOS によって CPU、メモリー、およびプラットフォームハードウェアが初期化されます。BIOS フェーズが完了すると、構成されたブートデバイスからブートローダーが読み込まれ、システムの制御がブート

ローダーに移されます。それによって、ブートプロセスが開始されます。「ブートローダー」は、システムの電源を投入したあとに最初に動作するソフトウェアプログラムです。このプログラムによってブートプロセスが開始されます。Oracle Solaris では、GRand Unified Bootloader (「GRUB」とも呼ばれる) が x86 ベースシステム上のデフォルトのブートローダーです。

GRUB ベースのブート

Oracle Solaris では、オープンソースの GRUB (GRand Unified Bootloader) が x86 ベースシステム上のデフォルトのブートローダーです。GRUB は、ブートアーカイブをシステムのメモリーにロードするソフトウェアです。ブートアーカイブは、システムの起動中にルートファイルシステムがマウントされる前に必要な、重要なファイルの集まりです。ブートアーカイブとは、Oracle Solaris のブートに使用されるインタフェースのことです。GRUB の詳細は、<http://www.gnu.org/software/grub/grub.html> を参照してください。grub(5) のマニュアルページも参照してください。

GRUB は、menu.lst ファイルと呼ばれる構成ファイルに定義済みのブートオプションが含まれているメニューインタフェースを実装します。また、デフォルトのブートパラメータの変更など、さまざまなブート機能の実行に使用できる GUI メニューインタフェースからアクセス可能なコマンド行インタフェースも備えています。

Oracle Solaris カーネルは、マルチブート仕様に完全に準拠しています。GRUB を使用すると、1 つのシステムにインストールされる可能性のある異なるオペレーティングシステムをブートできます。たとえば、システムのブート時に GRUB メニューで適切なブートエントリを選択することによって、Oracle Solaris、Linux、または Windows を個別にブートできます。または、デフォルトで特定の OS インスタンスをブートするように menu.lst ファイルをカスタマイズできます。

GRUB では、ファイルシステムとカーネル実行可能ファイルの形式が認識されるため、ディスク上のカーネルの物理的な位置を記録しておかなくても、オペレーティングシステムを読み込むことができます。GRUB ベースのブートでは、カーネルはそのファイル名だけでなく、カーネルが存在するドライブとパーティションを指定することによって読み込まれます。

GRUB のコンポーネント

GRUB ブートローダーのコンポーネントは次のとおりです。

- `stage1` – `fdisk` パーティションの最初のセクターにインストールされるイメージです。`installgrub` コマンドに `-m` オプションを指定して、`stage1` をマスターブートセクターにインストールすることもできます。詳細は、[installgrub\(1M\)](#) のマニュアルページおよび『Oracle Solaris の管理: デバイスとファイルシステム』の「GRUB ブート環境でのディスク管理」を参照してください。
- `stage2` – `fdisk` パーティションの予約済み領域にインストールされるイメージです。`stage2` イメージは GRUB のコアイメージです。
- `menu.lst` ファイル – 通常は、ZFS ルートファイルシステムを含むシステム上の `/pool-name/boot/grub` ディレクトリにあります。ここで、`/pool-name/boot/grub` は ZFS ストレージプールの名前です。このファイルは、GRUB `stage2` ファイルによって読み取られます。詳細は、[64 ページの「menu.lst ファイルの編集によるブートエントリとパラメータの変更」](#)のセクションを参照してください。

`dd` コマンドを使用して、`stage1` および `stage2` のイメージをディスクに書き込むことはできません。`stage1` イメージは、ディスク上にある `stage2` イメージの場所に関する情報を受け取れる必要があります。`installgrub` コマンドを使用してください。このコマンドは、GRUB ブートブロックをインストールするためにサポートされている方法です。

GRUB メニューの目的と機能

x86 ベースのシステムをブートするときに表示されるメニューは GRUB メニューです。このメニューは、GRUB の `menu.lst` ファイル内の構成情報に基づいています。ブートシーケンスが始まると、GRUB メニューが表示されます。ブートシーケンスに割り込まない場合、デフォルトエントリ (通常は `menu.lst` ファイルの最初のエントリ) がデフォルトでブートされます。

ブート時に GRUB メニューを編集して、別のオペレーティングシステムのブート、またはデフォルトブートエントリのパラメータの変更を実行できます。そのためには、GRUB メニューが表示されたらすぐに `e` を入力します。`e` を入力すると、ブートプロセスに割り込んで、GRUB 編集メニューが表示されます。ここでは、ブートする別の OS の選択、またはデフォルトブートエントリのデフォルトブートパラメータの変更を行うことができます。変更されたブート動作は、システムの次回ブート時までしか持続しないことに注意してください。手順については、[59 ページの「ブート時のブートパラメータの変更」](#)を参照してください。

GRUB デバイス命名規則

GRUB では、以前のリリースで使用されていたものとは若干異なるデバイス命名規則が使用されます。GRUB で使用されるデバイス命名規則を理解しておく、システム上で GRUB を構成するときにドライブやパーティションの情報を正しく指定するのに役立ちます。

次の表で、GRUB で使用されるデバイス命名規則について説明します。

表 1-5 GRUB デバイスの規則

デバイス名	説明
(fd0)	最初のコピーディスク
(fd1)	2 番目のコピーディスク
(nd)	ネットワークデバイス
(hd0,0)	最初のディスク上の最初の fdisk パーティション
(hd0,1)	最初のディスク上の 2 番目の fdisk パーティション
(hd0,0,a),	最初のディスク上の最初の fdisk パーティション上のスライス a
(hd0,0,b)	最初のディスク上の最初の fdisk パーティション上のスライス b

注 - GRUB デバイス名はすべて括弧で囲む必要があります。

Solaris 10 10/08 リリース以降では、GRUB で以前使用されていた `root` コマンドが `findroot` コマンドに置き換われました。`findroot` コマンドは、ブートデバイスには関係なく、対象のディスクを検出するための拡張機能を提供します。

x86 および GRUB ブートの用語

x86 ベースシステムのブートおよびシャットダウンには、次の基本的な用語が使用されます。

BIOS (Basic Input/Output System) x86 ベースシステムでは、BIOS は、電源投入時に PC によって最初に行われるコードになるように設計されたブートファームウェアです。BIOS の最初の機能は、ビデオディスプレイ

	カード、ハードディスク、フロッピーディスク、その他のハードウェアなどのシステムデバイスを識別、テスト、および初期化することです。
ブートアーカイブ	Oracle Solaris OS のブートに使用されるクリティカルなファイルの集まり。これらのファイルは、ルートファイルシステムがマウントされる前、システムの起動中に必要です。
ブートローダー	システムの電源を投入したあとに最初に行われるソフトウェアプログラム。このプログラムがブートプロセスを開始します。
GRUB (GRand Unified Bootloader)	GRUB は、x86 ベースシステム上で使用されるマルチブートブートローダーです。ブートローダーは、システムの起動時に最初に行われるソフトウェアプログラムです。オペレーティングシステムのカーネルソフトウェア (Oracle Solaris、Linux、および Windows) を読み込み、そのカーネルソフトウェアに制御を移す役割を果たします。
GRUB 編集メニュー	GRUB メインメニューのサブメニュー。このサブメニューには、GRUB コマンドが表示されます。これらのコマンドを編集して、ブート動作を変更できます。
GRUB メインメニュー	システムにインストールされているオペレーティングシステムがリストされたブートメニュー。このメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
menu.lst ファイル	システムにインストールされているすべてのオペレーティングシステムがリストされた構成ファイル。このファイルの内容は、GRUB メニューに表示されるオペレーティングシステムの一覧を記述したものです。GRUB のメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。

指定された状態への x86 ベースシステムのブート (タスク)

この章では、x86 ベースシステムをさまざまなシステム状態 (「実行レベル」とも呼ばれる) にブート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。

- 27 ページの「指定された状態への x86 ベースシステムのブート (タスクマップ)」
- 28 ページの「指定された状態への x86 ベースシステムのブート」

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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 SPARC ベースシステムのブートについては、『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』の第 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)」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 (タスクマップ)

表 2-1 指定された状態への x86 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。	28 ページの「システムの現在の実行レベルの確認」
x86 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。	29 ページの「マルチユーザー状態 (実行レベル 3) への x86 ベースシステムのブート」

表 2-1 指定された状態への x86 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
x86 ベースのシステムをシングルユーザー状態でブートします。	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。	30 ページの「シングルユーザー状態 (実行レベル S) への x86 ベースシステムのブート」
x86 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。	31 ページの「x86 ベースシステムの対話式ブート」

指定された状態への x86 ベースシステムのブート

次の手順では、指定された状態に x86 ベースシステムをブートする方法 (実行レベルのブートとも呼ばれる) を説明しています。

システムの現在の実行レベルの確認

システムの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
$
```

who -r コマンドの出力	説明
run-level 3	現在の実行レベル
Dec 13 10:10	実行レベルが最後に変更された日時
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
S	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 x86 ベースシステムのブート

システムの電源が切れている場合に電源を投入するか、または `reboot` コマンドを使用すると、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。28 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 x86 ベースシステムを実行レベル 3 にブートします。

- 1 システムをリブートします。

```
# reboot
```

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メインメニューが表示されます。

- 2 GRUB メインメニューが表示されたら、**Enter** キーを押してデフォルトの OS インスタンスをブートします。

10 秒以内にエントリを選択しないと、システムは自動的に実行レベル 3 にブートします。

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。

- 3 システムにログインします。

```
hostname console login:
```

- 4 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。

```
$ who -r
.          run-level 3  Mar  2 09:44    3      0  S
```

シングルユーザー状態 (実行レベル S) への x86 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 システムをリブートします。

reboot

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メインメニューが表示されます。

- 2 GRUB メインメニューが表示されたら、**e**と入力して **GRUB** メニューを編集します。
- 3 実行しているリリースによっては、矢印キーを使用して **kernel\$** 行を選択します。矢印キーを使用できない場合は、キャレット (^) キーを使って上方へスクロールしたり、英字キー **v** を使って下方へスクロールしたりします。
- 4 **e**と再度入力してブートエントリを編集します。
ここから、**kernel** 行または **kernel\$** 行にオプションと引数を追加できます。
- 5 システムをシングルユーザー状態でブートするには、ブートエントリ行の末尾に **-s** を入力してから、**Return** キーを押して前の画面に戻ります。

注 - 他のブート動作を指定するには、**-s** オプションを適切なブートオプションに置き換えます。

この方法で、次の代替ブート動作を指定できます。

- 再構成用ブートを実行する
- カーネルデバッグを使ってシステムをブートする
- コンソールをリダイレクトする

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

- 6 システムをシングルユーザー状態でブートするには、**b**と入力します。

- 7 プロンプトが表示されたら、**root** パスワードを入力します。
- 8 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
.          run-level S   Jun 13 11:07      S      0  0
```
- 9 実行レベルを **S** に移行して行う必要があるシステム保守タスクを行います。
- 10 システムの保守タスクが完了したら、システムをリブートします。

x86 ベースシステムの対話式ブート

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。または、代替ブート環境を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。[47 ページの「新しくアクティブにしたブート環境または代替ブート環境へのシステムのリブートの開始」](#)を参照してください。

▼ システムを対話式でブートする方法

- 1 `/etc/system` ファイルおよび `boot/solaris/filelist.ramdisk` ファイルのバックアップコピーを作成し、`/etc/system.bak` 内のファイル名を `/boot/solaris/filelist.ramdisk` ファイルに追加します。例:

```
# cp /etc/system /etc/system.bak
# cp /boot/solaris/filelist.ramdisk /boot/solaris/filelist.ramdisk.orig
# echo "etc/system.bak" >> /boot/solaris/filelist.ramdisk
```
- 2 ブートアーカイブを更新します。

```
# bootadm update-archive -v
```
- 3 システムをリブートします。

```
# reboot
```
- 4 GRUB メニューが表示されたら、ブートする OS を対話式で選択し、**e** と入力します。
- 5 矢印キーを使用して、`kernel$` 行を選択し、**e** と入力して、指定したブートエントリを編集します。
- 6 行の終わりに **-a** と入力し、**Return** キーを押します。
- 7 **b** と入力して、対話式でシステムをブートします。

- 8 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。
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. 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。
- 9 破損した **/etc/system** ファイルを修復します。
- 10 システムを実行レベル 3 にリブートします。
reboot

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。x86 ベースシステムのシャットダウン手順は、SPARC ベースシステムのシャットダウン手順と同じです。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」
- 41 ページの「システムデバイスの電源の切断」

x86 ベースシステムのブートに関する概略の情報は、[第 1 章「x86 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。

SPARC 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、『[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](#)』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。	システムがサーバーである場合は、who 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。	41 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、[20 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。

システムのアボートキーシーケンスを使用したり、電源をオフにしてからオンにしたりする方法では、システムサービスが突然終了してしまうので、正常なシャットダウン方法とはいえません。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 5 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。

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
init	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。	このコマンドではシステムシャットダウンが高速化されるため、ほかのユーザーに影響を与えないスタンドアロンシステムのシャットダウンには、このコマンドが推奨されます。すぐにシャットダウンされることについての通知は送信されません。
reboot	ディスクを同期させ、ブート命令を uadmin 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。	init 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 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
```

ログインしているすべてのユーザーがリスト表示されます。

- 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	x86 ベース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)	#
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システム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、3 分後にシステムを実行レベル S (シングルユーザー状態) にしています。

```
# who
root      console      Apr 15 06:20

# shutdown -g180 -y

Shutdown started.      Fri Apr 15 06:20:45 MDT 2011

Broadcast Message from root (console) on portia Fri Apr 15 06:20:46...
The system portia will be shut down in 3 minutes

showmount: portia: RPC: Program not registered
Broadcast Message from root (console) on portia Fri Apr 15 06:21:46...
The system portia will be shut down in 2 minutes

showmount: portia: RPC: Program not registered
Broadcast Message from root (console) on portia Fri Apr 15 06:22:46...
The system portia will be shut down in 1 minute

showmount: portia: RPC: Program not registered
Broadcast Message from root (console) on portia Fri Apr 15 06:23:16...
The system portia will be shut down in 30 seconds

showmount: portia: RPC: Program not registered
Changing to init state s - please wait
svc.startd: The system is coming down for administration. Please wait.
root@portia:~# Apr 15 06:24:28 portia svc.startd[9]:

Apr 15 06:24:28 portia syslogd: going down on signal 15
svc.startd: Killing user processes.
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
SINGLE USER MODE

Enter user name for system maintenance (control-d to bypass):xxxxxx
#
```

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5分後にシステムを実行レベル0にしています。

```
# who
root          console      Jun 17 12:39...
userabc       pts/4          Jun 17 12:39   (:0.0)
# shutdown -i0 -g300 -y
Shutdown started.   Fri Apr 15 06:35:48 MDT 2011

Broadcast Message from root (console) on murky Fri Apr 15 06:35:48...
The system pinkytusk will be shut down in 5 minutes

showmount: murkey: RPC: Program not registered
showmount: murkey: RPC: Program not registered
Broadcast Message from root (console) on murky Fri Apr 15 06:38:48...
The system murkey will be shut down in 2 minutes

showmount: murkey: RPC: Program not registered
Broadcast Message from root (console) on murky Fri Apr 15 06:39:48...
The system murkey will be shut down in 1 minute

showmount: murkey: RPC: Program not registered
Broadcast Message from root (console) on murky Fri Apr 15 06:40:18...
The system murkey will be shut down in 30 seconds

showmount: murkey: RPC: Program not registered
Broadcast Message from root (console) on murky Fri Apr 15 06:40:38...
THE SYSTEM murkey IS BEING SHUT DOWN NOW !!!
Log off now or risk your files being damaged

showmount: murkey: RPC: Program not registered
Changing to init state 0 - please wait
root@murkey:~# svc.startd: The system is coming down. Please wait.
svc.startd: 122 system services are now being stopped.
Apr 15 06:41:49 murkey svc.startd[9]:
Apr 15 06:41:50 murkey syslogd: going down on signal 15
svc.startd: Killing user processes.
Apr 15 06:41:57 The system is down. Shutdown took 69 seconds.
syncing file systems... done
Press any key to reboot.
Resetting...
```

すべてのデバイスの電源を切断するためにシステムを実行レベル0にする場合は、[41 ページの「システムデバイスの電源の切断」](#)を参照してください。

例3-3 shutdown コマンドを使用してシステムをマルチユーザー状態 (実行レベル3) にする

次の例では、shutdown コマンドを使用して、2分後にシステムを実行レベル3にリブートしています。確認用プロンプトが表示されないようにyオプションを指定しています。

```
# who
root          console      Jun 14 15:49    (:0)
userabc      pts/4          Jun 14 15:46    (:0.0)
# shutdown -i6 -g120 -y
Shutdown started.    Fri Apr 15 06:46:50 MDT 2011

Broadcast Message from root (console) on venus Fri Apr 15 06:46:50...
The system venus will be shut down in 2 minutes

showmount: venus: RPC: Program not registered
showmount: venus: RPC: Program not registered
Broadcast Message from root (console) on venus Fri Apr 15 06:47:50...
The system venus will be shut down in 1 minute

showmount: venus: RPC: Program not registered
showmount: venus: RPC: Program not registered
Broadcast Message from root (console) on venus Fri Apr 15 06:48:20...
The system venus will be shut down in 30 seconds

showmount: venus: RPC: Program not registered
Broadcast Message from root (console) on venus Fri Apr 15 06:48:40...
THE SYSTEM venus IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged

showmount: venus: RPC: Program not registered
Changing to init state 6 - please wait
root@venus:~# svc.startd: The system is coming down. Please wait.
svc.startd: 123 system services are now being stopped.
Apr 15 06:49:32 venus svc.startd[9]:
Apr 15 06:49:32 venus syslogd: going down on signal 15
svc.startd: Killing user processes.
Apr 15 06:49:40 The system is down. Shutdown took 50 seconds.
syncing file systems... done
rebooting...
SunOS Release 5.11 Version 2010-12-10 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is solaris.us.oracle.com
.
.
.
venus console login:
```

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、[29 ページの「マルチユーザー状態\(実行レベル3\)への x86 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 3-4 **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、x86 ベースのスタンドアロンシステムを安全な電源切断が可能な実行レベルにしています。

```
# 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、[29 ページの「マルチユーザー状態 \(実行レベル 3\) への x86 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

次のような場合は、すべてのシステムデバイスの電源を落とす必要があります。

- ハードウェアを置換または追加する。
- システムの設置場所を変更する。
- 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。

x86 ベースシステムのリブート (タスク)

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、x86 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。

- 43 ページの「x86 ベースシステムのリブート (タスクマップ)」
- 44 ページの「x86 ベースシステムのリブート」
- 46 ページの「x86 ベースシステム上のリブートプロセスの高速化」

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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。

SPARC ベースシステムのリブートについては、『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』の第 4 章「SPARC ベースシステムのリブート (手順)」を参照してください。

x86 ベースシステムのリブート (タスクマップ)

表 4-1 x86 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。	45 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 x86 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。	46 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」
BIOS をバイパスして、x86 ベースシステムのリブートを開始します。	高速リブートはこのリリースでのデフォルトのブートモードであるため、reboot または init 6 コマンドを使用してシステムの高速リブートをブートすることができます。	47 ページの「BIOS をバイパスしてシステムをリブートする方法」
新しく作成されたブート環境への x86 ベースシステムのリブートを開始します。	reboot コマンドで、新しく作成されたブート環境または代替ブート環境を指定することによって、そのブート環境への x86 ベースシステムの高速リブートを開始します。	47 ページの「新しくアクティブにしたブート環境または代替ブート環境へのシステムのリブートの開始」
x86 ベースシステム上の高速リブート機能のデフォルト動作を変更します。	x86 プラットフォーム上では、高速リブートとパニック高速リブートの両方がデフォルトで有効になっており、このどちらも boot-config サービスによって管理されます。これらの機能の一方または両方を無効にすることによって、このデフォルト動作を変更できます。	49 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になっている x86 ベース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 -p オプションとともに使用します。	50 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」

x86 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

reboot コマンドを使用して x86 ベースシステムをリブートすると、高速リブートがデフォルトで開始され、BIOS がバイパスされます。BIOS をバイパスしないでシステムをリブートするには、**reboot** コマンドに **-p** オプションを使用します。[50 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](#)」を参照してください。

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

```
# reboot
```

x86 ベースシステム上のリブートプロセスの高速化

Oracle Solaris の高速リブート機能を使用すると、ファームウェアとブートローダーのプロセスをバイパスして、x86 ベースシステムをリブートできます。高速リブートでは、カーネルをメモリーにロードしてからそのカーネルに切り替えるインカーネルブートローダーを実装することにより、リブート処理を数秒で実行できます。高速リブートとパニック高速リブート(システムパニックのあとのシステムの高速リブート)がデフォルトで有効になっているため、x86 ベースシステムの高速リブートを開始するために **reboot** コマンドで **-f** オプションを使用する必要はありません。

高速リブートのサポートは、新しい **boot-config** サービスである **svc:/system/boot-config:default** によって促進されています。このサービスによって、システムのデフォルトのブート構成プロパティを必要に応じて設定または変更するための手段が提供されます。**config/fastreboot_default** プロパティが **true** に設定されている場合は、システムが自動的に高速リブートを実行します。このプロパティの値は、x86 ベースシステムでは **true** に設定されています。詳細は、[49 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。

新しい OS イメージのブート時にファームウェアをバイパスするシステムの機能は、デバイス動作の新しいエントリポイントである **quiesce** のデバイスドライバでの実装に依存しています。サポートされているドライバでは、この実装によってデバイスが休止されるため、関数の完了時にそのドライバによって割り込みが生成されなくなります。この実装ではまた、デバイスがハードウェア状態にリセットされます。この状態からは、システムの電源を切ってすぐに入れ直したり、ファームウェアで構成したりしなくても、ドライバの **attach** ルーチンによってそのデバイスを正しく構成できます。この機能についての詳細は、[quiesce\(9E\)](#) および [dev_ops\(9S\)](#) のマニュアルページを参照してください。

注-すべてのドライバで `quiesce` 関数が実装されているわけではありません。トラブルシューティングの手順については、94 ページの「[x86 プラットフォーム上で高速リブートが動作できなくなる可能性のある状態のトラブルシューティング](#)」を参照してください。

高速リブートプロセスをさらに詳細に説明したデモを表示するには、http://download.oracle.com/otndocs/tech/OTN_Demos/x86/x86-OTN-Demo/x86-OTN-Demo.html にアクセスしてください。

▼ BIOS をバイパスしてシステムをリブートする方法

注- この Oracle Solaris リリースでは、高速リブートは x86 ベースのシステムでデフォルトの動作モードです。以前は、x86 ベースのシステムで高速リブートをブートするには、`reboot` コマンドで `-f` オプションを指定して、システムの高速リブートをブートする必要がありました。このオプションを指定する必要はなくなりました。

- 1 `root` の役割になります。
- 2 システムの高速リブートをブートするには、次のどちらかのコマンドを入力します:

```
# reboot
```

```
# init 6
```

新しくアクティブにしたブート環境または代替ブート環境へのシステムのリブートの開始

代替ブート環境への x86 ベースシステムの高速リブートを実行するには、複数の方法を使用できます。次の例は、これらの方法のいくつかを示しています。

例 4-2 x86: 新しくアクティブにしたブート環境へのシステムのリブートの開始

次の例は、新しくアクティブになったブート環境 `2010-12-10-be` にシステムの高速リブートを開始する方法を示しています。

```
# bootadm list-menu
the location for the active GRUB menu is: /rpool/boot/grub/menu.lst
default 0
0 oracle solaris 11
1 2010-12-10-be
2 zfsbe2
3 2010-12-10-be-s
```

例 4-2 x86: 新しくアクティブにしたブート環境へのシステムのリブートの開始 (続き)

```
# beadm activate 2010-12-10-be
# reboot
```

例 4-3 x86: 代替ブート環境の指定によるシステムのリブートの開始

代替ブート環境 (たとえば、zfsbe2) にシステムを高速リブートするには、次のコマンドを入力します。

```
# reboot -- 'rpool/zfsbe2'
```

rpool/zfsbe1 という名前のデータセットへのシステムの高速リブートを開始するには、次のコマンドを入力します。

```
# reboot -- 'rpool/zfsbe1'
```

代替 ZFS ルートデータセットへのシステムの高速リブートを開始するには、次のコマンドを入力します。

```
# reboot -- 'rpool/ROOT/zfsroot2'
```

例 4-4 カーネルデバッグを有効にした状態での代替ブート環境へのシステムの高速リブートの開始

カーネルデバッグを有効にした状態で zfsbe3 ブート環境へのシステムの高速リブートを開始するには、次のコマンドを入力します。

```
# reboot -- 'rpool/zfsbe3 /platform/i86pc/kernel/amd64/unix -k'
```

例 4-5 x86: 新しいカーネルへのシステムのリブートの開始

my-kernel という名前の新しいカーネルへのシステムの高速リブートを開始するには、次のコマンドを入力します。

```
# reboot -- '/platform/i86pc/my-kernel/amd64/unix -k'
```

例 4-6 x86: マウントされたディスクまたはマウントされたデータセットのリブートの開始

マウントされたディスクまたはマウントされたデータセットの高速リブートを開始するには、次のコマンドを入力します。

```
# reboot -- '/mnt/platform/i86pc/my-kernel/amd64/unix -k'
```

例 4-7 x86: カーネルデバッグを有効にした状態での、シングルユーザー状態へのシステムのリブートの開始

カーネルデバッグを有効にした状態でシステムのシングルユーザー状態への高速リブートを開始するには、次のコマンドを入力します。

例 4-7 x86: カーネルデバッグを有効にした状態での、シングルユーザー状態へのシステムのリブートの開始 (続き)

```
# reboot -- '-ks'
```

高速リブート機能のデフォルト動作の変更

高速リブート機能は SMF によって制御され、ブート構成サービス `svc:/system/boot-config` を通して実装されます。boot-config サービスは、デフォルトのブートパラメータを設定または変更するための手段を提供します。

boot-config サービスの `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。このプロパティの値は、x86 ベースシステムではデフォルトで `true` に設定されています。

`svc:/system/boot-config:default` サービスは、次のプロパティで構成されています。

- `config/fastreboot_default`
- `config/fastreboot_onpanic`

例 4-8 x86: boot-config サービスのプロパティの構成

boot-config サービスに含まれているプロパティは、`svccfg` および `svcadm` コマンドを使用して構成できます。

たとえば、x86 ベースシステム上の `fastreboot_onpanic` プロパティのデフォルト動作を無効にするには、次に示すように、このプロパティの値を `false` に設定します。

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

あるプロパティの値を変更しても、ほかのプロパティのデフォルト動作には影響を与えないことに注意してください。

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

高速リブート機能が有効になっている x86 ベースシステムをリブートするには、この機能を無効にするように `boot-config` サービスを再構成するのではなく、次に示すように `reboot` コマンドで `-p` オプションを使用します。

```
# reboot -p
```

x86 ベースシステムのネットワークからのブート (作業)

この章では、x86 ベースシステムをネットワークからブートするための概要、ガイドライン、および作業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。

- 51 ページの「x86 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)」
- 52 ページの「x86 システムをネットワークからブートする」

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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。

SPARC 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、『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』の第 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。

x86 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 x86 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
1. x86 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確認します。	最初に、x86 ベースシステムをネットワークからブートするためのすべての要件を確認します。一部の要件には、システムをネットワークからブートできるようにするために実行する必要がある、個別の前提条件となる作業が含まれています。	53 ページの「x86 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」

表 5-1 x86 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
2. x86 ベースシステムをネットワークからブートします。	すべての要件を確認し、予備的な作業をすべて実行したら、システムをネットワークからブートする準備が整います。reboot コマンドを使用して、x86 ベースシステムをネットワークからブートします。	54 ページの「x86 ベースシステムをネットワークからブートする方法」

x86 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。

自動インストーラ (AI) を使用して 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からブートする場合は、追加の AI インストールサービスをすべて実行してください。詳細は、『Oracle Solaris 11 システムのインストール』を参照してください。

x86 ネットワークブートプロセス

Oracle Solaris でのネットワークブートは、PXE (Preboot eXecution Environment) (実行前環境とも呼ばれる) に準拠したファームウェアを通してサポートされています。これは、データストレージデバイス (ハードディスクなど) やインストールされているオペレーティングシステムには関係なく、ネットワークインタフェースを使用してシステムをブートするための環境です。このファームウェアは、ブートプログラム (pxegrub という名前の特殊な GRUB Stage 2 ファイル) を読み込む役割を果たします。pxegrub ファイルには、TFTP (Trivial File Transfer Protocol)、DHCP、UDP (ユーザーデータグラムプロトコル)、IP (インターネットプロトコル)、およびネットワーク全体にパケットを転送するために UNDI (Universal Network Device Interface) ファームウェアインタフェースを使用するミニドライバなどの基本的な実装が含まれています。

x86 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。復旧を目的として、または Oracle Solaris をインストールするために、スタンドアロン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。x86 ベースのシステムは、PXE ネットワークブートプロトコルをサポートするネットワークから直接ブートすることができます。PXE プロトコルは、Intel Preboot Execution Environment 仕様を実装するデバイス上でのみ使用可能であることに注意してください。

PXE デバイスと非 PXE デバイスの両方で使用されるデフォルトのネットワークブート方式は DHCP です。Oracle Solaris をインストールするために、または復旧を目的として x86 ベースシステムのネットワークブートを実行するには、PXE クライアント用に構成された DHCP サーバーが必要です。また、tftp サービスを提供するブートサーバーも必要です。PXE または DHCP サーバーを使用できない場合は、GRUB をフロッピーディスク、CD-ROM、またはローカルディスクから読み込むことができます。

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。DHCP についての詳細は、『[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。

DHCP サーバーは、DHCP クラス PXEClient および GRUBClient に次の情報を返せるように設定されている必要があります。

- ファイルサーバーの IP アドレス
- ブートファイルの名前 (pxegrub)

Oracle Solaris OS の PXE ネットワークブートは、次の順序で実行されます。

1. BIOS が、ネットワークインタフェースからブートするように構成されます。
2. BIOS が DHCP 要求を送信します。
3. DHCP サーバーが、サーバーのアドレスとブートファイルの名前を返します。
4. BIOS が、tftp を使用して pxegrub をダウンロードし、pxegrub を実行します。
5. tftp を使用して GRUB メニューファイルがダウンロードされます。
このファイルによって、利用可能なブートメニューエントリが表示されます。
6. メニューエントリを選択すると、システムは Oracle Solaris の読み込みを開始します。

▼ x86 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。
53 ページの「[x86 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 x86 ベースシステムをネットワークからブートする場合は、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する必要があります。前提条件およびさらに詳細な手順については、『[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 システムをリブートします。
 - 2 ネットワークからブートするように BIOS に指示します。
 - 特定のキーストロークシーケンスを使用してネットワークブートするように設定されている場合は、BIOS 画面が表示されたときにそのキーストロークを入力します。
 - ネットワークブートのために BIOS 設定を手動で変更する必要がある場合は、BIOS 設定ユーティリティにアクセスするためのキーストロークシーケンスを入力します。その画面で、ネットワークブートのブート優先順位を変更します。
 - 3 GRUB メニューが表示されたら、インストールするネットワークインストールイメージを選択します。

x86 ベースシステム上のブートパラメータの変更(タスク)

この章では、x86 ベースシステム上のブートパラメータの変更に関する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。

- 55 ページの「x86 ベースシステム上のブートパラメータの変更(タスクマップ)」
- 57 ページの「x86 ベースシステム上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 x86 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/E19694-01/E21741-02/index.html> 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。

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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。

SPARC ベースシステム上のブートパラメータの変更については、『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』の第 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)」を参照してください。

x86 ベースシステム上のブートパラメータの変更(タスクマップ)

表 6-1 x86 ベースシステム上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
x86 ベースシステム上のデフォルトのブートパラメータを表示します。	eeeprom コマンドの適切なパラメータを指定してその値を表示します。	58 ページの「 eeeprom コマンドを使用したブートパラメータの表示および設定 」

表 6-1 x86 ベースシステム上のブートパラメータの変更:タスクマップ (続き)

タスク	説明	参照先
EEPROM コマンドを使用して、x86 ベースシステムのブートパラメータを変更します。	EEPROM コマンドを使用して、x86 ベースシステムのブートパラメータを変更します。EEPROM コマンドを使って設定するブートパラメータは、ブート時に GRUB メニューを編集してオプションを上書きしないかぎり、システムをリブートしても持続します。	58 ページの「EEPROM コマンドを使用してブートパラメータを変更する方法」
ブート時に x86 ベースシステム上のブートパラメータを変更します。	ブート時に GRUB メニューを編集することによって、ブートパラメータを変更します。ブート時に GRUB メニューを編集することによって指定されるブートオプションは、次にシステムをブートするまでしか持続しません。	61 ページの「x86: ブート時にブートパラメータを変更する方法」
ブート時に x86 ベースシステム上のコンソールパラメータを構成します。	Oracle Solaris リリースは、x86 ベースシステム上で、以前の VGA (Video Graphics Array) 640x480 16 色コンソールに比べて高い解像度と発色数をサポートしています。コンソール設定を変更するには、ブート時に適切なコマンド行 <code>-B console=val</code> パラメータを指定します。	例 6-2 および 例 6-3
デフォルトのシャットダウンアニメーションの動作を無効にします。	シャットダウン時に進捗ステータスインジケータが表示されないようにするには、 <code>svc:/system/boot-config:SMF</code> サービスの新しい <code>splash-shutdown</code> プロパティを <code>false</code> に設定します。	63 ページの「シャットダウンアニメーションの無効化」
<code>menu.lst</code> ファイルを編集することによって、x86 ベースシステム上のブートパラメータを変更します。	新しい OS エントリを追加したり、コンソールをリダイレクトしたりするには、 <code>menu.lst</code> 構成ファイルを編集することによってブートパラメータを変更します。ファイルに加えた変更は、システムをリブートしても持続します。	64 ページの「 <code>menu.lst</code> ファイルの編集によるブートエントリとパラメータの変更」

表 6-1 x86 ベースシステム上のブートパラメータの変更: タスクマップ (続き)

タスク	説明	参照先
Oracle Solaris をインストールしたあと、menu.lst ファイルに Linux エントリを追加します。	最初にあるパーティションに Linux をインストールしてから、あとで別のパーティションに Oracle Solaris をインストールする場合、新しいインストールの GRUB メニュー情報によって以前のインストールの GRUB メニュー情報が消去されないようにするには、特別な手順に従う必要があります。	65 ページの「Oracle Solaris のインストール後に Linux エントリを GRUB メニューに追加する方法」
アクティブな GRUB メニューを探して、メニューエントリを一覧表示します。	bootadm コマンドを使用して、アクティブな GRUB メニューの場所を表示し、メニューエントリを表示します。	67 ページの「アクティブな GRUB メニューを検出し、現在のメニューエントリを一覧表示する方法」
アクティブな GRUB メニューのデフォルトのエントリを設定します。	bootadm コマンドを使用して、システム上のアクティブな GRUB メニューのデフォルトのエントリを設定します。	67 ページの「アクティブな GRUB メニューのデフォルトのブートエントリを設定する方法」

x86 ベースシステム上のブートパラメータの変更

x86 ベースシステム上のブートパラメータを変更するための主な方法は次のとおりです。

- eeprom コマンドを使用する方法

eeprom コマンドを使用して、一連の標準パラメータに別の値を割り当てます。これらの値は、SPARC の OpenBoot PROM NVRAM 変数と等価で、/boot/solaris/bootenv.rc ファイルまたは menu.lst ファイルに格納されます。eeprom コマンドを使用してブートパラメータに加えた変更は、システムをリブートしても持続し、ソフトウェアのアップグレード中も保持されます。詳細は、[eeprom\(1M\)](#) のマニュアルページを参照してください。



注意—menu.lst ファイルを直接編集する場合、特定のブートパラメータ (boot-file、boot-arguments、および console) を、eeprom コマンドを使用してあとから変更することはできません。

- ブート時に GRUB メニューを編集する方法

ブート時にデフォルトのカーネル使用法を変更して加えた変更は、`eeeprom` コマンドを使用して設定するオプションより優先されます。ただし、これらの変更の効果は、次にシステムをブートするまでしか持続しません。詳細は、[kernel\(1M\)](#) のマニュアルページを参照してください。

- GRUB 構成ファイルを編集する

eeeprom コマンドを使用したブートパラメータの表示および設定

特定のブートパラメータのデフォルト値を確認するには、次のように `eeeprom` コマンドを使用します。

```
$ eeeprom parameter
```

たとえば、`boot-device` パラメータのデフォルト値を表示するには、次のように入力します。

```
$ eeeprom boot-device
```

注- ブートパラメータを表示するために `root` ユーザーである必要はありません。ただし、`eeeprom` コマンドを使用してブートパラメータやその他の任意のパラメータを変更するには、`root` ユーザーになる必要があります。

▼ eeeprom コマンドを使用してブートパラメータを変更する方法

- 1 `root` の役割になります。
- 2 指定されたパラメータを変更します。
`# eeeprom parameter=new-value`
- 3 新しいパラメータが設定されていることを確認します。

```
# eeeprom parameter
```

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

例 6-1 eeeprom コマンドを使用して auto-boot パラメータを設定する

次の例では、`auto-boot` ブートパラメータを `true` に設定する方法を示します。

```
# eeeprom auto-boot?=true
```

eeeprom コマンドをユーザーモードで実行するときは、シェルが疑問符を解釈しないようにするために、末尾に疑問符 (?) の付いたすべてのパラメータを二重引用符で囲む必要があります。疑問符の前にエスケープ文字 (\) を付けても、シェルが疑問符を解釈しなくなります。例:

```
# eeeprom "auto-boot?"=true
```

ブート時のブートパラメータの変更

ブート時にシステムのブート動作を変更するには (たとえば、システムをシングルユーザー状態にしたり、カーネルデバッグを有効にしたりした状態でブートする場合など)、GRUB メニューが表示されたときに **e** を入力することによってブートプロセスに割り込み、GRUB メニューでブートエントリを編集します。

次の一覧では、ブート時に GRUB メニューを編集して指定できるブート引数とオプションについて説明します。

unix	ブートするカーネルを指定します。
-a	構成情報の入力を求めるプロンプトを表示します。
-s	システムをシングルユーザー状態にブートします。
-r	再構成ブートを指定します。
	接続されているすべてのハードウェアデバイスを検索してから、実際に見つかったデバイスだけに、ファイルシステムのノードを割り当てます。
-v	詳細メッセージを有効にした状態でシステムをブートします。
-x	クラスタモードにあるシステムをブートしません。
-k	カーネルデバッグを有効にした状態でシステムをブートします
-m <i>smf-options</i>	サービス管理機能 (Service Management Facility、SMF) のブート動作を制御します
	復元オプションとメッセージオプションという、2 種類のオプションがあります
-i <i>altinit</i>	代替実行可能ファイルを原始プロセスとして指定します。altinit は実行可能ファイルへの有効なパスです。
-B <i>prop=value[,prop=value]...</i>	カーネルのブートパラメータを指定します。

-B *prop=val* オプションを使用して、GRUB メニューでブートパラメータを変更できる各種の方法を次に示します。

-B *acpi-enum=off*

デバイスの ACPI (Advanced Configuration and Power Interface) 列挙を無効にします。

-B *acpi-user-options=0x2*

ACPI を完全に使用不可にします。

-B *console=force-text*

ブートに VGA テキストモードを使用することを指定します。[61 ページの「ビットマップコンソールのサポート」](#)を参照してください。

-B *console=graphics*

コンソールでブートにグラフィックスモードを使用することを指定します。これにより、高解像度の状態が可能になります。

-B *console=text*

コンソールでブートにテキストモードを使用することを指定します。これにより、高解像度の状態が可能になります。

-B *screen-#columns= value, screen-#rows= value*

フレームバッファコンソールの行数と列数を指定します。選択した行数または列数にもっとも適したフォントがシステムによって自動的に検出されます。このオプションは、フレームバッファコンソールのサイズを最適化するために使用されます。[61 ページの「ビットマップコンソールのサポート」](#)を参照してください。

-B *console=ttya*

出力先をコンソールから *ttya* に変更します。

-B *console=ttya,acpi-enum=off*

出力先をコンソールから *ttya* に変更し、デバイスの ACPI 列挙を使用不可にします。

注 - *eeprom* コマンドを使用し、かつ GRUB コマンド行でパラメータが指定された場合は、GRUB コマンド行が優先されます。

▼ x86: ブート時にブートパラメータを変更する方法

ブート時に GRUB メニューを編集してデフォルトのカーネル使用法を変更すると、その変更はシステムをリブートするまでしか持続しません。次にシステムをブートすると、デフォルトのブートパラメータが復元されます。

- 1 システムをリブートします。
ブートシーケンスが始まると、GRUB メインメニューが表示されます。
- 2 矢印キーを使用して、編集するブートエントリを選択します。
- 3 **e** と入力して、**GRUB 編集メニュー**にアクセスします。
- 4 メニューの **kernel\$** 行を選択します。
- 5 **e** と入力して、ブート引数をその行に追加します。
- 6 指定する追加のブート引数を入力します。
- 7 **Return** キーを押して変更を保存し、前のメニューに戻ります。

注 - 変更を保存せずに GRUB メインメニューに戻るには、**Escape** キーを押します。

- 8 システムをブートするために、**b** と入力します。
適用した変更は、システムをブートしたあとで有効になります。

ビットマップコンソールのサポート

Oracle Solaris 11 は、x86 ベースシステム上で、以前の VGA (Video Graphics Array) 640×480 16 色コンソールに比べて高い解像度と発色数をサポートしています。このサポートは、従来の BIOS および VESA (Video Electronics Standards Association) オプション読み取り専用メモリー (ROM) を使用するシステムを対象に提供されます。サポートは、物理または仮想コンソールとしてグラフィックカードまたはフレームバッファが使用されるときに限られることに注意してください。シリアルコンソールの動作への影響はありません。

この機能をサポートするために、次の 2 つのコマンド行 **-B option=val** パラメータを使用できます。

-B console=force-text

ブートに VGA テキストモードを使用することを指定します。

`-B screen-#columns= value, screen-#rows= value`

フレームバッファコンソールの行数と列数を指定します。選択した行数または列数にもっとも適したフォントがシステムによって自動的に検出されます。このオプションは、フレームバッファコンソールのサイズを最適化するために使用されます。

GRUB では、デフォルトで、インストールされているビデオカードとモニターで動作する解像度と発色数が検出されます。ただし、別の解像度 (たとえば、より高い解像度) や別の発色数を指定できます。

GRUB は、ビデオモードを指定するための次の 2 つの方法をサポートしています。

`vbset hexmode`

目的の VESA モードの 16 進数コードを指定します。カードとモニターでサポートされているすべてのモードの一覧を取得するには、GRUB コマンドプロンプトで `vbeprobe` コマンドを使用します。これにより、次のような一覧が表示されます。

```
0x117: Direct Color, 1024x768x16
0x118: Direct Color, 1024x768x32
0x11a: Direct Color, 1280x1024x16
0x11b: Direct Color, 1280x1024x32
[...]
```

1024x768x32 の構成を指定する `vbset` エントリは、次のように示されます。

```
vbset 0x118
```

`vbset` エントリは、GRUB メニューで `kernel$` および `module$` エントリのあとに指定する必要があります。

`vbematch xres yres depth`

指定された構成 (たとえば、1024x768x32) を検索するよう GRUB に指示します。見つかった場合、GRUB は指定された構成を設定します。

`vbset` エントリの代わりに使用した場合、1024x768x32 の構成のための `vbematch` エントリは次のようになります。

```
vbematch 1024 768 32
```

`vbematch` エントリは、GRUB メニューで `kernel$` および `module$` エントリのあとに指定する必要があります。

例 6-2 x86: コンソールのテキストモードブートパラメータの構成

テキストモードでは、コンソール出力はフレームバッファーに送信され、入力はキーボードから受信されます。テキストモードのバリエーションであるグラフィックスモードでは、キーが押されるか、またはコンソールの `login`、`sulogin`、または `kmdb` コマンドによってコンソールの対話が必要になるまで、アニメーションを含むイメージが表示されます。テキストの新しいプロパティ `force-text` は、VGA アダプタをビットマップデバイスとして使用しないようシステムに指示し、そのアダプタを VGA テキストモードに設定します。

このプロパティが存在しない場合、コンソールデバイスは、`input-device` と `output-device` のプロパティペアで指定されたデバイスに戻ります。コンソールのプロパティと、`input-device` と `output-device` のプロパティペアのどちらも存在しない場合、コンソールはデフォルトでフレームバッファーとキーボードになります。

次の例は、ブート時にカーネルコマンド行で `-B console=force-text` プロパティを指定する方法を示しています。

-B console=force-text

例 6-3 x86: グラフィカル表示の有効化およびコンソールテキストモードのパラメータの構成

デフォルトでは、コンソールテキストモードは 80 列×24 行です。このパラメータを再構成するには、`-B` オプションと `screen-#columns=value` および `screen-#rows=value` パラメータを使用します。

たとえば、グラフィカル表示を有効にして 100 列×60 行のコンソール端末を割り当てるには、カーネルコマンド行で次のパラメータを指定できます。

-B console=graphics, screen-#columns=100,screen-#rows=60

シャットダウンアニメーションの無効化

システムをブートするために `console=graphics` オプションが使用され、かつ Xorg サーバーによってシャットダウンがトリガーされた場合は、シャットダウンプロセス中に進捗ステータスインジケータが表示されます。進捗ステータスインジケータが表示されないようにするには、次のように `svc:/system/boot-config SMF` サービスの新しい `splash-shutdown` プロパティを `false` に設定します。

```
# svccfg -s svc:/system/boot-config:default setprop config/splash_shutdown = false
# svcadm refresh svc:/system/boot-config:default
```

menu.lst ファイルの編集によるブートエントリとパラメータの変更

GRUB メニューは、menu.lst 構成ファイル内のエントリに基づいており、カスタマイズ可能です。Solaris は、Oracle Solaris ブート環境 (BE) の GRUB menu.lst エントリを自動的に管理します。ブート環境が (パッケージシステムによって、または beadm コマンドを使用して明示的に) 作成されると、menu.lst ファイルに GRUB エントリが追加されます。beadm destroy コマンドを destroy サブコマンドとともに使用してブート環境を削除すると、対応するエントリが GRUB menu.lst ファイルから削除されます。Oracle Solaris は、システム上にインストールされているほかのオペレーティングシステムの menu.lst エントリを自動的に追加することはありません。これらのオペレーティングシステムのメニューエントリは手動で追加する必要があります。詳細は、[65 ページの「Oracle Solaris のインストール後に Linux エントリを GRUB メニューに追加する方法」](#)を参照してください。

Oracle Solaris 11 の標準的な menu.lst ファイルには、次の情報が含まれている可能性があります。

```
#----- ADDED BY BOOTADM - DO NOT EDIT -----
title Oracle Solaris 11      1
findroot (pool_rpool,0,a)    2
bootfs rpool/ROOT/solaris    3
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS,console=graphics  4
module$ /platform/i86pc/$ISADIR/boot_archive      5
#-----END BOOTADM-----
```

1. メニューエントリのオペレーティングシステムのタイトルを指定します。
2. SIGNATURE ファイル名を見つけるためにすべてのパーティションを検索します。この例では、ファイル名は pool_rpool です。

GRUB は、/boot/grub/bootsign ディレクトリのみでファイル名を検索し、ファイルの最初のインスタンスが見つかるかすぐに停止します。有効にするために、署名ファイルの名前はすべてのパーティションにわたって一意である必要があります。署名ファイルを見つけたあと、GRUB はそのパーティション上の root コマンドを呼び出します。検索を最適化するために、省略可能なパーティションとスライスを指定できます。

3. 現在の ZFS ブートファイルシステムを指定された値に設定します。menu.lst ファイルのこの例では、このプロパティは現在の ZFS ブートファイルシステムを rpool/ROOT/solaris に設定します。
4. 指定されたパスからプライマリブートイメージを読み込みます。この行の残りは、そのままカーネルコマンド行として渡されます。ドル記号 (\$) は、\$ISADIR エントリの拡張です。
5. 指定されたカーネルのブートアーカイブを読み込みます。ドル記号 (\$) は、\$ISADIR エントリの拡張です。

注- 特定の GRUB コマンドの詳細情報を表示するには、コマンド行から `help` コマンドを入力します。

デフォルトのオペレーティングシステムをブートする場合は、構成可能なタイムアウトを使用できます。ブートされるデフォルトのエントリは、`default` コマンドを使用して構成できます。インストールソフトウェアは通常、いずれかの有効なブートエントリをブートするようにこのコマンドを設定します。別のバージョンの Oracle Solaris (存在する場合) をブートするか、または Oracle Solaris のほかに別のオペレーティングシステムをブートするには、矢印キーを使用してそのブートエントリを選択したあと、`Enter` キーを押してそのオペレーティングシステムをブートします。`default` コマンドが設定されていない場合は、GRUB メニューにある最初のブートエントリがブートされることに注意してください。

システムのブートに使用されるのは、アクティブな `menu.lst` ファイルだけです。システムのブート時に表示される GRUB メニューを変更するには、アクティブな GRUB `menu.lst` ファイルを編集します。ほかのどの `menu.lst` ファイルを変更しても、システムをブートしたときに表示されるメニューに影響はありません。アクティブな `menu.lst` ファイルの場所を確認するには、`bootadm` コマンドの `list-menu` サブコマンドを使用します。

▼ Oracle Solaris のインストール後に Linux エントリを GRUB メニューに追加する方法

あるパーティションに Linux を先にインストールしてから別のパーティションに Oracle Solaris をインストールするといった方法でブート環境を設定する場合、新規インストールの GRUB メニュー情報によって以前のインストールの GRUB メニュー情報が消去されないようにするには、次に示す特別な手順に従う必要があります。次の手順に、すでにインストールされている Linux のエントリを含めるように `menu.lst` ファイルを手動で更新する方法を示します。この手順では、システムに Linux がすでにインストールされていて、そのあとに Oracle Solaris がインストールされたことを前提にしています。

- 1 **Linux** のインストールの完了時に、アクティブな `menu.lst` ファイルを USB ドライブにコピーして、**Oracle Solaris** のインストールを完了したあとにそれらの情報を再利用できるようにします。

通常、このファイルは `/boot/grub/menu.lst` です。

- アクティブな `menu.lst` ファイルの場所がわからない場合は、`bootadm` コマンドを使用してファイルを見つけます。

```
# bootadm list-menu
```

- USBドライブの場所がわからない場合は、オプションを指定せずに **mount** コマンドを使用し、USBドライブがマウントされている場所を特定します。次に、その場所に **menu.lst** ファイルをコピーします。
- 2 インストールの完了後に、次の方法でアクティブな **menu.lst** ファイルを編集します。

- a. 端末ウィンドウを開き、**root** の役割になります。

```
$ su root
Password:
```

- b. テキストエディタを使用して、**menu.lst** ファイルを編集します。

例:

```
# vi /pool-name/boot/grub/menu.lst
```

ここで、*pool-name* は ZFS ストレージプールの名前です。

- c. 手順1で **menu.lst** ファイルをコピーした USB ドライブを使用して、元の Linux インストールの **Linux menu.lst** 情報を新しい **menu.lst** ファイルの末尾にコピーします。

たとえば、Ubuntu インストールの **menu.lst** ファイルは次のようになります。

```
title          Ubuntu 8.04, kernel 2.6.24-18-generic
  root          (hd0,4)
  kernel        /vmlinuz-2.6.24-18-generic \
root=UUID=1ed7fa17-6d77-4b49-be1a-22481310fd1b ro quiet splash
  initrd        /initrd.img-2.6.24-18-generic
  quiet

  title          Ubuntu 8.04, kernel 2.6.24-18-generic (recovery mode)
  root          (hd0,4)
  kernel        /vmlinuz-2.6.24-18-generic \
root=UUID=1ed7fa17-6d77-4b49-be1a-22481310fd1b ro single
  initrd        /initrd.img-2.6.24-18-generic
```



注意 - **menu.lst** ファイルの元の内容を直接編集しないでください。新しい情報は必ずファイルの末尾に追加します。また、ファイルの内容を変更する場合には、既存の内容を複製してから変更するようにしてください。

- d. ファイルを保存して終了します。

- 3 システムをリブートします。

システムがリブートすると、Linux と Oracle Solaris の両方のオペレーティングシステムのエントリが GRUB メニューに含まれているはずです。

bootadm コマンドを使用したブートエントリのパラメータの表示および設定

▼ アクティブな **GRUB** メニューを検出し、現在のメニューエントリを一覧表示する方法

次の手順に従って、アクティブな GRUB メニューの場所を調べ、現在の GRUB メニューエントリを一覧表示します。

- 1 **root** の役割になります。
- 2 アクティブな **GRUB** メニューの場所と現在の **GRUB** メニューエントリを一覧表示するには、次のように入力します。

```
# bootadm list-menu
```

```
list-menu      アクティブな GRUB メニューの場所と現在の GRUB メニューエントリ
                を一覧表示します。この出力には、autoboot-timeout デフォルトエン
                トリ番号に関する情報と、各エントリのタイトルが含まれます。
```

例 6-4 アクティブな GRUB メニューの場所と現在の GRUB メニューエントリを一覧表示する

```
# bootadm list-menu
The location for the active GRUB menu is: /stubboot/boot/grub/menu.lst
default=0
timeout=30
0 2010-12-10-be
1 Oracle Solaris 11
2 Linux
```

▼ アクティブな **GRUB** メニューのデフォルトのブートエントリを設定する方法

- 1 **root** の役割になります。
- 2 アクティブな **GRUB** メニューのデフォルトのブートエントリを設定するには、次のように入力します。

```
# bootadm set-menu menu-entry
```

```
set-menu      GRUB メニューを保守します。アクティブな GRUB メニューの場所は
                boot/grub/menu.lst です。
```

```
menu-entry    デフォルトとして設定する GRUB メニューエントリを指定します。
```

- 3 デフォルトのメニューエントリが変更されたことを確認するには、次のように入力します。

```
# bootadm list-menu
```

新しいデフォルトのメニューエントリが表示されるはずです。

例 6-5 GRUB のデフォルトのメニューエントリを切り替える

次の例は、デフォルトの GRUB メニューを、前の例に表示されているメニューエントリのいずれかに切り替える方法を示しています。選択されたメニューエントリは、Linux (メニューエントリ 2) です。

```
# bootadm set-menu default=2
```

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)

この章では、x86 ベースシステム上で ZFS ブート環境 (BEとも呼ばれる) 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。

- 70 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)」
- 71 ページの「ブート環境の作成と管理」
- 76 ページの「x86 プラットフォーム上の ZFS ブート環境またはルートファイルシステムからのブート」

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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。

SPARC プラットフォーム上での ZFS ブート環境からのブートについては、『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』の第 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)」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
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。	71 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。	73 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。	73 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。	73 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。	74 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」
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。	75 ページの「ブート環境を破棄する方法」
x86 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。	ZFS ブートローダーをサポートする Oracle Solaris リリースをシステムにインストールするか、またはシステムをそのリリースにアップグレードした場合は、デフォルトの ZFS ブート環境の GRUB メニューエントリに <code>-B \$ZFS-BOOTFS</code> ブート引数が含まれます。したがって、そのシステムは自動的に ZFS ルートからブートします。 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。	76 ページの「x86 プラットフォーム上の ZFS ブート環境またはルートファイルシステムからのブート」

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` ユーティリティについての詳細は、[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` ユーティリティ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、[73 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、既存のアクティブでない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 x86 GRUB メニューのデフォルトエントリになります。

注- ブート環境がブートに失敗した場合は、GRUB メニューまたはブートメニューから以前のブート環境をリブートして選択します。

▼ 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 システム上で使用可能なすべてのデータセットを一覧表示するには、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。このオプションでは、下位のスナップショットとデータセットが含まれます。
- 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。

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` コマンドでは、破棄されたブート環境のエントリが x86 GRUB 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    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。
```

x86 プラットフォーム上の **ZFS** ブート環境またはルートファイルシステムからのブート

インストール処理中または `beadm activate` 操作中に、ZFS を自動的にブートするための次のエントリが `/pool-name/boot/grub/menu.lst` ファイルに追加されます。

```
title 2010-12-10-be-s
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/2010-12-10-be_152
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS -s
module$ /platform/i86pc/$ISADIR/boot_archive
```

GRUB によってブートデバイスとして識別されたデバイスに ZFS ストレージプールが含まれている場合、`menu.lst` ファイルを使用して GRUB メニューが作成されます。複数の ZFS ブート環境がある x86 ベースのシステムでは、ブート中に GRUB メニューからブート環境を選択できます。このメニューエントリに対応するルートファイルシステムが ZFS データセットである場合は、次のオプションが追加されます:

```
-B $ZFS-BOOTFS
```

\$ZFS-BOOTFS キーワードにより、x86 ベースのシステムで Oracle Solaris ZFS ルートファイルシステムからのブートを実行できます。このオプションは、どのブート環境またはデータセットをブートするかを識別します。ZFS ブートローダーをサポートする Oracle Solaris リリースをインストールする場合は、GRUB `menu.lst` ファイルおよび GRUB ブートメニューにデフォルトでこの情報が含まれます。

例 7-3 ZFS ブート環境、データセット、またはファイルシステムからのブート

ZFS ファイルシステムからブートする場合、ルートデバイスは、GRUB メニューの `kernel$` 行にある `-B $ZFS-BOOTFS` ブートパラメータで指定されます。この値は、`-B` オプションで指定されるすべてのパラメータと同様に、GRUB からカーネルに渡されます。例:

```
title Oracle Solaris 11 Express snv_152
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/solaris
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/$ISADIR/boot_archive
```


x86 ベースシステムのブート可能状態の維持(タスク)

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、x86 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。

- 79 ページの「x86 ベースシステムのブート可能状態の維持(タスクマップ)」
- 80 ページの「Oracle Solaris ブートアーカイブの説明」
- 81 ページの「ブートアーカイブ SMF サービスの管理」
- 82 ページの「ブートアーカイブの整合性の維持」

x86 ベースシステムのブートに関する概略の情報は、第 1 章「x86 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。

SPARC ベースシステムのブート可能状態の維持については、『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』の第 8 章「SPARC ベースシステムのブート可能状態の維持(手順)」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(タスクマップ)

表 8-1 x86 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。	81 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 x86 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。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。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。	81 ページの「ブートアーカイブ SMF サービスの管理」
auto-reboot-safe プロパティを使用して、x86 ベースのシステムでブートアーカイブ更新障害を消去します。	auto-reboot-safe プロパティが false に設定されているために、x86 ベースのシステムでブートアーカイブ更新が失敗した場合に、この手順を使用します。	83 ページの「auto-reboot-safe プロパティを使用してブートアーカイブ自動更新障害を消去する方法」
bootadm コマンドを使用して、x86 ベースシステム上でのブートアーカイブ更新障害を解決します。	この手順は、auto-reboot-safe プロパティが有効になっている場合に、x86 ベースシステム上でのブートアーカイブ更新障害を手動で解決するために使用します。	83 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。

x86 ブートアーカイブの場所と内容に関する情報の取得

x86 ブートアーカイブ内のファイルは、`/platform/i86pc/amd64/boot_archive` 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るたびに、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
```

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を確認する

`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         10:35:14      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 管理者になります。

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については、[82 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、Oracle Solaris 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。
- 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。

▼ **auto-reboot-safe** プロパティを使用してブートアーカイブ自動更新障害を消去する方法

x86 プラットフォームでのブートアーカイブ回復は、高速リブート機能によって自動化されます。ただし、システムのブートプロセス中に、次のような警告が表示される場合があります:

```
WARNING: Reboot required.
The system has updated the cache of files (boot archive) that is used
during the early boot sequence. To avoid booting and running the system
with the previously out-of-sync version of these files, reboot the
system from the same device that was previously booted.
```

ここで、システムは保守モードに入ります。結果として、ブートアーカイブの自動更新は失敗します。問題を修正するには、ここで説明する手順に従います。

- 1 **root** の役割になります。
- 2 システムをリブートします。
reboot
- 3 アクティブな **BIOS** ブートデバイスと **GRUB** メニューエントリが現在のブートインスタンスを指している場合は、ブートアーカイブ更新障害を防ぐために、次の手順に従います:

- a. 次のように入力して、**svc:/system/boot-config:SMF** サービスの **auto-reboot-safe** プロパティを **true** に設定します。

```
# svccfg -s svc:/system/boot-config:default setprop config/auto-reboot-safe = true
```

- b. **auto-reboot-safe** プロパティの設定が正しいことを確認します。

```
# svccfg -s svc:/system/boot-config:default listprop |grep config/auto-reboot-safe
config/auto-reboot-safe          boolean true
```

▼ 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た場合は、その結果、ブートアーカイブの自動更新が失敗します。

```
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- 同じ手順を使用して、x86 ベースシステム上のブートアーカイブを手動で更新できます。

- 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- 代替ルートのブートアーカイブを更新するには、次のように入力します。

```
# 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
```

x86 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)

x86 ベースシステムのブートをトラブルシューティングするための手順を次に示します。

この章の内容は次のとおりです。

- 86 ページの「x86 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)」
- 87 ページの「復旧を目的とした x86 ベースシステムのシャットダウンおよびブート」
- 94 ページの「x86 プラットフォーム上での高速リブートに関する問題のトラブルシューティング」

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/E19694-01/E21741-02/index.html> 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。

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、82 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 ベースシステム上でのブートのトラブルシューティングについては、『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』の第 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)」を参照してください。

x86 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 x86 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 x86 ベースシステムを停止します。	破損したファイルが原因で x86 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。	87 ページの「復旧を目的としてシステムを停止する方法」
x86 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。	88 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」
x86 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。	88 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」
x86 ベースシステムをメディアからブートして、システムのブートを妨げている menu.lst ファイルの問題を解決します。	システムをメディアからブートして、ルートプールをインポートし、menu.lst ファイルの問題を分析および修正します。	90 ページの「メディアからブートして、システムのブートを妨げている menu.lst ファイルの問題を解決する方法」
x86 ベースシステムのクラッシュダンプとリポ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、x86 ベースシステムのクラッシュダンプとリポートを強制的に実行します。	92 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) を有効にして、x86 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、カーネルデバッガを有効にして x86 ベースシステムをブートします。	93 ページの「カーネルデバッガ (kmdb) を有効にした状態でシステムをブートする方法」
ZFS ルート環境内で、復旧を目的として x86 ベースシステムをブートします。	この手順は、root パスワードの紛失や、同様の問題から回復できるようにシステムをブートする必要がある場合に使用します。	90 ページの「メディアからブートして、システムのブートを妨げている menu.lst ファイルの問題を解決する方法」
x86 ベースシステム上での高速リブート機能に関する問題をトラブルシューティングします。	x86 ベースシステムが高速リブートを開始できなくなる可能性のある問題をトラブルシューティングします。	94 ページの「x86 プラットフォーム上での高速リブートに関する問題のトラブルシューティング」

復旧を目的とした **x86** 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 x86 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。

システム復旧のための停止とブート

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を次に示します。

- 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。この方法では、ルートプールをインポートしたあとに、ブート環境をマウントする必要があります。
- ルートプールをインポートすることにより、ブート構成の問題を解決します。`menu.lst` ファイルで問題が発生した場合は、ブート環境をマウントする必要はなく、単にルートプールをインポートするだけで、ブート関連のコンポーネントを含む `rpool` ファイルシステムが自動的にマウントされます。

▼ 復旧を目的としてシステムを停止する方法

1 システムを停止します。

- 最初に `root` の役割になり、次にキーボードとマウスが機能している場合は `init 0` を入力します。
- `Press any key to reboot` プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
- システムをリブートするには、`init 6` を入力します。

- 2 マウスからの入力に対してシステムが応答しない場合は、次のいずれかを行います。
 - **Reset** キーを押してシステムをリブートします。
 - 電源スイッチを使用してシステムをリブートします。

▼ 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 システムを停止します。
`# init 0`
- 2 システムをリブートします。
`# reboot`
- 3 **GRUB** メニューが表示されたら、次を行います。
 - a. 該当するブートエントリを選択し、**e** と入力して、そのエントリを編集します。
 - b. 矢印キーを使用して、**kernel\$** 行を選択します。
矢印キーを使用できない場合は、キャレット (^) キーを使って上方へスクロールしたり、英字キー **v** を使って下方へスクロールしたりします。
 - c. **\$kernel** 行の終わりに **-s** と入力して、**Return** キーを押し、変更を保存して、前の画面に戻ります。
 - d. **b** と入力して、シングルユーザーモードでシステムをブートします。
- 4 **/etc/passwd** ファイルのシェルエントリを修正します。
`# vi /etc/password`
- 5 **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

次の手順は、不明な **root** パスワードの問題や、同様の問題を修正するためにシステムをブートする必要がある場合に使用します。この手順では、ルートプールをインポートしたあとに、ブート環境をマウントする必要があります。ルートプールまたはルートプールのスナップショットを回復する必要がある場合は、『[Oracle Solaris の管理: ZFS ファイルシステム](#)』の「[ZFS ルートプールのディスクを交換する方法](#)」を参照してください。

1 Oracle Solaris のメディアからブートします。

- **Live Media**- インストールメディアからブートし、回復手順のために **GNOME** 端末を使用します。
- テキストインストール - **GRUB** メニューから「**Text Installer and command line**」ブートエントリを選択し、次にテキストインストール画面からオプション「**3 Shell**」を選択します。
- 自動インストール-ネットワーク上のインストールサーバーからのブートには **PXE** ブートが必要です。**GRUB** メニューから「**Text Installer and command line**」エントリを選択します。次に、テキストインストール画面からオプション「**3 Shell**」を選択します。

例:

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently xterm)
5 Reboot
```

```
Please enter a number [1]: 3
To return to the main menu, exit the shell
```

2 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 /a にブート環境をマウントします

```
# beadm mount solaris-instance|bename /a
```

例:

```
# beadm mount solaris-2 /a
```

5 パスワードまたはシャドウエントリが原因でコンソールログインできない場合は、その問題を訂正します。

a. TERM タイプを設定します。

```
# TERM=vt100
# export TERM
```

b. shadow ファイルを編集します。

```
# cd /a/etc
# vi shadow
# cd /
```

- 6 ブートアーカイブを更新します。

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

- 7 ブート環境をマウント解除します。

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

- 8 システムを停止します。

```
# halt
```

- 9 シングルユーザーモードでシステムをリブートして (88 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](#)」を参照)、**root** パスワードの入力を求められたら、**Return** キーを押します。

- 10 **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
```

- 11 **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、システムのブートを妨げている **menu.lst** ファイルの問題を解決する方法

次の手順は、デフォルトの **menu.lst** ファイルの問題を修正するためにシステムをブートする必要がある場合に使用します。この手順では、ブート環境をマウントする必要はありません。ルートプールまたはルートプールのスナップショットを回復する必要がある場合は、『[Oracle Solaris の管理: ZFS ファイルシステム](#)』の「[ZFS ルートプールのディスクを交換する方法](#)」を参照してください。

- 1 **Oracle Solaris** のメディアからブートします。

- **Live Media**-インストールメディアからブートし、回復手順のために **GNOME** 端末を使用します。
- テキストインストール - **GRUB** メニューから「**Text Installer and command line**」ブートエントリを選択し、次にテキストインストール画面からオプション「**3 Shell**」を選択します。

- 自動インストール-ネットワーク上のインストールサーバーからのブートには **PXE** ブートが必要です。**GRUB** メニューから「**Text Installer and command line**」エントリを選択します。次に、テキストインストール画面からオプション「**3 Shell**」を選択します。

例:

```
1  Install Oracle Solaris
2  Install Additional Drivers
3  Shell
4  Terminal type (currently xterm)
5  Reboot
```

```
Please enter a number [1]: 3
To return to the main menu, exit the shell
```

- 2 ルートプールをインポートします。

```
zpool import -f rpool
```

- 3 **menu.lst** ファイル内のエントリを検査し、必要に応じて修正します。

```
# cd /rpool/boot/grub
# vi menu.lst
```

- 4 ブートアーカイブを更新します。

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

- 5 シェルを終了してシステムをリブートします。

```
exit
1  Install Oracle Solaris
2  Install Additional Drivers
3  Shell
4  Terminal type (currently sun-color)
5  Reboot
```

```
Please enter a number [1]: 5
```

クラッシュダンプを強制してシステムをリブートする

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の詳細は、『[Oracle Solaris の管理: 一般的なタスク](#)』の「[システムクラッシュダンプ情報の管理](#)」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

reboot -d コマンドまたは halt -d コマンドを使用できない場合は、カーネルデバッグ kldb を使用してクラッシュダンプを強制的に実行できます。次の手順を実行するには、ブート時に、または mdb - k コマンドを使用してカーネルデバッグが読み込まれている必要があります。

注-カーネルデバッグ (kldb) にアクセスするにはテキストモードでなければなりません。したがって、まずウィンドウシステムを終了してください。

- 1 カーネルデバッグにアクセスします。
デバッグにアクセスするための方法は、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。
 - ローカル接続されているキーボードを使用している場合は、F1-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でブレイクを送信します。

kldb プロンプトが表示されます。

- 2 クラッシュを強制的に実行するには、**systemdump** マクロを使用します。

```
[0]> $<systemdump
```

パニックメッセージが表示され、クラッシュダンプが保存され、システムがリブートします。

- 3 コンソールログインプロンプトでログインして、システムがリブートされていることを確認します。

例 9-1 x86: 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 および boot コマンドを使用して、x86 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する方法を示しています。

```
# halt -d
4ay 30 15:35:15 wacked.<domain>.COM halt: halted by user

panic[cpu0]/thread=ffffffff83246ec0: forced crash dump initiated at user request

fffffe80006bbd60 genunix:kadmin+4c1 ()
fffffe80006bbec0 genunix:uadmin+93 ()
fffffe80006bbf10 unix:sys_syscall32+101 ()

syncing file systems... done
dumping to /dev/dsk/clt0d0s1, offset 107675648, content: kernel
NOTICE: adpu320: bus reset
```

```
100% done: 38438 pages dumped, compression ratio 4.29, dump succeeded
```

```
Welcome to kmdb
Loaded modules: [ audiosup crypto ufs unix krtld s1394 sppp nca uhci lofs
genunix ip usba specfs nfs md random sctp ]
[0]>
kmdb: Do you really want to reboot? (y/n) y
```

▼ カーネルデバッグ (kmdb) を有効にした状態で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) を読み込むための基本的な操作を示します。デフォルトでは、savecore 機能を使用できます。

- 1 システムをブートします。
システムがブートすると、GRUB メニューが表示されます。
- 2 GRUB メニューが表示されたら、**e** と入力して **GRUB 編集メニュー** にアクセスします。
- 3 矢印キーを使用して、**kernel\$** 行を選択します。
矢印キーを使用できない場合は、キャレット (^) キーを使って上方へスクロールしたり、英字キー **v** を使って下方へスクロールしたりします。
- 4 **e** と入力して、その行を編集します。
ブートエントリメニューが表示されます。このメニューで、**kernel\$** 行の末尾に追加のブート引数を指定して、ブート動作を変更できます。
- 5 GRUB 編集メニューで、**kernel\$** 行の末尾に **-kmdb** または **-k** を入力します。

```
grub edit> kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS -s -k
```
- 6 **Enter** キーを押して 1 つ前の画面に戻ってから、**b** を入力して、カーネルデバッグを有効にした状態でシステムをブートします。
-kmdb または **-k** を入力すると、デバッグが読み込まれたあと、オペレーティングシステムが直接ブートされます。
- 7 カーネルデバッグにアクセスします。
デバッグにアクセスするための方法は、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。
 - ローカル接続されているキーボードを使用している場合は、**F1-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、そのタイプのシリアルコンソールに適した方法を使用してブレイクを送信します。

システムが完全にブートする前にカーネルデバッグにアクセスするには、`-kd` オプションを使用します。

`-kd` オプションを使用すると、デバッグが読み込まれたあと、オペレーティングシステムをブートする前にデバッグと対話する機会が与えられます。

カーネルデバッグにはじめてアクセスした場合、ウェルカムメッセージが表示されます。

参照 `kldb` と、`kldb` によって提供される実行制御機能を使用してシステムと対話する方法についての詳細は、[kldb\(1\)](#) のマニュアルページを参照してください。

x86 プラットフォーム上での高速リブートに関する問題のトラブルシューティング

以降のセクションでは、x86 プラットフォーム上で Oracle Solaris の高速リブート機能に関して発生する可能性のあるいくつかの一般的な問題を特定および解決する方法について説明します。

発生する可能性のある早期システムパニックのデバッグ

`boot-config` サービスはマルチユーザーマイルストーンに依存しているため、早期システムパニックをデバッグする必要のあるユーザーは、次の例に示すように `/etc/system` ファイル内の大域変数 `fastreboot_onpanic` にパッチを適用できます。

```
# echo "set fastreboot_onpanic=1" >> /etc/system
# echo "fastreboot_onpanic/W" | mdb -kw
```

x86 プラットフォーム上で高速リブートが動作できなくなる可能性のある状態のトラブルシューティング

次のような状況では、高速リブート機能が正しく動作しない可能性があります。

- GRUB メニューを処理できない。
- ドライバに `quiesce` 関数が実装されていない。
サポートされていないドライバを使用しているシステムの高速リブートを試行すると、次のようなメッセージが表示されます。

```
Sep 18 13:19:12 too-cool genunix: WARNING: nvidia has no quiesce()
reboot: not all drivers have implemented quiesce(9E)
```

quiesce 関数をサポートしていないドライバがグラフィックドライバだけの場合は、次のコマンドを実行して、強制的に高速リブートを試行することができます。

```
# echo "force_fastreboot/W 1" | mdb -kw# echo "set force_fast \
reboot = 1" #x26;#x26;#x3e;#x26;#x26;#x3e; /etc/system
```

注-ネットワークインタフェースカード (NIC) のドライバに quiesce 関数が実装されていない場合は、まずインタフェースの unplumb を試行してから、次にシステムの高速リブートを試行します。

- メモリーが不足している。

システム上に十分なメモリーがないか、または新しいカーネルとブートアーカイブを読み込むための十分な空きメモリーがない場合は、高速リブートの試行が失敗して次のメッセージが表示されたあと、通常のリブートにフォールバックします。

```
Fastboot: Couldn't allocate size below PA 1G to do fast reboot
Fastboot: Couldn't allocate size below PA 64G to do fast reboot
```

- 環境がサポートされていない。

高速リブート機能は、次の環境ではサポートされていません。

- 準仮想化 (PV) ゲストドメインとして実行されている Oracle Solaris リリース
- 非大域ゾーン

詳細は、次のマニュアルページを参照してください。

- [reboot\(1M\)](#)
- [init\(1M\)](#)
- [quiesce\(9E\)](#)
- [uadmin\(2\)](#)
- [dev_ops\(9S\)](#)

索引

B

bootadm, ブートアーカイブを管理するために使用されるコマンド, 81-82

E

eeeprom コマンド
 ブートパラメータを設定する方法
 GRUB, 58-59
eeeprom コマンドを使用したブートパラメータの設定, GRUB ベースのブート, 58-59

G

GRUBClient, x86 ベースのネットワーク
 ブート, 52-54
GRUB デバイス命名規則, 25
GRUB のコンポーネント, 24
GRUB ベースのブート, ブート時に GRUB カーネルの使用法を変更, 61
GRUB メニュー, 説明, 64-66
GRUB メニューエントリ, Linux 情報の保持, 65
GRUB メニューでカーネルの使用法を変更, 61

H

halt -d, クラッシュダンプとリブートの強制的な実行, 92-93
halt コマンド, 36

I

init コマンド, 説明, 36
init 状態, 「実行レベル」を参照

K

kldb コマンド, 93-94
kldb の有効化, トラブルシューティング, 93-94

L

Linux メニューエントリ, menu.lst ファイルの更新, 65

M

menu.lst ファイル, GRUB のコンポーネント, 24
menu.lst ファイル
 Linux エントリの追加方法, 65
 ブート時の対話
 説明, 64-66

O

Oracle Solaris ブートアーカイブ, 整合性を維持する方法, 79-80
Oracle Solaris ブート動作, 管理方法, 55-68

P

poweroff コマンド, 36
PXECClient, x86 ベースのネットワーク
ブート, 52-54

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
stage1, GRUB のコンポーネント, 24
stage1 イメージ, 24
stage2, GRUB のコンポーネント, 24

W

who コマンド, 28

X

x86 ブートの用語, 25-26
x86 ブートパラメータの変更 (タスク
マップ), 55-57
x86 プラットフォーム上のネットワーク
ブート, 52-54
x86 ベースシステムのネットワークからの
ブート, 52-54
(作業マップ), 51-52
x86 ベースシステムのブート可能状態の維持, (タ
スクマップ), 79-80
x86 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), 86-87
x86 ベースシステムのリブート, (タスク
マップ), 43-44

Z

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), 70-71

か

カーネルデバッグ (kmdb), システムの
ブート, 93-94
確認, 実行レベル (方法), 28
管理
boot-archive サービス, 81-82
ブート動作, 55-68

く

クラッシュダンプとリブートの強制的な実行
halt -d, 92-93
トラブルシューティング, 91-93

こ

高速リブート
x86 プラットフォームでのブート方法, 47
新しくアクティブになったブート環境への開
始, 47-49
高速リブートが動作できなくなる可能性のある
状態のトラブルシューティング, 94-95
問題のトラブルシューティング, 94-95
高速リブートが動作できなくなる状態, トラブル
シューティング, 94-95
高速リブートに関する問題のデバッグ, 94-95
高速リブートの開始, 新しくアクティブになった
ブート環境, 47-49
高速リブートのトラブルシューティング, 94-95

し

システムシャットダウンコマンド, 35
システムのクラッシュダンプとリブート, 強制的
な実行, 91-93
システムの高速リブートのブート, (方法), 47

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
 シャットダウン, 35
 ガイドライン, 34-36
 (タスクマップ), 33-34
 復旧の目的, 86-87

システムのブート
 ガイドライン, 17-18
 実行レベル S, 30-31
 シングルユーザー状態, 30-31
 対話式, 31-32

システムのブート可能状態の維持, 79-80
 タスク, 79-84

システムのブートのトラブルシューティング,
 kmdb コマンド, 93-94

システムの復旧, システムを停止する方法, 87

システムのリブート, クラッシュダンプの強制的
 な実行, 91-93

実行レベル
 0 (電源切断レベル), 20
 1 (シングルユーザーレベル), 20
 2 (マルチユーザーレベル), 20
 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, 21
 ブート, 29
 3 (NFS を持つマルチユーザー), 20
 6 (リブートレベル), 20
 s または S (シングルユーザーレベル), 20
 確認 (方法), 28
 定義, 20
 デフォルトの実行レベル, 20

実行レベル 3, マルチユーザー状態, 29

実行レベル 3 へのブート, マルチユーザー状態, 29

実行レベル S, システムをシングルユーザー状態に
 ブートする方法, 30-31

実行レベルとマイルストーンの使い分け, 21

実行レベルまたはマイルストーン, 使い分け, 21

指定された状態への x86 ベースシステムのブート,
 (タスクマップ), 27-28

シングルユーザー状態
 システムのブート
 実行レベル S, 30-31

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

そ

早期システムパニック

デバッグ

高速リブート, 94

早期システムパニックのデバッグ, 高速リ
 ブート, 94

た

対話式ブート, システムのブート (方法), 31-32

て

停止

復旧を目的としたシステムの停止 (方法)

x86, 87

デバイスの命名規則, GRUB, 25

デバイス命名規則, GRUB, 25

と

トラブルシューティング, 高速リブート, 94-95

は

パニック, 高速リブートのデバッグ, 94

ふ

ブート, トラブルシューティングの方法, 86-87

ブートアーカイブ

管理, 79-84

操作, 79-80

ブート時の対話, GRUB メニュー, 64-66

ブート動作

GRUB メニューでの変更方法, 61

管理, 55-68

ブートのトラブルシューティング

クラッシュダンプの強制的な実行, 91-93

方法, 86-87

ブートの用語, x86, 25-26

ブートパラメータ, x86 ベースシステム上の変

更, 55-57

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, 86-87

ま

マイルストーンまたは実行レベル, 使い分け, 21

マルチユーザー状態, ブート (方法), 29

マルチユーザーレベル, 「実行レベル 3」を参照

め

メニュー

GRUB

説明, 64-66

よ

用語, x86 ブート, 25-26