

Oracle® Solaris 11.3 システムのブートと シャットダウン

ORACLE®

Part No: E62516
2016年11月

Part No: E62516

Copyright © 1998, 2016, Oracle and/or its affiliates. All rights reserved.

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.

このソフトウェアまた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。このソフトウェアまたはハードウェアは、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。このソフトウェアまた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場合、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。このソフトウェアまた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、Oracle Corporation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はオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

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。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporation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。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporation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。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用法	9
1 システムのブートとシャットダウンの概要	11
システムのブートおよびシャットダウンの新機能	11
システムのブートに関するガイドライン	12
システムをブートする理由	13
Oracle Solaris ブートアーキテクチャーの概要	13
Oracle Solaris ブートアーカイブの説明	16
ブートプロセスの説明	17
x86: UEFI と BIOSのブート方法の相違点	18
サービス管理機構とブート	19
SMF 使用時のブート動作に関する変更	20
2 x86: GRand Unified Bootloader の管理	21
x86: GRUB 2 の概要	21
x86: GRUB 2 構成の説明	23
x86: GRUB 2 パーティションおよびデバイス命名スキーム	24
x86: GRUB 2 と GRUB Legacy のタスクの比較	27
x86: bootadm コマンドを使用して GRUB 構成を管理する	30
x86: GRUB メニューのパスワード保護	32
x86: ユーザーへの GRUB メニューにアクセスする承認の付与	33
x86: GRUB メニューまたは特定のメニューエントリの表示	33
▼ x86: GRUBメニューを手動で再生成する方法	34
▼ x86: GRUB メニューを変更する方法	35
▼ x86: GRUB メニューの指定したブートエントリの属性を設定する方 法	38
▼ x86: GRUB メニューにブートエントリを追加する方法	41
▼ x86: GRUB メニューからブートエントリを削除する方法	42
x86: ブート時に GRUB メニューを編集してカーネル引数を追加する	43

x86: GRUB メニューを編集して、ブート時に <code>-B prop=val</code> カーネル引数を追加する	45
x86: GRUB 構成のカスタマイズ	47
x86: GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする	48
▼ x86: GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする方法	49
x86: GRUB Legacy メニューエントリが GRUB 2 に移行される方法	52
x86: GRUB 2 と GRUB Legacy のブート環境を同じシステムで保守する	53
x86: 高度な GRUB 管理とトラブルシューティング	54
x86: <code>bootadm install-bootloader</code> コマンドを使用して GRUB 2 をインストールする	55
x86: GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする	56
3 システムのシャットダウン	59
システムのシャットダウン	59
システムのシャットダウンに関するガイドライン	60
システムシャットダウンコマンド	60
システムのシャットダウン	61
▼ システムにログインしているユーザーを確認する方法	61
▼ <code>shutdown</code> 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	62
▼ <code>init</code> コマンドを使用してスタンドアロンシステムをシャットダウンする方法	66
システムデバイスの電源の切断	66
4 システムのブート	69
ブート属性の表示と設定	69
SPARC: OpenBoot PROM を使用したブート属性の表示と設定	70
EEPROM パラメータの使用	74
x86: SMF によるシャットダウンアニメーションの管理	78
システムのブート	79
実行レベルの動作	79
▼ 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	81
▼ 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	83

▼ システムを対話式でブートする方法	87
代替のオペレーティングシステムまたはブート環境からのブート	91
▼ SPARC: 代替のオペレーティングシステムまたはブート環境から ブートする方法	91
▼ x86: 代替のオペレーティングシステムまたはブート環境からブート する方法	93
システムのリブート	94
▼ init コマンドを使用してシステムをリブートする方法	95
▼ reboot コマンドを使用してシステムをリブートする方法	96
リブートプロセスの高速化	96
5 システムのネットワークからのブート	103
SPARC: システムのネットワークからのブート	103
SPARC: ネットワークブートプロセス	104
SPARC: システムをネットワークからブートするための要件	104
SPARC: OpenBoot PROM へのネットワークブート引数の設定	105
SPARC: DHCP を使用して自動ブートするための NVRAM 別名の設 定	107
▼ SPARC: システムをネットワークからブートする方法	107
x86: システムのネットワークからのブート	108
x86: システムをネットワークからブートするための要件	109
x86: GRUB 2 PXE ブートイメージがインストールされる場所	110
x86: UEFI および BIOS ファームウェアを搭載するシステムのネット ワークからのブート	111
▼ x86: システムをネットワークからブートする方法	112
6 ブートプールを含むシステムの管理	115
ファームウェアからアクセス不可能なストレージデバイスからのブートの概 要	115
ブートプール、ブートプールデータセット、およびフォールバックイメージ の管理	115
▼ BE をブートする機能を保持する方法	116
すべてのブートプールデータセットに対する削除動作の変更	117
▼ BE をブート可能にする方法	117
▼ フォールバックイメージを更新する方法	118
Oracle Solaris 11.3 の OpenBoot プロパティ	119
os-root-device 変数	119
▼ 通常のブートを有効にする方法	120

7 システムのブートに関するトラブルシューティング	123
Oracle Solaris ブートアーカイブの管理	123
▼ ブートアーカイブの内容を一覧表示する方法	124
boot-archive SMF サービスの管理	124
▼ 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	125
▼ x86: 高速リブートをサポートしないシステムのブートアーカイブの 自動更新障害を解決する方法	126
復旧目的のシステムのシャットダウンおよびブート	127
▼ SPARC: 復旧を目的としてシステムを停止する方法	128
▼ x86: 復旧目的でのシステムの停止とリブートの方法	129
▼ シングルユーザー状態でブートして、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	130
▼ 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	131
▼ x86: 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 GRUB 構成の問題を解決する方法	133
クラッシュダンプを強制してシステムをリブートする	134
▼ SPARC: 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	134
▼ x86: 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	136
カーネルデバッグ (kldb) を有効にしてシステムをブートする	137
▼ SPARC: カーネルデバッグ (kldb) を有効にしてシステムをブートす る方法	137
▼ x86: カーネルデバッグ (kldb) を有効にしてシステムをブートする方 法	138
x86: 高速リブートに関する問題のトラブルシューティング	139
x86: 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	140
x86: 高速リブートが機能しない可能性のある状況	140
ブートとサービス管理機構の問題のトラブルシューティング	141
インストール後のブートに関する問題	141
A Oracle Solaris ブートプロセス	143
ブートプロセスの変更点	143
フォールバックイメージからのブート	144
索引	147

このドキュメントの使用方法

- **概要** – システムをブートおよびシャットダウンする方法について説明します。
- **対象読者** – 技術者、システム管理者、および認定サービスプロバイダ
- **必要な知識** – Oracle Solaris システムの管理経験

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E62101-01> で入手可能です。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

システムのブートとシャットダウンの概要

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、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、Oracle Solaris 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。この章で SPARC または x86 ベースのシステムのみにも適用される情報については、その旨が示されています。

注記 - このドキュメントでは主に、物理マシン上の単一の 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備えたシステムや、複数の物理ドメインを含むシステム上の Oracle Solaris のブートとシャットダウンに関する情報は、詳細には記載されていません。詳細については、<http://www.oracle.com/products/oracle-a-z.html> で、使用している特定のハードウェアや構成についての製品ドキュメントを参照してください。

この章では、次の内容について説明します。

- 11 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」
- 12 ページの「システムのブートに関するガイドライン」
- 13 ページの「Oracle Solaris ブートアーキテクチャーの概要」
- 17 ページの「ブートプロセスの説明」
- 19 ページの「サービス管理機構とブート」

システムのブートおよびシャットダウンの新機能

このリリースでは、GRUB メニューをセキュリティー保護するために `bootadm` コマンドが拡張されています。GRUB メニューからの表示、編集、またはブートを、特定のパスワードを持つ特定の 1 人または複数のユーザーにのみ許可できるようになりました。詳細は、30 ページの「[bootadm コマンドを使用して GRUB 構成を管理する](#)」を参照してください。

IP over Infiniband (IPoIB) を使用してアクセスする iSCSI デバイスなどの、ファームウェアからアクセス不可能なストレージデバイスを使用してブートできるシステムでは、Oracle Solaris ブートプロセスが次のように拡張されています。

- ファームウェアからアクセス可能なデバイス上のブートプール内のブートアーカイブにアクセスできます。ブートアーカイブには、そのブートデータセットが関連付けられているブート環境 (BE) の Oracle Solaris カーネルのブートに必要なファイルのセットが含まれています。ブートプールには、ブートローダーのデータファイルと回復データが含まれています。ブートプール内の各データセットは BE にリンクされています。詳細は、[第6章「ブートプールを含むシステムの管理」](#)を参照してください。
- OpenBoot からはアクセスできない、ファームウェアからアクセス不可能なストレージデバイス上にルートプールを配置できます。ルートプールを作成すると、OpenBoot からアクセス可能なデバイス上にブートプールが自動的に作成されます。詳細は、[143 ページの「ブートプロセスの変更点」](#)を参照してください。
- OpenBoot からブートプール内のデバイスにアクセスできない場合は、サービスプロセッサ (SP) 上に格納されているフォールバックイメージからブートできます。このフォールバックイメージは、サービスプロセッサとそれに関連付けられた rKVMS サービスにアクセスできる任意のドメインで使用できます。詳細は、[144 ページの「フォールバックイメージからのブート」](#)を参照してください。

システムのブートに関するガイドライン

ブートストラップとは、ブート可能なオペレーティングシステムをロードし、実行するプロセスです。一般に、スタンドアロンプログラムはオペレーティングシステムカーネルですが、任意のスタンドアロンプログラムをブートできます。カーネルはロードされると、UNIX システムを起動して、必要なファイルシステムをマウントし、`/usr/sbin/init` を実行して、システムを `/etc/inittab` ファイルに指定されている `initdefault` 状態にします。

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。

- SPARC システムをシャットダウンしたあとでブートするには、PROM レベルで `boot` コマンドを使用します。SPARC ベースのシステムをオンにすると、システムファームウェア (PROM 内) は電源投入時自己診断 (POST) を実行します。これらの診断テストの形式と範囲は、システムに搭載されているファームウェアのバージョンによって異なります。テストが正常に完了したあと、ファームウェアが使用する不揮発性ストレージ領域に適切なフラグが設定されていれば、ファームウェアは自動ブートを試みます。ロードするファイルの名前とそのファイルをロードするデバイスも操作することができます。
- ブート時に表示される GRUB メニューでオペレーティングシステムを選択して、x86 ベースのシステムをブートします。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、`grub.cfg` 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してもシステムをリブートできます。

システムをブートする理由

次の表に、システムのブートが必要になる理由を示します。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 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	詳細
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」
/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)	81 ページの「システムをマルチユーザー状態 (実行レベル 3) にブートする方法」
ファイルシステムを保守します (システムデータのバックアップや復元など)	シングルユーザー状態 (実行レベル S) で Ctrl + D キーを押してシステムをマルチユーザー状態 (実行レベル 3) に戻します	83 ページの「システムをシングルユーザー状態 (実行レベル S) にブートする方法」
/etc/system などのシステム構成ファイルを修復します	対話式ブート	87 ページの「システムを対話式でブートする方法」
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 11.3 でのデバイスの管理』の「ZFS ファイルシステム用のディスクの設定」
root パスワードがわからなくなったか、ファイルシステムを修正するなどの問題のため、復旧の目的でシステムをブートします。	エラー状況や問題によっては、システムをメディアからブートするか、ブート環境をマウントするか、またはその両方を実行する必要がある場合があります。	127 ページの「復旧目的のシステムのシャットダウンおよびブート」
x86 のみ: GRUB 構成の問題から回復します。	メディアから回復ブートします。	133 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 GRUB 構成の問題を解決する方法」
クラッシュダンプを強制してシステムをハング状態から回復させます。	回復ブート	134 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」 136 ページの「クラッシュダンプを強制してシステムをリブートする方法」
カーネルデバッガ (kmdb) を使用してシステムをブートし、システムの障害を調査します	kmdb のブート	137 ページの「カーネルデバッガ (kmdb) を有効にしてシステムをブートする方法」 138 ページの「カーネルデバッガ (kmdb) を有効にしてシステムをブートする方法」

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、次の基本特性があります。

- **ブートアーカイブを使用します。**

ブートアーカイブには、メモリー内ディスクを使用してマウントされるファイルシステムイメージが含まれています。このイメージは自己記述型であり、特にブートブロックにはファイルシステムリーダー (または x86 プラットフォームの場合に GRUB ブートローダー) が含まれています。SPARC プラットフォームで、ファイルシステムリーダーは RAM ディスクイメージをマウントして開いたあと、その中に含まれているカーネルを読み取って実行します。デフォルトで、このカーネルは `/platform/`uname -m`/kernel/unix` にあります。x86 プラットフォームでは、GRUB ブートローダーが、カーネルファイルとブートアーカイブをメモリーにロードしてから、制御をカーネルに転送します。x86 プラットフォームのデフォルトのカーネルは `/platform/i86pc/kernel/amd64/unix` です。

注記 - ディスクから SPARC ベースのシステムをブートすると、OBP ファームウェアがブートデバイスとして指定されているパーティションからブートブロックを読み取ります。このスタンドアロンブートプログラムには通常、Oracle Solaris ブートアーカイブを読み取れるファイルリーダーが含まれています。 [boot\(1M\)](#) を参照してください。

ZFS ルートファイルシステムからブートする場合は、アーカイブとカーネルファイルの両方のパス名が、ブート用に選択されたルートファイルシステム (データセット) 内で解決されます。

- **ブート管理インタフェースを使用して、Oracle Solaris ブートアーカイブを保守し、x86 プラットフォームで GRUB 構成と GRUB メニューを管理します。**

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

x86 ベースのシステムでは、`grub.cfg` ファイルと x86 ブートローダーは `bootadm` コマンドで管理します。このリリースでは、`bootadm` コマンドが変更され、いくつかの新しいサブコマンドが追加されて、これまで `menu.lst` ファイルを編集して実行していたほとんどの管理タスクを実行できるようになりました。これらのタスクには、GRUB メニューの管理、特定のブートエントリへのカーネル引数の設定、ブートローダーの管理などが含まれます。手順については、[30 ページの「bootadm コマンドを使用して GRUB 構成を管理する」](#) を参照してください。

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

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

- **インストール時にルートファイルシステムとして RAM ディスクイメージを使用します。**

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

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

ソフトウェアをインストールする場合、RAM ディスクイメージは、インストールプロセス全体に使用されるルートファイルシステムになります。この目的の RAM ディスクイメージにより、リムーバブルメディアから、頻繁に必要なインストールコンポーネントにアクセス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) または UFS を指定できます。

- **GPT ラベル付きディスクからのブートをサポートします。**

Oracle Solaris には、GPT ラベル付きディスクからのブートのサポートが含まれます。GPT ラベル付きディスクからのブートは、MSDOS パーティション分割スキームを使用したディスクからのブートとはやや異なります。x86 または GPT 対応ファームウェアを搭載した SPARC ベースのシステムに Oracle Solaris 11.3 をインストールすると、ほとんどの場合にディスク全体を使用するルートプールディスクで GPT ディスクラベルが適用されます。サポートされる SPARC ベースのシステムで GPT 対応ファームウェアを適用する方法については、『[Oracle Solaris 11.3 ご使用にあたって](#)』の「[SPARC: GPT ラベル付きディスクのサポート](#)」を参照してください。それ以外の場合は、SPARC ベースのシステムに Oracle Solaris 11.3 をインストールすると、単一のスライス 0 でルートプールディスクに SMI (VTOC) ラベルが適用されます。

x86 プラットフォームでは、GRUB 2 の導入により、このサポートが可能になります。BIOS ファームウェアを搭載したシステムでは、MBR は、引き続きファームウェアがロードしてブートプロセスを開始する最初のコードのチャンクです。GPT ラベル付きディスクには VTOC がなくなり、個別のパーティションだけです。GRUB は、GPT パーティション分割スキームの読み取りおよび解釈を直接サポートするようになったため、ブートローダーが、ZFS GPT パーティションでホ

ストされるルートプール内の Oracle Solaris カーネルおよびブートアーカイブを検出できます。

UEFI ファームウェアを搭載したシステムでの主な違いは、ファームウェアがブートアプリケーションを (FAT ベースの) EFI システムパーティションからロードすることです。UEFI システムに GRUB がロードされると、BIOS 対応の GRUB と同様のタスクを実行します。

Oracle Solaris ブートアーカイブの説明

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。

`bootadm` コマンドは、ブートアーカイブの更新と検証の詳細など、SPARC プラットフォームと x86 プラットフォームの両方で、ブートアーカイブを管理します。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った場合、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 ブートアーカイブに含まれるファイルは、`/platform/i86pc/amd64/archive_cache` ディレクトリにあります。SPARC ブートアーカイブ内のファイルは `/platform/^uname -m^/archive_cache` ディレクトリにあります。SPARC プラットフォームと x86 プラットフォームの両方で、ブートアーカイブの内容を一覧表示するには、`bootadm list-archive` コマンドを使用します。

```
$ bootadm list-archive
```

ブートアーカイブ内のファイルが更新された場合、アーカイブを再構築する必要があります。`bootadm update-archive` コマンドにより、ブートアーカイブを手動で再構築できます。コマンドは、予防措置としても回復プロセスの一部としても使用できます。

```
# bootadm update-archive
```

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。詳細は、[123 ページの「Oracle Solaris ブートアーカイブの管理」](#)を参照してください。

ブートプロセスの説明

このセクションでは SPARC プラットフォームと x86 プラットフォーム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/products/oracle-a-z.html> にある特定ハードウェアについての製品ドキュメントを参照してください。

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

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

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

x86 ベースのシステムでは、ブートストラッププロセスは、カーネルのロードと、カーネルの初期化という概念的に異なる 2 つのフェーズからなります。カーネルのロードは、システムボード上のファームウェアおよび周辺ボード上の ROM のファームウェア拡張機能を使用して、GRUB によって実装されます。システムファームウェアは GRUB をロードします。ロードメカニズムは、システムボード上に標準装備されるシステムファームウェアのタイプにより異なります。

- PC 互換システムの電源を投入すると、システムのファームウェアが電源投入時自己診断テスト (POST) を実行し、周辺ボードの ROM からファームウェア拡張機能を検出してインストールし、ファームウェア固有のメカニズムを使ってブートプロセスを開始します。
- BIOS ファームウェアを搭載するシステムでは、ハードディスクの最初の物理セクター (ブートセクターと呼ばれる) がメモリーにロードされて、そのコードが実行されます。GPT (GUID Partition Table) を使用してパーティション化されるディスクには、別の場所からコードをロードする、動作の異なるブートセクターコードが必要です (GPT スキームは各パーティションの最初のセクターをブートセクターのコードストレージとして予約しないため)。GRUB が BIOS ファームウェア上で実行される場合、別の場所とは BIOS ブートパーティションと呼ばれる専用のパーティションです。GRUB のブートセクターコードが GRUB の残りをメモリーにロードすると、ブートプロセスが続行されます。

次に、ブートプログラムは次の段階 (Oracle Solaris の場合は GRUB 自体) をロードします。ネットワークからのブートでは、BIOS ファームウェアを搭載するシステ

ムと異なるプロセスが含まれます。詳細は、[第5章「システムのネットワークからのブート」](#)を参照してください。

- UEFI ベースのファームウェアを搭載するシステムでは、ブートプロセスが大きく異なります。UEFI ファームウェアは、列挙されたディスク上の EFI システムパーティション (ESP) を検索し、UEFI 仕様で定義されたプロセスに従って、UEFI ブートプログラムをロードして実行するため、UEFI ブートアプリケーションがメモリーにロードされ、実行されます。Oracle Solaris では、その UEFI ブートアプリケーションは GRUB です。このリリースの GRUB のバージョンは、UEFI ブートアプリケーションとして実行するように構築されています。ブートプロセスは、BIOS ファームウェアを搭載したシステムの場合と同じ処理を続けます。

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: UEFI と BIOSのブート方法の相違点

GRUB 2 は GPT ラベル付きディスクに加えて、BIOS と UEFI の両方のファームウェアを搭載するシステムをブートできます。UEFI ファームウェアおよび BIOS ファームウェアのブートをサポートするために、GRUB 2 は、i386-pc (BIOS) と x86_64-efi (64 ビット UEFI 2.1+) の 2 つの異なるプラットフォームに対応するように構築されているため、2 つの個別のバイナリセットとして配布されています。

x86 ベースのシステムをブートする場合、UEFI 対応システムと BIOS 対応システムの次の違いに注意してください。

- **コマンドの違い** – BIOS ブート方法で使われる特定のコマンドは UEFI ファームウェアで使用できません。同様に、BIOS ブート方法をサポートするシステムでは、特定の UEFI コマンドを使用できません。
- **PXE ネットワークブートの違い** – ネットワークから、UEFI ファームウェアを搭載するシステムのブートをサポートするために、DHCP サーバー構成が変更されました。これらの変更には、新しい UEFI クライアントアーキテクチャー識別子値 (DHCP オプション 93) のサポートが含まれます。

注記 - UEFI ファームウェアまたは BIOS ブート方法のいずれかを使用して、ブートするように構成できるシステムは、技術的に Oracle Solaris と連携します。GRUB は、インストール (またはイメージ更新) 時に、システムファームウェアのタイプに従って最初にインストールされます。明示的なコマンドを実行して、ほかのファームウェアタイプが必要とするブートの場所に GRUB をインストールすることはできますが、この方法はサポートされていません。Oracle Solaris をインストールしたあとに、代替のファームウェアインタフェースタイプを使用してブートするように、特定のファームウェアタイプのシステムを再構成しないようにしてください。

x86: UEFI および BIOS ファームウェアを搭載するシステムをサポートするブートパーティションの作成

新しい `-B` オプションが `zpool create` コマンドに追加されました。ディスク全体が `zpool create` 作成コマンドに渡されると、`-B` オプションにより、`zpool` コマンドは、指定されたデバイスを2つのパーティションで分割します。最初のパーティションはファームウェア固有のブートパーティション、2つ目のパーティションは ZFS データパーティションです。このオプションは、必要に応じて、ディスク `vdev` 全体を既存の `rpool` に追加するか接続する場合に、必要なブートパーティションを作成するためにも使用できます。`bootfs` プロパティを使用できる条件も変更されました。プールですべてのシステムおよびディスクのラベル付け要件が満たされている場合に、プールのブート可能データセットを識別するように `bootfs` プロパティを設定できます。ラベル付けの要件の一部として、必要なブートパーティションも存在している必要があります。詳細は、『Oracle Solaris 11.3 での ZFS ファイルシステムの管理』の「ZFS ルートプールの管理」を参照してください。

サービス管理機構とブート

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

システムがブートされる時、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。次に例を示します。

- SPARC ベースのシステムで次のコマンドを使用すると、ブート先の特定のマイルストーンを選択できます。

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

デフォルトのマイルストーンは、有効になっているすべてのサービスを起動する `all` です。ほかの有用なマイルストーンは `none` で、これは `init`、`svc.startd`、および `svc.configd` のみを開始します。このマイルストーンは、サービスを手動で開始できる非常に有用なデバッグ環境を提供します。`none` マイルストーンの使用の詳細は、『Oracle Solaris 11.3 でのシステムサービスの管理』の「システムブート時のサービスの起動に関する問題を調査する方法」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
```

選択できるロギングレベルは `quiet`、`verbose`、および `debug` です。ロギングレベルの具体的な情報については、『[Oracle Solaris 11.3 でのシステムサービスの管理](#)』の「[起動メッセージの量の指定](#)」を参照してください。

- x86 ベースのシステムを特定のマイルストーンにブートするか、`svc.startd` のログのレベルを選択するには、ブート時に GRUB メニューを編集し、`-m smf-options` カーネル引数を指定したブートエントリの `$multiboot` 行の末尾に追加します。
例:

```
$multiboot /ROOT/s11.3_18/!/$kern $kern -B $zfs_bootfs -m logging-level
```

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が付きません。それ以外の機能には新しいコマンドでアクセスします。

もっともよく目にする動作の変更の一覧を次に示します。

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

◆◆◆ 第 2 章

x86: GRand Unified Bootloader の管理

この章では、GRand Unified Bootloader (GRUB) の概要およびタスクに関する情報を提供します。元の GRUB 0.97 ベースのブートローダーの子孫である GRUB 2 は、このリリースの x86 プラットフォーム上でのシステムブートローダーです。

注記 - 元の GRUB (GRUB Legacy) は、Oracle Solaris 10 および Oracle Solaris 11 11/11 を実行する x86 プラットフォーム上で、引き続きデフォルトのブートローダーです。GRUB のレガシーバージョンをサポートする Oracle Solaris リリースを実行している場合、[Booting and Shutting Down Oracle Solaris on x86 Platforms](#)を参照してください。

この章の内容は次のとおりです。

- 21 ページの「GRUB 2 の概要」
- 30 ページの「bootadm コマンドを使用して GRUB 構成を管理する」
- 43 ページの「ブート時に GRUB メニューを編集してカーネル引数を追加する」
- 47 ページの「GRUB 構成のカスタマイズ」
- 48 ページの「GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする」
- 54 ページの「高度な GRUB 管理とトラブルシューティング」

x86: GRUB 2 の概要

このセクションでは、次の情報について説明しています。

- 23 ページの「GRUB 2 構成の説明」
- 24 ページの「GRUB 2 パーティションおよびデバイス命名スキーム」
- 27 ページの「GRUB 2 と GRUB Legacy のタスクの比較」

GRUB 2 は、BIOS または UEFI ファームウェアを搭載したシステム上の Unified Extensible Firmware Interface (UEFI) ファームウェアからのブートや任意のサイズの

GUID パーティションテーブル (GPT) でパーティション分割されたディスクからのブートなど、幅広いプラットフォームとファームウェアのタイプをサポートする強力なモジュラーブートローダーです。GRUB 2 は UEFI 指定の GPT パーティション分割スキームもサポートしています。

GRUB Legacy と同様に、GRUB 2 は 2 段階のブートプロセスを使用します。GRUB 2 と GRUB Legacy の主な違いは、GRUB 2 では動的にロードされたモジュールに多くの機能を配置し、これにより、コア GRUB 2 (第 2 段階ブートローダー) のイメージを小さくできるため、ロードが速くなり、柔軟性が高まることです。結果として、ブート時に要求に応じて GRUB 機能がロードされます。

GRUB 2 には次の主な変更が導入されています。

■ 構成の変更

GRUB 2 構成は、GRUB Legacy 構成と構文的に異なります。GRUB Legacy によって使用される `menu.lst` ファイルは、新しい構成ファイル `grub.cfg` に置き換えられました。`menu.lst` ファイルと異なり、`grub.cfg` ファイルはブート管理コマンドによって自動的に再生成されます。そのため、`grub.cfg` ファイルの再生成時に、すべての編集がただちに破棄されるため、このファイルは直接編集しないでください。23 ページの「GRUB 2 構成の説明」を参照してください。

■ パーティションとデバイスの命名の変更

0 を基準にするインデックスの代わりに、GRUB 2 ではパーティションと変更されたデバイス命名スキームに 1 を基準にするインデックスを使用します。24 ページの「GRUB 2 パーティションおよびデバイス命名スキーム」を参照してください。

■ ブートローダーと GRUB メニューの管理の変更

`bootadm` コマンドによって `grub.cfg` ファイルを管理します。変更されたサブコマンドと新しいサブコマンドによって、これまで `menu.lst` ファイルを編集することによって行っていた管理タスクのほとんどを管理できるようになります。2 つの例として、Oracle Solaris ブートインスタンスのブート属性 (カーネル引数など) の設定とブートローダー設定の管理があります。30 ページの「`bootadm` コマンドを使用して GRUB 構成を管理する」を参照してください。

■ GRUB メニューと画面の変更

ブート時に GRUB メニューを編集することによって、カーネル引数を追加するなど、さまざまな GRUB メニューやいくつかのタスクがいくぶん異なって動作するようになりました。これらの違いは、このドキュメント内の各種タスクの該当箇所でも説明しています。

■ その他のブートローダー関連のコマンドの変更

`installgrub` コマンドはこのリリースで非推奨になりました。このコマンドを使用して、GRUB 2 をサポートするシステムにブートローダーをインストールしないでください。システムのブートが妨げられることがあります。代わりに、GRUB 2 をサポートするリリースを実行する場合は、`bootadm install-bootloader` コマンドを使用してください。このコマンドは、x86 プラットフォームでの `installgrub` コマンドおよび SPARC プラットフォームでの `installboot`

コマンドの機能に優先されます。55 ページの「[bootadm install-bootloader コマンドを使用して GRUB 2 をインストールする](#)」を参照してください。

`installgrub` コマンドを使用して、システムに GRUB Legacy をインストールすることもできますが、インストールする GRUB Legacy のバージョンが、ルートプールの ZFS プールバージョンをサポートしており、さらにシステムに残りの GRUB 2 ブート環境がないことを確認したあとに限ります。手順については、56 ページの「[GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする方法](#)」を参照してください。

x86: GRUB 2 構成の説明

GRUB 2 は GRUB Legacy とまったく異なる構成を使用します。GRUB Legacy 構成は `menu.lst` ファイルから管理しますが、GRUB 2 は `menu.lst` ファイルを使用しません。代わりに、GRUB 2 は構成ファイル `grub.cfg` を使用して、同じタイプの情報を格納します。`menu.lst` ファイルと同様に、`grub.cfg` ファイルはルートプール `/pool-name/boot/grub` の ZFS データセットの最上位レベルにあり、たとえば `/rpool/boot/grub/grub.cfg` などです。

`grub.cfg` ファイルの構文は、`bash` スクリプトに基づいており、次の例に示す `menu.lst` ファイルで使用されている指令のような言語よりも複雑で強力です。

```
title title
    bootfs pool-name/ROOT/bootenvironment-name
    kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS
    module$ /platform/i86pc/$ISADIR/boot_archive
```

比較のために、次のように、同じ構成を `grub.cfg` ファイルに格納しています。

```
menuentry "<title>" {
    insmod part_msdos
    insmod part_sunpc
    insmod part_gpt
    insmod zfs
    search --no-floppy --fs-uuid --set=root f3d8ef099730bafa
    zfs-bootfs /ROOT/<BE name>/@/ zfs_bootfs
    set kern=/platform/i86pc/kernel/amd64/unix
    echo -n "Loading ${root}/ROOT/<BE name>/@$kern: "
    $multiboot /ROOT/<BE name>/@$kern $kern -B $zfs_bootfs
    set
    gfxpayload="1024x768x32;1024x768x16;800x600x16;640x480x16;640x480x15;640x480x32"
    insmod gzio
    echo -n "Loading ${root}/ROOT/<BE name>/@/platform/i86pc/amd64/boot_archive: "
    $module /ROOT/<BE name>/@/platform/i86pc/amd64/boot_archive
}
```

`grub.cfg` ファイルと `menu.lst` ファイルのほかの大きな違いは、`grub.cfg` ファイルを編集しないことです。`menu.lst` ファイルでは、システムによって自動的に生成されるメニューエントリに加えて、ユーザー作成メニューエントリと、グローバル GRUB 構成ファイルの設定および変数の手動の変更をサポートしています。対照的に、`grub.cfg` ファイルは、各種 `bootadm` サブコマンドを使用して管理します。それ

らの多くはこのリリースで新しいものです。各種 `bootadm` サブコマンドにより、ほとんどのブートローダー管理タスクを管理できます。30 ページの「[bootadm コマンドを使用して GRUB 構成を管理する](#)」を参照してください。

`grub.cfg` ファイルの別の機能は、ブートローダー構成を変更するたびに自動生成されることです。ファイルは、特定の操作時および特定のブート管理コマンドが使用されたときにも自動生成されます。必要に応じて、`bootadm` コマンドと新しい `generate-menu` サブコマンドを実行して、新しい `grub.cfg` ファイルを手動で生成できます。このコマンドは、ブート構成が破損した場合にのみ、`grub.cfg` ファイルを作成するために使用します。34 ページの「[GRUB メニューを手動で再生成する方法](#)」を参照してください。

注記 - `grub.cfg` ファイルの自動生成メカニズムは、インストールされたシステムのみを対象とします。

参考のため、GRUB 2 構成は次のファイルに格納されています。

- `grub.cfg` - GRUB 2 によって使用されるメインの構成ファイルです。
- `/pool-name /boot/grub/menu.conf` - Oracle Solaris によって、最終の `grub.cfg` 構成ファイルを生成するために使用されるファイルです。
`menu.conf` ファイルは、GRUB 2 構成のマシン解析可能な表現を格納する個別の GRUB メタ構成ファイルです。

注記 - このファイルは編集しないでください。

- `/pool-name /boot/grub/custom.cfg` - `grub.cfg` ファイルおよび `menu.conf` ファイルと同じ場所にある編集可能なファイルです。`custom.cfg` ファイルは、管理者によって作成されます (デフォルトでシステム上に存在しません)。このファイルは基本 GRUB 構成に複雑な構造 (メニューエントリや他のスクリプト情報) を追加する目的で予約されています。

`custom.cfg` ファイルは `grub.cfg` ファイル内で参照されています。`custom.cfg` ファイルがシステムに存在する場合、そのファイル内にあるコマンドや指令が `grub.cfg` ファイルによって処理されます。詳細は、47 ページの「[GRUB 構成のカスタマイズ](#)」を参照してください。

x86: GRUB 2 パーティションおよびデバイス命名スキーム

GRUB Legacy のデバイス命名のしくみに精通している場合、GRUB Legacy の命名スキームと GRUB 2 命名スキームの違いを認識する必要があります。GRUB Legacy は

パーティションのインデックスに、0 を基準にする命名スキームを使用しますが、GRUB 2 は、パーティションのインデックスに 1 を基準にする命名スキームを使用します。

GRUB 2 デバイス命名スキームでは、次の形式を使用します。

(hdX, part-typeY, part-typeZ, ...)

パーティションスキームは入れ子にできるため、GRUB のデバイス命名スキームは、任意の入れ子のレベルをサポートするように変更されました。GRUB では、古いスタイルのデバイス命名 (「hd0, 1」) も、パーティションスキーム名を含む新しいスタイルのデバイス命名のいずれも受け付けます。例:

(hd0, gpt1)

前の例では、先頭のディスクの先頭の GPT パーティションを参照しています。

注記 - GRUB のパーティションの番号付けのみが変更されており、ディスクの番号付けは変更されていません。ディスク番号は 0 基準のままです。

GRUB 2 は、正しいデバイスまたはパーティション名を自動的に検索するために、ファイルシステム UUID (またはラベル) と組み込みの検索コマンドに依存しているため、手動でデバイス名を指定する必要がありません。次の表に、GRUB が使用するパーティションインデックスとデバイス名の例を示します。

表 2 GRUB 2 パーティションおよびデバイス命名スキーム

デバイス名	説明	注意事項
(hd0, msdos1)	先頭のディスク上の先頭の DOS パーティションを指定します。	
(hd0, gpt2)	ディスク上の 2 番目の GPT パーティションを指定します。	これは、現在のリリースをインストールする典型的なパーティションの例です。
(hd0, msdos1, sunpc1)	先頭のディスク上の先頭の DOS パーティションに格納されている Oracle Solaris パーティション内の先頭の VTOC スライスを指定します。	これは、このリリース以前の Oracle Solaris のバージョンをインストールする典型的なパーティションの例です。

関心のあるパーティションを参照しているパーティション番号を特定する必要がある場合、**C** キー (メニューエントリを編集している場合は **Control-C**) を押して、GRUB コマンド行インタプリタにアクセスします。次に、**ls** コマンドを実行して、次の図のように、GRUB が識別できるすべてのパーティションを一覧表示します。

```
GNU GRUB version 1.99,5.11.0.175.1.0.0.13.18988

Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device or file completions. ESC at any time exits.

grub> ls
(hd0) (hd0,gpt9) (hd0,gpt2) (hd0,gpt1) (fd0)

grub> _
```

ls コマンドで -l オプションを使用すれば、次の図のように、ファイルシステムおよびファイルシステム UUID 情報などの各パーティションに関する詳細情報が表示されます。

```
GNU GRUB version 1.99,5.11.0.175.1.0.0.13.18988

Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device or file completions. ESC at any time exits.

grub> ls
(hd0) (hd0,gpt9) (hd0,gpt2) (hd0,gpt1) (fd0)

grub> ls -l
Device hd0: Not a known filesystem - Total size 33554432 sectors
  Partition hd0,gpt9: Not a known filesystem - Partition start at
33538015 - Total size 16384 sectors
  Partition hd0,gpt2: Filesystem type zfs - Label "rpool" - Last
modification time 2012-04-04 05:20:24 Wednesday, UUID 165a678b96efcad5 -
Partition start at 524544 - Total size 33013471 sectors
  Partition hd0,gpt1: Not a known filesystem - Partition start at 256 -
Total size 524288 sectors
Device fd0: Not a known filesystem - Total size 2880 sectors

grub> _
```

注記 - GRUB はドライブのタイプに関係なく 0 からドライブ番号をカウントし、Integrated Drive Electronics (IDE) デバイスと Small Computer Systems Interface (SCSI) デバイスを区別しません。

x86: GRUB 2 と GRUB Legacy のタスクの比較

GRUB 2 は GRUB Legacy といくつかの特性を共有していますが、GRUB 2 は menu.lst ファイルを使用しないため、多くのブート管理タスクが GRUB 2 をサポートするシステムでは異なって実行されます。たとえば、bootadm コマンドの新しいサブコマンドを使用して、GRUB メニューを管理し、各種ブートローダー管理タスクを実行します。

ほとんどの bootadm サブコマンドで、新しい `-P pool` 引数を使用できます。このオプションにより、GRUB メニューや特定のルートプールのブートエントリを表示したり、変更したりすることができます。GRUB Legacy をサポートするオペレーティングシステムを実行している場合、これらの bootadm サブコマンドで、`-P` オプションを使用できない可能性があります。

たとえば、次のように、特定のルートプールの GRUB メニューを表示します。

```
# bootadm list-menu -P pool-name
```

次の表は、いくつかの一般的な GRUB 2 タスクおよびコマンドを GRUB Legacy の同等のものと比較しています。詳細な手順については、[bootadm\(1M\)](#) のマニュアルページおよび [30 ページ](#) の「[bootadm コマンドを使用して GRUB 構成を管理する](#)」を参照してください。

表 3 GRUB Legacy タスクと比較した GRUB 2 タスク

タスクまたはコマンド	GRUB 2 の方法	GRUB Legacy の方法
GRUB メニューの現在のブートエントリを一覧表示します。	<pre>bootadm list-menu</pre> <p>エントリ番号ごとまたはタイトルごとに個々のエントリを表示することもできます。タイトルごとにエントリを表示するには:</p> <pre>bootadm list-menu entry-title</pre> <p>タイトルにスペースが含まれる場合、タイトルが複数の引数として解釈されないように、引用符を使用する必要があります。例:</p> <pre>bootadm list-menu `This is a menu entry with a title`</pre> <p>エントリ番号ごとにエントリを表示するには:</p> <pre>bootadm list-menu -i entry-number</pre>	<pre>bootadm list-menu</pre>

タスクまたはコマンド	GRUB 2 の方法	GRUB Legacy の方法
<p>デフォルトのブートローダー設定と、システム上の各ルートプールの Oracle Solaris ブート環境ごとに 1 つのメニューエントリを格納する新しい GRUB 構成ファイル (<code>grub.cfg</code>) を生成します。</p>	<p><code>bootadm generate-menu</code></p> <p>システムに既存の <code>grub.cfg</code> ファイルがある場合、<code>generate-menu</code> サブコマンドに <code>-f</code> オプションを付けて使用します。この構文は既存の GRUB 2 構成を破棄し、新しい構成で置き換えます。</p> <p><code>-p</code> オプションを使用して、システム上の特定のルートプールの新しい GRUB 2 構成ファイルを生成する場合、生成された <code>grub.cfg</code> ファイルがそのルートプールの最上位 ZFS データセットに格納されることに注意してください。</p>	<p><code>menu.lst</code> ファイルを手動で編集して、新しい情報を追加します。</p>
<p>GRUB メニューに新しいエントリを追加します。</p>	<p>エントリ番号を指定してエントリを追加するには:</p> <pre>bootadm add-entry -i entry-number</pre> <p>タイトルを指定してエントリを追加するには:</p> <pre>bootadm add-entry entry-title</pre>	<p><code>menu.lst</code> ファイルを手動でエントリを追加します。</p>
<p>GRUB メニューのエントリを変更します。</p>	<p>エントリ番号を指定してエントリを変更するには:</p> <pre>bootadm change-entry -i entry-numberkey=value</pre> <p>タイトルを指定してエントリを変更するには:</p> <pre>bootadm change-entry entry-title key=value</pre> <p>タイトルにスペースが含まれる場合、タイトルが複数の引数として解釈されないように、引用符を使用する必要があります。</p> <p>このサブコマンドは、Oracle Solaris コンソールデバイスをカーネル引数として指定するなど、個々のブートエントリを変更するために使用します。エントリのタイトルが複数のメニューエントリと一致する場合、最初のエントリのみが変更されます。</p>	<p>持続的な変更をするには、<code>menu.lst</code> ファイルを手動で編集します。</p> <p>または、ブート時に GRUB メニューを編集して、システムが次回にブートするときまでにだけ持続するブートエントリの変更を行いません。</p>

タスクまたはコマンド	GRUB 2 の方法	GRUB Legacy の方法
	ブートエントリの変更は、以前のリリースで、GRUB Legacy エントリで実行していたように、ブート時に GRUB メニューを編集することによって実行することもできます。	
GRUB メニューからエントリを削除します。	エントリ番号を指定してエントリを削除するには: <code>bootadm remove-entry -i entry-number</code> タイトルを指定してエントリを削除するには: <code>bootadm remove-entry entry-title</code> タイトルを指定すると、そのタイトルのすべてのエントリが削除されます。	menu.lst ファイルから手でエントリを削除します。
GRUB メニューを管理します。たとえば、ブート元となるデフォルトの GRUB メニューエントリを設定します。	<code>bootadm set-menu key=value</code>	<code>bootadm set-menu</code>
Linux エントリなどのカスタムメニューエントリを GRUB メニューに追加します。	custom.cfg ファイルにエントリを追加し、正しい GRUB 2 構成ファイル構文を使用していることを確認してください。47 ページの「GRUB 構成のカスタマイズ」を参照してください。 注記 - まずこのファイルを作成する必要があります。	Oracle Solaris のインストール後に menu.lst ファイルに情報を追加します。
ブート引数を追加するには、ブート時に GRUB メニューを編集します。	1. ブートプロセスに割り込むには、矢印キーを使用して、目的のメニューエントリを選択してから、e と入力します。 2. 指定したブートエントリの \$multiboot 行の末尾にブート引数を追加します。 3. Control-X を押して、変更したエントリからブートします。システムコンソールがシリアルデバイス上にある場合、UEFI システムで、F10 キーが正しく認識されないことがあります。その場合、Ctrl-X を使用します。 注記 - メニューエントリの編集集中に Escape キーを押すとメニュー	1. ブートプロセスに割り込むには、e と入力します。 2. 指定したブートエントリの kernel\$ 行の末尾にブート引数を追加します。 3. Return キーを押してから b を入力してシステムをブートします。

タスクまたはコマンド	GRUB 2 の方法	GRUB Legacy の方法
	エントリリストに戻り、すべての編集内容が廃棄されます。	
ブートローダープログラムをインストールします。	<code>bootadm install-bootloader</code> このコマンドは、ミラー化ルートプールのすべてのデバイスに自動的にブートローダーをインストールします。	x86 ベースのシステムの場合 <code>installgrub</code> 、および SPARC ベースのシステムの場合 <code>installboot</code> 。
UEFI または BIOS ファームウェアのブートパーティションを作成します。	<code>zpool create</code> コマンドの新しい <code>-B</code> オプションを使用して、ファームウェアに適切なブートパーティションと新しい ZFS プールが格納される ZFS データパーティションを同時に自動的に作成します。 ルートプールにディスクを接続すると、適切なブートパーティションが自動的に作成され、そのディスクにブートローダーがインストールされます。『 Oracle Solaris 11.3 での ZFS ファイルシステムの管理 』の第 6 章、「 ZFS ルートプールの管理 」を参照してください。	GRUB Legacy は BIOS ファームウェアを搭載するシステムのみをサポートするため、個別のブートパーティションは必要ありません。

x86: bootadm コマンドを使用して GRUB 構成を管理する

このセクションの内容は次のとおりです。

- [32 ページの「GRUB メニューのパスワード保護」](#)
- [33 ページの「ユーザーへの GRUB メニューにアクセスする承認の付与」](#)
- [33 ページの「GRUB メニューまたは特定のメニューエントリの表示」](#)
- [34 ページの「GRUBメニューを手動で再生成する方法」](#)
- [35 ページの「GRUB メニューを変更する方法」](#)
- [38 ページの「GRUB メニューの指定したブートエントリの属性を設定する方法」](#)
- [41 ページの「GRUB メニューにブートエントリを追加する方法」](#)
- [42 ページの「GRUB メニューからブートエントリを削除する方法」](#)

GRUB Legacy をサポートするシステムでは、`menu.lst` ファイルを編集することによって、主に GRUB 構成と GRUB メニューを管理します。GRUB 2 をサポートするシステムでは、`grub.cfg` ファイルを使用します。ただし、このファイルを手動で編集する代わりに、ブート管理インタフェース `bootadm` を使用します。`bootadm` コマンド

を使用すると、以前は `menu.lst` ファイルを編集することによって行なっていたほとんどのタスクを管理できます。これらのタスクには、ブートローダー設定や GRUB メニューのほか、特定のブートエントリの個々の属性の管理が含まれます。

注記 - `bootadm` コマンドまたは `beadm` コマンドを使用してブートローダーに対して行われた変更によって `grub.cfg` ファイルが上書きされる場合があるため、このファイルを直接編集しないでください。

次の `bootadm` サブコマンドは、GRUB 2 構成の管理をサポートします。

<code>add-entry</code>	GRUB メニューにブートエントリを追加します。
<code>change-entry</code>	GRUB メニューの指定されたブートエントリの属性を変更します。
<code>generate-menu</code>	新しいブートローダー構成ファイルを生成します。
<code>install-bootloader</code>	システムブートローダーをインストールします。このサブコマンドは x86 プラットフォームと SPARC プラットフォームの両方に適用されます。
<code>list-menu</code>	GRUB メニューの現在のブートエントリを表示します。 -p オプションを指定すると、指定されたルートプールのブートエントリを表示できます。-i オプションは、インデックス番号によって識別される特定のメニューエントリに関する情報を表示します。-t オプションは、タイトルでメニューエントリを選択します。
<code>remove-entry</code>	GRUB メニューからブートエントリを削除します。
<code>set-menu</code>	GRUB メニューを保守します。このサブコマンドを使用すると、特定の GRUB メニューエントリをデフォルトとして設定したり、GRUB メニューにセキュリティー保護を追加したり、ほかのメニューオプションやブートローダーオプションを設定したりできます。 -p オプションを指定すると、複数のルートプールのメニューを変更できます。
<code>set-menu-password</code>	パスワードを設定して、GRUB メニューが表示されないようにします。
<code>show-entry</code>	GRUB メニューのブートエントリを表示します。このサブコマンドは <code>list-menu</code> と同等です。

注記 - SPARC プラットフォームでは GRUB を使用しないため、bootadm コマンドを使用したブートメニュー管理はありません。ただし、SPARC ベースのシステム上で 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たり、ブートアーカイブを手動で更新したり、ブートローダーをインストールしたりすることは可能です。123 ページの「[Oracle Solaris ブートアーカイブの管理](#)」を参照してください。

次の手順では、bootadm コマンドを使用して、GRUB 構成と GRUB メニューを管理する方法について説明します。詳細は、[bootadm\(1M\)](#)のマニュアルページを参照してください。

x86: GRUB メニューのパスワード保護

GRUB メニューは以前、コンソールに物理的にアクセスできるすべてのユーザーに解放されていました。Oracle Solaris 11.3 リリースでの変更により、パスワードを入力しなければ表示できないよう GRUB メニューをロックする機能が追加されました。

パスワードロックを設定することによって、GRUB メニュー全体へのアクセスを制御できます。コンソールにアクセスでき、パスワードを知っているすべてのユーザーが GRUB メニュー内のエントリを表示できます。パスワードロックを管理するための bootadm set-menu-password コマンドのオプションは次のとおりです。

- s GRUB メニュー内の任意のエントリの表示、編集、またはブートに必要なパスワードを設定します
- r GRUB メニューへのアクセスに必要なパスワードを削除します
- l パスワードロックが設定されているかどうか、および各メニューエントリにどのユーザーがアクセスできるかを一覧表示します

さらに、個々のユーザーに GRUB メニュー内のすべてのエントリを表示、編集、またはブートする機能を与えたり、ユーザーごとにアクセス可能なエントリを指定したりできます。詳細は、33 ページの「[ユーザーへの GRUB メニューにアクセスする承認の付与](#)」を参照してください。



注意 - GRUB メニュー全体に対してパスワードロックを設定した場合は、システムがブートする前に、ユーザーがパスワードを入力する必要があります。

x86: ユーザーへの GRUB メニューにアクセスする承認の付与

特定のユーザーに GRUB メニュー全体へのアクセスや、GRUB メニュー内の特定のエン트리へのアクセスの承認を与えることができます。どちらの状況でも、`bootadm set-menu adduser=username` コマンドを使用して、そのユーザーを認証されたユーザーのリストに追加する必要があります。GRUB メニューへのアクセスを認証するために使用するパスワードは、ブート時に OS で使用するパスワードとは異なります。

承認されたユーザーのリストに名前を追加したら、次を実行できます。

- ユーザーを GRUB メニュースーパーユーザーのリストに追加することにより、ユーザーはすべてのエントリにアクセスできる
- ユーザーがアクセスできる特定のエントリを定義する

個々のユーザーに GRUB 内のすべてのエントリを表示または編集する機能を与えるには、`bootadm set-menu add-superuser username` コマンドを使用して、ユーザー名を GRUB メニュースーパーユーザーのリストに追加します。GRUB メニューのプロンプトにユーザー名とパスワードを入力すると、ユーザーはメニュー内のすべてのエントリを表示、編集、またはブートできます。

ユーザーに GRUB メニュー内の特定のエントリをブートおよび編集する承認を与えるには、`change-entry` サブコマンドを使用して、ユーザーがインデックス番号またはタイトルでどのエントリにアクセスできるかを選択します。たとえば、`bootadm set-menu change-entry -i 3 add-auth=username` コマンドは、指定されたユーザーにインデックスエントリ 3 のメニュー項目を編集する機能を与えます。



注意 - デフォルトのブートエントリがロックされている場合は、システムがブートする前に、ユーザーがパスワードを入力する必要があります。システムで手動での介入なしでリブートできることが重要な場合は、デフォルトのエントリがパスワードでロックされていないことを確認してください。

x86: GRUB メニューまたは特定のメニューエントリの表示

現在システム上にある GRUB メニューエントリを一覧表示するには、`bootadm` コマンドの `list-menu` サブコマンドを使用します。この情報は `grub.cfg` ファイルによって提供されます。特定のメニューエントリに関する情報を表示するには、`list-menu` または `show-entry` サブコマンドを使用します。`-i` オプションを使用してエントリ番号で、または `-t` オプションを使用してタイトルで特定のエントリを選択できます。

例 1 GRUB メニュー内のすべての項目の一覧表示

`list-menu` サブコマンドを使用すると、現在の GRUB メニューの内容を表示できます。このサブコマンドはまた、ブートローダー構成ファイルの場所、デフォルトのブートエントリ番号、`autoboot-timeout` 値、および各ブートエントリのインデックス番号とタイトルも表示します。

```
$ bootadm list-menu
the location of the boot loader configuration files is: /rpool/boot/grub
default 0
console graphics
timeout 30
0 Oracle Solaris 11 FCS
1 Oracle Solaris backup-1
2 Oracle Solaris 11 11.2
```

例 2 特定のエントリ番号に関する情報の一覧表示

`list-menu` サブコマンドの実行時に `-i` オプションでエントリ番号を指定した場合は、選択されたエントリに関する情報が出力に表示されます。

```
$ bootadm list-menu -i 0
the location of the boot loader configuration files is: /rpool/boot/grub
title: Oracle Solaris 11 FCS
kernel: /platform/i86pc/kernel/$ISADIR/unix
kernel arguments: -B $ZFS-BOOTFS -v
boot archive: /platform/i86pc/$ISADIR/boot_archive
ZFS root pool: rpool
```

`show-entry` サブコマンドを `-i` オプションとともに使用した場合は、ブートローダー構成ファイルの場所を除き、同じ情報が表示されます。

```
$ bootadm show-entry -i 0
title: Oracle Solaris 11 FCS
kernel: /platform/i86pc/kernel/$ISADIR/unix
kernel arguments: -B $ZFS-BOOTFS -v
boot archive: /platform/i86pc/$ISADIR/boot_archive
ZFS root pool: rpool
```

▼ x86: GRUBメニューを手動で再生成する方法

現在システムにインストールされている OS インスタンスを格納する `grub.cfg` ファイルを手動で再生成するには、`bootadm generate-menu` コマンドを使用します。

`/usr/lib/grub2/bios/etc/default/grub` または `/usr/lib/grub2/uefi64/etc/default/grub` ファイルからの情報と GRUB メタ構成ファイル `rpool/boot/grub/menu.conf` からの情報を組み合わせて使用して、最終的な `grub.cfg` ファイルが生成されます。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. grub.cfg ファイルを生成します。

```
# bootadm generate-menu
```

- grub.cfg ファイルがすでに存在する場合、-f オプションを使用して、既存のファイルを上書きします。

```
# bootadm generate-menu -f
```

- 次のように、現在のルートプール以外のルートプールの新しい GRUB メニューを生成します。

```
# bootadm generate-menu -P pool-name
```

3. メニューが変更を反映して更新されていることを確認してください。

```
# bootadm list-menu
```

注記 - 変更が表示されない場合は、grub.cfg ファイルを調べて、変更が実行されているか確認します。

▼ x86: GRUB メニューを変更する方法

GRUB メニューを保守するには、bootadm コマンドの set-menu サブコマンドを使用します。たとえば、このコマンドを使用して、メニューのタイムアウトや GRUB メニューのデフォルトのブートエントリを変更できます。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. (オプション) GRUB メニューエントリを一覧表示します。

```
# bootadm list-menu
```

3. GRUB メニューに必要な変更を行ないます。

```
# bootadm set-menu [-P pool] [-R alroot [-p platform]] key=value
```

set-menu サブコマンドを使用して指定できる各値の詳細については、bootadm(1M)のマニュアルページを参照してください。set-menu サブコマンドを使用できる一般的な方法の例では、この手順に従います。

4. 変更が行なわれたことを確認します。

```
# bootadm list-menu
```

注記 - 変更が表示されない場合は、`grub.cfg` ファイルを調べて、変更が実行されているか確認します。

例 3 GRUB メニューのデフォルトのブートエントリを変更する

GRUB メニューでデフォルトのエントリ番号 (0、1、2 など) を設定するには、適切な `key=value` オプションを付けて `bootadm set-menu` コマンドを使用します。この番号は、タイマーの期限が切れたときにブートされるオペレーティングシステムを指定します。

次の例では、デフォルトのブートエントリを 2 (Oracle Solaris 11.3) に設定します。

```
# bootadm set-menu default=1
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 1
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test
```

この例では、デフォルトのメニューエントリが現在 1 になっています。システムはリブートすると、デフォルトのタイマーの期限が切れたあとに、新しい Oracle Solaris エントリを自動的にブートします。

`change-entry` サブコマンドを使用して、GRUB メニューにデフォルトのエントリを設定することもできます。[38 ページの「GRUB メニューの指定したブートエントリの属性を設定する方法」](#)を参照してください。

例 4 GRUB メニューのメニュータイムアウト値を変更する

メニュータイムアウト値を設定するには、適切な `key=value` オプションを付けて `bootadm set-menu` コマンドを使用します。

次の例では、`bootadm list-menu` コマンドの出力で、45 秒に変更された 30 秒のデフォルトのタイムアウト値を示しています。変更は、次回にシステムをブートしたときに有効になります。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
```

```

2 Oracle Solaris 11_test
# bootadm set-menu timeout=45
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 45
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test

```

例 5 GRUB コンソールタイプを設定する

bootadm コマンドの set-menu サブコマンドを使用して設定できる 1 つの値は、コンソールタイプです。この方法でのコンソールタイプの変更は、システムのリブートをまたがって維持されます。

たとえば、次のように、grub.cfg ファイルで、コンソールタイプを serial に設定します。

```
# bootadm set-menu console=serial
```

標準テキストコンソールの場合、コンソールタイプを text に設定することもできます。BIOS シリアルリダイレクトを使用している場合、このオプションを選択します。またはコンソールタイプを graphics に設定できます。このオプションは追加のグラフィカルメニューを提供し、背景画像が使われます。

コンソールタイプを serial に設定すると、ブート時のシリアルポートの初期化時に GRUB 2 が使用するシリアルパラメータを構成できます。serial_params 値を指定しないと、デフォルトは、シリアルポート 0 (COM1/ttya) を使用し、速度を指定しません。速度を指定しないで、ポートのみを指定した場合 (serial_params=0 など)、使われる速度は不定で、GRUB が実行される前にシリアルポートが初期化された速度になります。特定の速度が使われるようにする場合は、serial_params で明示的に設定する必要があります。

次のように、serial_params キー値を bootadm コマンド行に追加します。

<i>port</i>	ポート番号です。ポート ttya から ttyd または COM1 から COM4 をそれぞれ指定するには、0 から 3 の任意の数を使用できます (通常、ttya または COM1 には 0 が使われます)。
<i>speed</i>	シリアルポートで使用する速度です。この値を省略した場合、GRUB 2 はシリアルポートで使用するよう初期化された速度を使用します。シリアルポートが初期化されていない場合は、速度を指定しないと、予測できない出力になることがあります。シリアルポートが初期化されているかどうかわからない場合、また BIOS コンソールリダイレクトを使用しない場合は、速度値を指定することをお勧めします。
<i>data bits</i>	7 または 8 の値で指定します。

`parity` e、o、n (even、odd、none) としてそれぞれ指定します。

`stop bits` 0 または 1 の値で指定します。

`port` パラメータを除くすべてのシリアルパラメータはオプションです。

▼ x86: GRUB メニューの指定したブートエントリの属性を設定する方法

`bootadm` コマンドの `change-entry` サブコマンドを使用して、GRUB メニューの指定したブートエントリ、またはエントリのカンマ区切りのリストの特定のブート属性を設定します。エントリは、エントリタイトルまたはエントリ番号で指定します。複数のエントリが同じタイトルの場合、すべてのエントリが影響を受けます。

注記 - 特殊なプロパティ `set-default` は、タイマーの期限が切れたときに、ブートするデフォルトのエントリを設定します。このサブコマンドは、`set-menu default=value` サブコマンドと同じように機能します。例3「GRUB メニューのデフォルトのブートエントリを変更する」を参照してください。

ブート時に、GRUB メニューを編集して、特定のブートエントリの属性を設定する方法については、43 ページの「ブート時に GRUB メニューを編集してカーネル引数を追加する」を参照してください。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. (オプション) GRUB メニューエントリを一覧表示します。

```
# bootadm list-menu
```

3. 指定したエントリのブート属性を設定します。

```
# bootadm change-entry [-P pool] {[entry-title[,entry-title...]]}
| -i entry-number[,entry-number...] { key=value [ key=value ...]
| set-default }
```

空白を含む値を指定する場合は、値を引用符または二重引用符で囲む必要があります。

`change-entry` サブコマンドを使用して指定できる各値の詳細については、`bootadm(1M)`のマニュアルページを参照してください。`change-entry` サブコマンドを使用する一般的な方法の例については、次の手順に従います。

4. 指定したエントリに変更が行なわれたことを確認します。

```
# bootadm list-menu
```

注記 - 変更が表示されない場合は、grub.cfg ファイルを調べて、変更が実行されているか確認します。

例 6 GRUB メニューの指定したブートエントリのタイトルを設定する

bootadm コマンドの change-entry サブコマンドを使用して、指定したブートエントリのタイトルを設定できます。タイトルを設定する場合、エントリ番号またはエントリタイトルのいずれかを指定できます。次の例に、指定したブートエントリに両方の方法でタイトルを設定する方法を示します。複数のエントリが同じタイトルの場合、すべてのエントリが影響を受けます。

次のように、エントリ番号を指定してブートエントリのタイトルを設定します。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 1
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test
# bootadm change-entry -i 2 title="Oracle Solaris 11-backup1"
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 45
0 Oracle Solaris 11/11
1 Oracle Solaris 11.32 Oracle Solaris 11-backup1
```

次のように、タイトルを指定してブートエントリのタイトルを設定します。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 1
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.32 Oracle Solaris 11_test
# bootadm change-entry "Oracle Solaris 11_test" title="Oracle Solaris 11-backup1"
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 45
0 Oracle Solaris 11/11
1 Oracle Solaris 11.32 Oracle Solaris 11-backup1
```

例 7 カーネル引数を指定してブートエントリを変更する

次の例に、bootadm コマンドの change-entry サブコマンドを使用して、指定したブートエントリのカーネルブート引数を設定する方法を示します。

この例では、シングルユーザーモードでのブートにエントリ番号 1 が設定されています。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 1
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test
# bootadm change-entry -i 1 kargs=-s
# bootadm list-menu -i 1
The location of the boot loader configuration files is: /rpool/boot/grub
  title: Oracle Solaris 11.3
  kernel: /platform/i86pc/kernel/$ISADIR/unix
  kernel arguments: -s
  boot archive: /platform/i86pc/$ISADIR/boot_archive
  ZFS root pool: rpool
```

この例では、複数のカーネル引数がブートエントリ番号 2 に指定されています。

```
# bootadm change-entry -i 2 kargs="-v -s"
# bootadm list-menu -i 2
The location of the boot loader configuration files is: /rpool/boot/grub
  title: Oracle Solaris 11_test
  kernel: /platform/i86pc/kernel/$ISADIR/unix
  kernel arguments: -v -s
  boot archive: /platform/i86pc/$ISADIR/boot_archive
  bootfs: rpool/ROOT/snv_160-nightly-1
```

この例では、`-v` オプションと `-s` オプションが指定されており、システムを冗長モードのシングルユーザー状態にブートします。

空白を含む属性 (または複数の属性) を設定する場合は常に、値を引用符または二重引用符で囲む必要があります。

例 8 -B オプションを使用してカーネル引数を指定してブートエントリを変更する

次の例に、`-B` オプションを使用して、特定のブートエントリのカーネル引数を設定する方法をいくつか示します。

次のように、ブート時に `e1000g` ネットワークドライバを無効にし、カーネルデバッグをロードします。

```
# bootadm change-entry -i 0 kargs="-B disable-e1000g=true -k"
```

`bootadm change-entry` コマンドを使用して、複数の `-B` オプションを指定できます。たとえば、次のいずれかのコマンドを使用して、`e1000g` ドライバおよび `ACPI` を同時に無効にします。

```
# bootadm change-entry -i 0 kargs="-B disable-e1000g=true -B acpi-user-options=2"
```

```
# bootadm change-entry -i 0 kargs="-B disable-e1000g=true,acpi-user-options=2"
```

`-B` オプションを使用して、指定したブートエントリを編集して、ブート時に特定のブート属性を設定することもできます。手順については、43 ページの「ブート時に GRUB メニューを編集してカーネル引数を追加する」を参照してください。

例 9 以前に追加されたカーネル引数をブートエントリから削除する

次の例では、特定のブートエントリからカーネル引数 (-s) を削除します。

```
# bootadm list-menu -i 1
the location of the boot loader configuration files is: /rpool/boot/grub
title: s11.3.backup
kernel: /platform/i86pc/kernel/amd64/unix
kernel arguments: -s
boot archive: /platform/i86pc/amd64/boot_archive
bootfs: rpool/ROOT/s11.3.backup
# bootadm change-entry -i 1 kargs=
# bootadm list-menu -i 1
the location of the boot loader configuration files is: /rpool/boot/grub
title: s11.3.backup
kernel: /platform/i86pc/kernel/amd64/unix
kernel arguments:
boot archive: /platform/i86pc/amd64/boot_archive
bootfs: rpool/ROOT/s11.3.backup
```

▼ x86: GRUB メニューにブートエントリを追加する方法

指定したエントリタイトルで、新しいエントリを GRUB メニューに追加するには、bootadm コマンドの add-entry サブコマンドを使用します。エントリ番号を指定すると、新しいエントリが GRUB メニューの特定の位置に挿入されます。または、エントリ番号がメニュー内の現在のエントリ数よりも大きい場合、エントリはメニューの最後のエントリとして追加されます。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. (オプション) GRUB メニューの現在のブートエントリを一覧表示します。

```
# bootadm list-menu
```

3. GRUB メニューに新しいブートエントリを追加します。

```
# bootadm add-entry -P pool -i [entry-number] entry-title
```

4. 次のように、新しく追加されたエントリに bootfs プロパティを設定します。

```
# bootadm change-entry -i new-entry-number bootfs='pool-name/ROOT/be-name'
```

この段階により、新しく追加されたブートエントリが、bootfs pool-level プロパティに指定されており、ルートプールに設定されているデフォルトの bootfs 値を使用しなくなります。

5. ブートエントリが追加されたことを確認します。

```
# bootadm list-menu
```

注記 - 変更が表示されない場合は、`grub.cfg` ファイルを調べて、変更が実行されているか確認します。

例 10 x86: GRUB メニューにブートエントリを追加する

次の例に、`bootadm add-entry` コマンドを使用して、GRUB メニューにメニューエントリを追加する方法を示します。この例では、エントリ番号 2 が追加されます。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
# bootadm add-entry -i 2 Oracle Solaris 11_test
# bootadm change-entry -i 2 bootfs='rpool/ROOT/test'
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test
```

次のように、エントリ番号を指定して、新しいメニューエントリの内容を表示します。

```
# bootadm list-menu -i 2
the location of the boot loader configuration files is: /rpool/boot/grub
title: Oracle Solaris 11_test
kernel: /platform/i86pc/kernel/amd64/unix
kernel arguments: -B $ZFS-BOOTFS
boot archive: /platform/i86pc/amd64/boot_archive
ZFS root pool: rpool
```

▼ x86: GRUB メニューからブートエントリを削除する方法

GRUB メニューから、指定したエントリ、またはエントリのカンマ区切りのリストを削除するには、`bootadm` コマンドの `remove-entry` サブコマンドを使用します。同じタイトルの複数のエントリを指定すると、そのタイトルのすべてのエントリが削除されます。

1. `root` 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. (オプション) 現在のブートエントリを一覧表示します。

```
# bootadm list-menu
```

3. GRUB メニューから指定したエントリを削除します。

```
# bootadm remove-entry [-P pool] [{entry-title [,entry-title...]} |
-i entry-number[,entry-number...]]
```

4. エントリが削除されていることを確認します。

```
# bootadm list-menu
```

注記 - 変更が表示されない場合は、grub.cfg ファイルを調べて、変更が実行されているか確認します。

例 11 x86: GRUB メニューからブートエントリを削除する

次の例では、GRUB メニューからのエントリ番号 2 の削除を示しています。

```
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
2 Oracle Solaris 11_test
bootadm remove-entry -i 2
1 entry removed
# bootadm list-menu
The location of the boot loader configuration file is /rpool/boot/grub
default 2
console graphics
timeout 30
0 Oracle Solaris 11/11
1 Oracle Solaris 11.3
```

x86: ブート時に GRUB メニューを編集してカーネル引数を追加する

x86 プラットフォームでは、ブート時に GRUB メニューを編集して、特定のブートエントリのブート属性およびカーネル引数を設定できます。これらの変更は次回にシステムをブートするときまで持続します。

特定のブートエントリのブート属性を永続的に設定するには、`bootadm` コマンドと `change-entry` サブコマンドを使用します。38 ページの「[GRUB メニューの指定したブートエントリの属性を設定する方法](#)」を参照してください。

x86 ベースのシステムをブートすると、GRUB メインメニューが表示されます。このメニューには、現在システム上にあるすべてのブートエントリのリストが含まれます。特定のブートエントリを編集するには、矢印キーを使用してエントリを選択し、`e` と入力して、エントリを編集します。GRUB 編集画面で、`$multiboot` 行に移動し、行の末尾に追加のブートオプションやカーネル引数を入力します。

GRUB 編集メニューの `$multiboot` 行は次のようになります。

```
$multiboot /ROOT/transition/@@/$kern $kern -B console=graphics -B $zfs_bootfs
```

たとえば、ブート時に `e1000g` ネットワークドライバを無効にし、`kmdb` をロードするには、次のように、指定したエントリの GRUB メニューを編集します。

```
$multiboot /ROOT/solaris/@@/$kern $kern -B disable-e1000g=true -k -B $zfs_bootfs
```

GRUB 編集メニューを終了し、編集したばかりのエントリをブートするには、`Control-X` を押します。UEFI ファームウェアを搭載するシステムがあり、シリアルコンソールを使用していない場合、`F10` キーを押すことによってもエントリがブートします。

注記 - ブート時に GRUB メニューを編集する予定がある場合、ブートシーケンス時に GRUB メニューが表示されるようにするため、`reboot` コマンドの `-p` オプションを使用して、システムをリブートする必要があります。

ブート時に GRUB メニューを編集する場合、次のカーネル引数とオプションを指定できます。

<code>unix</code>	ブートするカーネルを指定します。
<code>-a</code>	構成情報の入力を求めるプロンプトを表示します。
<code>-i altinit</code>	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。 <code>altinit</code> は実行可能ファイルへの有効なパスです。
<code>-k</code>	カーネルデバグを有効にした状態でシステムをブートします
<code>-m smf-options</code>	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
<code>-r</code>	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。

- s システムをシングルユーザー状態にブートします。
- 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。

注記 - eeprom ユーティリティを使用し、さらに GRUB コマンド行で、パラメータを指定すると、GRUB コマンド行が優先されます。

詳細については、[kernel\(1M\)](#)のマニュアルページを参照してください。

x86: GRUB メニューを編集して、ブート時に `-B prop=val` カーネル引数を追加する

ブート時に特定のカーネル引数を指定できます。たとえば、`-B prop=val` オプションを指定して、Oracle Solaris システムコンソールを設定できます。次に、指定したブートエントリに `-B prop=val` オプションを追加して、ブート時に x86 プラットフォームでブートパラメータを変更できるさまざまな方法を示します。

- `-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。[46 ページの「ブート時に Oracle Solaris コンソールをリダイレクトする」](#)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。
- `-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。

-B Oracle Solaris で UEFI 実行時サービスの使用を無効にします。
uefirt_disable=1

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

例 12 Oracle Solaris システム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 `console=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。

コンソールに `console=force-text` プロパティを設定しても、UEFI ファームウェアを搭載するシステムでは、VGA アダプタはテキストモードに移行しません。

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および `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 13 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
```

ブート時に Oracle Solaris コンソールをリダイレクトする

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、UEFI ファームウェアと従来の BIOS ファームウェアおよび 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。

<pre>-B console=force- text</pre>	<p>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。</p>
<pre>-B screen- #columns=value, screen- #rows=value</pre>	<p>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。</p>

Oracle Solaris ブートエントリは、特定の一連のグラフィックモードを特定の順番で試みます。これらのモードは、`grub.cfg` ファイルの `$multiboot` 行に続く `set gfxpayload` 行に一覧表示されます。一覧表示されていないモードを希望する場合は、この行を変更できます。この変更を永続的にするには、エントリを `custom.cfg` ファイルにコピーする必要があります。そうしないと、次回に `grub.cfg` ファイルが自動生成されたときに、`gfxpayload` 設定が上書きされます。

`set gfxpayload` 引数の構文は次のようになります。

```
WidthxHeight[xbit-depth]
```

「x」は実際の文字です。例:

```
set gfxpayload=1024x768;1280x1024x32
```

この設定は GRUB がまず任意のビット深度 (高いビット深度を推奨) で、1024x768 モードの検索を試み、次に 32 ビット深度で 1280x1024 の検索を試みることを意味します。特殊なキーワード `text` はテキストモードを選択します。このキーワードは UEFI ファームウェアで機能しない場合があることに注意してください。keep キーワードは、グラフィカルコンソールタイプが使用されている場合に、GRUB が使用しているモードを維持し、そのモードを Oracle Solaris がフレームバッファコンソール解像度として使用することを指定します。

x86: GRUB 構成のカスタマイズ

`grub.cfg` ファイルにはほとんどの GRUB 構成が格納されます。GRUB 構成にメニューエントリやその他のスクリプトなどの、複雑な構造を追加したい場合

は、`custom.cfg` という追加の編集可能ファイルを使用できます。このファイルはデフォルトでシステムに存在しません。このファイルを作成し、`/pool-name/boot/grub/`にある `grub.cfg` ファイルと `menu.conf` ファイルと同じ場所に置く必要があります。

GRUB は、`grub.cfg` ファイルの末尾にある次のコードによって、`custom.cfg` ファイルにあるコマンドとカスタマイズを処理します。

```
if [ -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
fi
```

これらの命令では、GRUB に、`boot/grub` サブディレクトリ内で、ルートプールの最上位データセット内の `custom.cfg` ファイルの存在をチェックするように指示します。`custom.cfg` ファイルが存在する場合、GRUB はそのファイルをソースとし、内容がテキストで `grub.cfg` ファイルに挿入されたかのように、ファイル内にあるすべてのコマンドを処理します。

64 ビット UEFI ファームウェアを搭載するシステムで、このファイルのエントリは、次のようになります。

```
menuentry "Windows (64-bit UEFI)" {
    insmod part_gpt
    insmod fat
    insmod search_fs_uuid
    insmod chain
    search --fs-uuid --no-floppy --set=root cafe-f4ee
    chainloader /efi/Microsoft/Boot/bootmgfw.efi
}
```

BIOS ファームウェアを搭載するシステムで、このファイルのエントリは、次のようになります。

```
menuentry "Windows" {
    insmod chain
    set root=(hd0,msdos1)
    chainloader --force +1
}
```

x86: GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする

このセクションでは、次の情報について説明しています。

- [49 ページの「GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする方法」](#)
- [52 ページの「GRUB Legacy メニューエントリが GRUB 2 に移行される方法」](#)
- [53 ページの「GRUB 2 と GRUB Legacy のブート環境を同じシステムで保守する」](#)

▼ x86: GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする方法

デフォルトのブートローダーとして GRUB 2 をサポートする Oracle Solaris リリースの新規インストールの場合、インストールを実行する前に必要なものは何もありません。

Oracle Solaris 11.1 以上へのアップグレードの場合、アップグレードの前いくつかの前提条件となるパッケージをインストールする必要があります。これらのパッケージは Oracle Solaris パッケージリポジトリに含まれます。

始める前に システムを GRUB 2 をサポートするリリースにアップグレードする前に、次を実行します。

- インストールやアップグレードに影響する可能性のある既知の問題を確認してください。『Oracle Solaris 11.3 ご使用にあたって』を参照してください。
- 52 ページの「GRUB Legacy メニューエントリが GRUB 2 に移行される方法」および 53 ページの「GRUB 2 と GRUB Legacy のブート環境を同じシステムで保守する」の情報とガイドラインを確認してください。
- 既存の GRUB Legacy 構成を維持します。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 前提条件となるパッケージをインストールします。

```
$ pkg update
```

3. 手順 2 で作成したばかりの新しいブート環境にシステムをリポートします。

4. システムが新しいブート環境で実行したら、次のコマンドを実行して、アップグレードを完了するために必要な修正プログラムで pkg パッケージを更新します。

```
$ pkg update pkg
```

このコマンドを実行すると、*pkg に一致する名前を持つすべてのパッケージが更新されます。これは pkg コマンドとその依存関係を含むパッケージです。

5. Oracle Solaris 11.3 へのアップグレードを完了するには、もう 1 回、次のように、pkg update コマンドを実行します。

```
$ pkg update --accept
```

注記 ---accept オプションを指定して、表示されているパッケージのライセンスの条件に同意し、受け入れることを示す必要があります。

最後の更新で、デフォルトのシステムブートローダーとして GRUB 2 がインストールされます。更新によって、GRUB Legacy menu.lst ファイルの内容に基づく grub.cfg ファイルも作成されます。

新しいブート環境がアクティブになると、GRUB Legacy 構成が GRUB 2 に移行され、GRUB 2 がシステムのデフォルトのブートローダーになります。menu.lst ファイル

から Oracle Solaris ブートエントリが、表示されている順番で `grub.cfg` ファイルにコピーされます。すべてのチェーンローダーエントリも移行されます。

```
GNU GRUB version 1.99,5.11.0.175.1.0.0.13.18988

Oracle Solaris 11.1 X86
Legacy GRUB Menu (from root pool rpool)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
```

```
GNU GRUB version 1.99,5.11.0.175.1.0.0.13.18988

Oracle Solaris 11 11/11

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line. ESC to return
previous menu.
```

x86: GRUB Legacy メニューエントリが GRUB 2 に移行される方法

GRUB 2 をサポートする Oracle Solaris のバージョンへのアップグレード後、すべての Oracle Solaris メニューエントリが GRUB Legacy `menu.lst` ファイルから新しい `grub.cfg` ファイルに自動的に移行されます。すべてのチェーンローダーエントリも移行されます。システムのリブート時に、移行されたブートエントリのみがメイン GRUB メニューに表示されます。メイン GRUB メニューに表示したいほかのブートエントリを手動で変換し、`custom.cfg` ファイルに追加する必要があります。47 ページの「GRUB 構成のカスタマイズ」を参照してください。

注記 - `menu.lst` ファイルのすべてのブートエントリが、そのルートプールの GRUB Legacy サブメニューに存在します。

重要な点として、GRUB 2 では、Solaris 10 1/06 リリース以降、Oracle Solaris 10 リリースに加えて、Oracle Solaris 11 のサポートされるすべてのリリースを直接ブートできます。以前の Oracle Solaris リリースはチェーンロードメカニズムを使用して、間接的にブートできます。他のカスタムエントリを追加する同じ方法で、チェーンロードを使用するメニューエントリを `custom.cfg` ファイルに追加できます。

GRUB Legacy の場合と同様に、チェーンロードの原則は GRUB 2 でも同じですが、構文はやや異なります。次の例で、エントリはディスク 0 上のマスターブートレコード (MBR) にチェーンロードされます。このタイプのチェーンロードは、GRUB 2 がその場所にインストールされていない場合のみ役立ちます。このようなチェーンロードは BIOS ファームウェアを搭載するシステム (すべての Oracle Solaris 10 システムを含む) でのみ機能します。

```
menuentry "Boot from Hard Disk" {
    set root=(hd0)
    chainloader --force +1
}
```

次の例では、2 番目の DOS パーティションに Oracle Solaris 10 がインストールされています。さらに、GRUB Legacy の Oracle Solaris 10 バージョンがそのパーティションのパーティションブートレコード (PBR) にインストールされます。

```
menuentry "Solaris 10" {
    set root=(hd0,msdos2)
    chainloader --force +1
}
```

この例では、エントリが Oracle Solaris 10 GRUB Legacy メニューにチェーンロードされます。結果として 2 つのメニューのレベルが存在します: 1 つは GRUB 2 から Oracle Solaris 10 GRUB Legacy メニューにチェーンロードし、1 つは Oracle Solaris 10 GRUB Legacy メニューから Oracle Solaris 10 カーネルをブートします。システムをブートするには、該当する Oracle Solaris 10 メニューエントリを選択する必要があります。

menu.lst ファイルから変換された Oracle Solaris メニューエントリに加えて、ルートプールごとに、GRUB Legacy menu.lst ファイルを格納する 1 つのサブメニューがあります。このサブメニューには、各 menu.lst ファイル内のすべてのメニューエントリが含まれ、最大の下位互換性のため、すべての menu.lst エントリにアクセスできます。

GRUB 2 の前提条件のパッケージを含まない Oracle Solaris ブート環境に戻すと、beadm コマンドや bootadm コマンドを使用して実行された変更など、ブート構成の変更のみが該当するルートプールの menu.lst ファイルに対して行われます。次にシステムをリブートしても、GRUB 2 メニューにそれらの変更は反映されません。該当するルートプールの Legacy GRUB サブメニューのみに変更が反映されます。

さらに、GRUB 2 対応のブート環境がブートされ、grub.cfg ファイルが再生成されるまで、これらの変更は main GRUB メニューに表示されません。システムで GRUB 2 を使用するブート環境が実行されると、可能であれば常に、menu.lst ファイルが grub.cfg ファイルと同期されます。この同期は、beadm または bootadm コマンドを使用して、GRUB 2 構成が変更されると常に実行されます。

x86: GRUB 2 と GRUB Legacy のブート環境を同じシステムで保守する

GRUB Legacy ブート環境のあるシステムで、GRUB 2 ブート環境をアクティブ化できますが、GRUB Legacy ブート環境が GRUB 2 対応である場合にのみ限ります。さらに、GRUB 2 ブート環境から GRUB Legacy ブート環境をアクティブ化できます。GRUB Legacy ブート環境のあるシステムで GRUB 2 ブート環境をアクティブ化する場合の注意の 1 つとして、pkg update コマンドを呼び出して GRUB 2 をサポートする Oracle Solaris リリースをインストールする前に、現在のブート環境に GRUB 2 の前提条件のパッケージをインストールしておく必要があります。49 ページの「[GRUB Legacy システムを GRUB 2 をサポートするリリースにアップグレードする方法](#)」を参照してください。

ブート環境は、beadm コマンドによって管理します。beadm(1M)を参照してください。beadm create コマンドを使用して、新しいブート環境を作成すると、そのブート環境のメニューエントリも自動的に作成されます。beadm list コマンドを使用して、システム上にあるすべてのブート環境を表示できます。

```
$ beadm list
BE                Active Mountpoint Space Policy Created
--                -
oracle-solaris11-backup - - 64.0K static 2014-03-29 11:41
oracle-solaris2    - - 64.0K static 2014-03-29 11:41
solaris11.3        NR / 3.35G static 2015-05-17 13:22
```

beadm コマンドは GRUB 2 構成と GRUB Legacy 構成の両方で機能します。ブート環境のリストに GRUB 2 ブート環境が存在する場合、GRUB 2 がデフォルトのブート

ローダーとして維持されます。Oracle Solaris は、GRUB Legacy ブート環境がアクティブ化されても、GRUB Legacy をデフォルトのブートローダーとして再インストールしようとしません。最後の GRUB 2 ブート環境をシステムから削除した場合、GRUB Legacy をシステムブートローダーとして手動でインストールする必要があります。システムに GRUB 2 の前提条件のパッケージが含まれている場合、`bootadm install-bootloader -f` コマンドを使用して、ブートローダーを手動でインストールできます。55 ページの「[bootadm install-bootloader コマンドを使用して GRUB 2 をインストールする](#)」を参照してください。それ以外の場合は、`installgrub` コマンドを使用できます。[installgrub\(1M\)](#) を参照してください。

`bootadm install-bootloader -f` コマンドを使用して、GRUB Legacy をデフォルトのブートローダーとして手動で再インストールすると、GRUB Legacy がシステムブートローダーとして強制的にインストールされます。すべてのブート環境をブート可能のままにするため、このコマンドは、最新の GRUB Legacy ブートローダーバージョンを格納するブート環境から実行する必要があります。さらに、GRUB Legacy を再インストールする前に、`beadm` コマンドを使用して、すべての GRUB 2 ブート環境をシステムから削除してください。56 ページの「[GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする方法](#)」を参照してください。

注記 - 古いブートローダーのあるシステムで、`-f` オプションを指定して `bootadm install-bootloader` コマンドを使用する場合は、古いブートローダーで、ブートディスク上の ZFS バージョンを読み取ることができなければなりません。そうでない場合、GRUB はブート時にルートプールを読み取ることができず、システムがブート不能になることがあります。

この状況が発生した場合、別のブート環境からブートするか、回復メディアからブートして、プールバージョンに一致するブートローダーバージョンをインストールすることによって、新しいブートローダーをインストールする必要があります。133 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 GRUB 構成の問題を解決する方法](#)」を参照してください。

x86: 高度な GRUB 管理とトラブルシューティング

このセクションでは、次の情報について説明しています。

- 55 ページの「[bootadm install-bootloader コマンドを使用して GRUB 2 をインストールする](#)」
- 56 ページの「[GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする方法](#)」

x86: bootadm install-bootloader コマンドを使用して GRUB 2 をインストールする

GRUB 2 ブートローダーが破損し、システムがブートできなくなった場合、メディアからブートし、ブートローダーを手動で再インストールする必要がある可能性があります。ブートローダーを再インストールするには、Oracle Solaris インストールメディアからブートして (たとえば、テキストインストーラ ISO イメージを使用して)、コマンドプロンプトに到達する必要があります。

▼ x86: ブートローダーをインストールする方法

GRUB 2 を再インストールする前に、ルートプールをインポートする必要があります。次に手順を示します。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. Oracle Solaris のメディアからシステムをブートします。

3. ルートプールをインポートします。

```
# zpool import -f pool-name
```

4. ブートローダーをインストールします。

```
# bootadm install-bootloader [-f] -P pool-name
```

-f システム上のブートローダーのバージョンをダウングレードしないよう、ブートローダーのインストールの際に、バージョンチェックを強制的にバイパスします。

注記 - 確実にブートローダーをメディア上にあるバージョンで上書きするのでない限り、-f オプションを使用しないでください。

-P 使用するプールのブート構成を指定します。

5. ルートプールをエクスポートします。

```
# zpool export pool-name
```

6. システムをリブートします。

▼ x86: デフォルトではない場所に GRUB をインストールする方法

BIOS ファームウェアを搭載するシステムでは、GRUB 2 をマスターブートレコードにインストールする必要があるか、望ましいことがあります。次の手順にその実行方法を説明します。インストール後、アクティブなパーティションとしてマークされている DOS パーティションに関係なく GRUB 2 はデフォルトのシステムブートローダーになります。BIOS ファームウェアを搭載するシステムで DOS パーティションが使用されており、Solaris パーティションがプライマリパーティションである場合、デフォルトの GRUB 2 インストールの場所がパーティションブートレコードです。パーティションが論理パーティションの場合、GRUB 2 は常に MBR にインストールされます。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. ブートローダーを MBR の場所にインストールします。

```
# bootadm install-bootloader -M
```

3. システムをリブートします。

x86: GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする

最後の GRUB 2 ブート環境を破棄しても、システムによって GRUB Legacy ブートローダーが自動的に再インストールされないため、GRUB Legacy ブートローダーを再インストールする場合、まず GRUB Legacy ブートローダーファイル (/boot/grub/stage1 および /boot/grub/stage2 内) を含む最新のブート環境にブートする必要があります。

installgrub コマンドはこのリリースで非推奨になったため、GRUB Legacy ブートローダーをサポートするリリースを実行している場合にのみ使用してください。installgrub(1M) を参照してください。

▼ x86: GRUB 2 がインストールされているシステムに GRUB Legacy をインストールする方法

次の手順は、システムを GRUB Legacy をサポートするリリースから、Oracle Solaris 11.3 にアップグレードした場合に適用されます。

システムを古い GRUB Legacy ブートローダーに戻す場合は、次の手順を使用します。



注意 - 次のステップは、Oracle Solaris リリースまたは Oracle Solaris 11.3 への更新に使用した Support Repository Update (SRU) を含むブート環境から実行してください。さらに、zpool upgrade コマンドの以前のバージョン 33 を使用して、ZFS プールの機能をアップグレードしている場合は、GRUB Legacy にダウングレードできないか、この手順の手順 2 を実行できません。ルートプールを以前のバージョン 33 にアップグレードしたあとに GRUB Legacy に強制的にダウングレードすると、システムがブート不能になります。

1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. **Oracle Solaris 11.3 リリースにアップグレードされたブート環境からブートします。**
3. **beadm destroy コマンドを使用して、システムからすべての GRUB 2 ブート環境を削除します。** 『Oracle Solaris 11.3 ブート環境の作成と管理』の「ブート環境の破棄」を参照してください。
Oracle Solaris 11.3 リリースを含むいずれかのブート環境をアクティブ化すると Legacy GRUB ブートローダーが GRUB 2 に置き換えられてしまうため、このステップを実行することで、GRUB 2 を誤ってアクティブ化したり、インストールしたりすることはなくなります。
4. **最新の GRUB Legacy バージョンを含むブート環境で、次のように、システムに GRUB Legacy を強制的に再インストールします。**

```
# bootadm install-bootloader -f
```

注記 - これらの手順後にリブートする必要はありません。次の完全リブート時に、GRUB Legacy ブートローダーが実行します。

◆◆◆ 第 3 章

システムのシャットダウン

この章では、Oracle Solaris 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。この章で SPARC または x86 ベースのシステムのみに適用される情報については、その旨が示されています。

この章の内容を示します。

- [59 ページの「システムのシャットダウン」](#)
- [60 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」](#)
- [61 ページの「システムのシャットダウン」](#)
- [66 ページの「システムデバイスの電源の切断」](#)

システムのブートに関する概要情報は、[第1章「システムのブートとシャットダウンの概要」](#)を参照してください。

システムのシャットダウン

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` の役割になる必要があります。
- `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、[79 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` コマンドは、システムをシャットダウンするため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。SMF によって管理されるシステムサービスは、逆の依存関係の順序でシャットダウンされます。

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があります。

次の表に、各種シャットダウンコマンドとその使用時の推奨事項を説明します。

表 4 シャットダウンコマンド

コマンド	説明	使用する状況
shutdown	init プログラムを呼び出してシステムをシャットダウンする実行可能ファイル。デフォルトでは、システムは実行レベル S に移行します。	このコマンドは、実行レベル 3 で動作しているシステムをシャットダウンするために使用します。
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。

- [61 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」](#)
- [62 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」](#)
- [66 ページの「init コマンドを使用してスタンドアロンシステムをシャットダウンする方法」](#)

halt コマンドの使用など、復旧のためのシステムのシャットダウンについては、[128 ページの「復旧を目的としてシステムを停止する方法」](#)を参照してください。

▼ 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 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 複数のユーザーが居るシステムをシャットダウンする場合は、システムにログインしているユーザーが居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 -iinit-state -ggrace-period -y
```

`-iinit-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せん。

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

例 14 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 2015

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 15 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 2015

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 murkey 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 murkey 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 murkey 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 murkey 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、[66 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 16 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 2015

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

Broadcast Message from root (console) on venus Fri Apr 15 06:47:50...
The system venus will be shut down in 1 minute

Broadcast Message from root (console) on venus Fri Apr 15 06:48:20...
The system venus will be shut down in 30 seconds

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 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is solaris.example.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、[第4章「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スタンドアロン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** 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

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 17 **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、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。

注記 - 電源ボタンを押して、x86 ベースのシステムをシャットダウンできます。このようにして、システムを停止すると、ユーザーがシャットダウンをリクエストしたことをシステムに警告する ACPI イベントがシステムに送信されます。このようにして電源をオフにすることは、`shutdown - i0` または `init 0` コマンドを実行することと同じです。

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。

システムのブート

この章では、Oracle Solaris システムのブートおよびリブートのタスク関連の情報を提供します。この章で SPARC または x86 ベースのシステムのみ適用される情報については、その旨が示されています。

この章の内容は次のとおりです。

- 69 ページの「ブート属性の表示と設定」
- 79 ページの「システムのブート」
- 91 ページの「代替のオペレーティングシステムまたはブート環境からのブート」
- 94 ページの「システムのリブート」

システムのブートに関する概要情報は、第1章「システムのブートとシャットダウンの概要」を参照してください。

ブート属性の表示と設定

次に、SPARC および x86 プラットフォームでブート属性を表示し、設定できるさまざまな方法について説明します。x86 ベースのシステムで、ブート時または `bootadm` コマンドを使用して、ブート属性を設定する特定の情報については、43 ページの「ブート時に GRUB メニューを編集してカーネル引数を追加する」を参照してください。

このセクションでは次の手順について説明します。

- 70 ページの「OpenBoot PROM を使用したブート属性の表示と設定」
- 74 ページの「EEPROM パラメータの使用」
- 78 ページの「SMF によるシャットダウンアニメーションの管理」

SPARC: OpenBoot PROM を使用したブート属性の表示と設定

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

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

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

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

▼ SPARC: システムの PROM リビジョン番号を確認する方法

1. システムを **ok PROM** プロンプトにします。

```
# init 0
```

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

```
ok banner
```

▼ SPARC: システム上のデバイスを確認する方法

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

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

- PROM の **auto-boot?** 値を **false** に変更します。

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

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

```
ok reset-all
```

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

```
ok sifting probe
```

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

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

1. システム上のデバイスを確認します。

```
ok probe-device
```

2. (オプション) 電源障害後や `reset` コマンドの使用後にシステムがリブートされるようにするには、`auto-boot?` 値を `true` にリセットします。

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

3. システムをマルチユーザー状態にブートします。

```
ok reset-all
```

例 18 SPARC: システム上のデバイスを確認する

次の例は、システムに接続されているデバイスの確認方法を示しています。

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

Sun Fire T200, No Keyboard
.
.
.
Ethernet address 0:14:4f:1d:e8:da, Host ID: 841de8da.
ok probe-ide
Device 0 ( Primary Master )
Removable ATAPI Model: MATSHITACD-RW CW-8124

Device 1 ( Primary Slave )
Not Present

Device 2 ( Secondary Master )
Not Present

Device 3 ( Secondary Slave )
Not Present

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

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

```

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

▼ SPARC: デフォルトのブートデバイスを調べる方法

1. システムを **ok PROM** プロンプトにします。

```
# init 0
```

2. デフォルトのブートデバイスを確認します。

```
ok printenv boot-device
```

`boot-device` ブートするデバイスを設定する値を示します。
 詳細は、[printenv\(1B\)](#)のマニュアルページを参照してください。

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

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

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

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

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

始める前に デフォルトのブートデバイスをほかのデバイスに変更するには、まずシステム上のデバイスを確認しておく必要があります。システム上のデバイスを確認する方法については、[70 ページ](#)の「システム上のデバイスを確認する方法」を参照してください。

1. システムを **ok PROM** プロンプトにします。

```
# init 0
```

2. boot-device の値を変更します。

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

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

3. デフォルトのブートデバイスが変更されていることを確認します。

```
ok printenv boot-device
```

4. 新しい boot-device 値を保存します。

```
ok reset-all
```

新しい boot-device 値が PROM に書き込まれます。

例 19 SPARC: ブート PROM を使用してデフォルトのブートデバイスを変更する

この例では、デフォルトのブートデバイスをディスクに設定しています。

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

screen not found.
Can't open input device.
Keyboard not present. Using ttya for input and output.
.
.
.
Rebooting with command: boot disk1
Boot device: /pci@1f,4000/scsi@3/disk@1,0 File and args:
```

この例では、デフォルトのブートデバイスをネットワークに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
```

```
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device =          net
ok printenv boot-device
boot-device            net                disk
ok reset
.
.
.
Boot device: net  File and args:

pluto console login:
```

EEPROM パラメータの使用

`eeprom` コマンドを使用して、EEPROM のパラメータの値を表示し、変更できます。EEPROM パラメータを表示するために、特別な権限は必要ありません。ただし、これらのパラメータを変更するには、`root` 役割になる必要があります。

x86 プラットフォームで EEPROM プロパティを設定し、保存する方法についての次の追加情報を参照してください。

- X86 プラットフォームでは、EEPROM プロパティの設定を次のようにしてシミュレートします。
 - Oracle Solaris 固有のプロパティを `/boot/solaris/bootenv.rc` ファイルに格納します。
 - GRUB メニューを操作して、特定の EEPROM プロパティを設定した場合の効果をシミュレートします。
 - UEFI 環境に固有の変数の NVRAM ストレージを実装します。
- `boot-args` または `boot-file` プロパティを設定して、特殊な GRUB メニューエントリを作成し、操作します。これは x86 プラットフォームで影響をシミュレートする唯一の方法です。特殊な GRUB メニューエントリのタイトルは、`Solaris bootenv rc` です。この特殊なエントリは、作成時に、デフォルトのエントリとしてマークされます。
- `eeprom` コマンドを使用して設定されたプロパティは、たとえば、ブート時に GRUB メニューを編集するなど、カーネルコマンド行でそれらのプロパティ名を別の値に設定することによってオーバーライドできます。たとえば、`eeprom` コマンドを使用し、次にブート時にカーネルコマンド行に `B console=text` を追加して、コンソールプロパティを `graphics` に設定します。この場合、`bootenv.rc` ファイルで `graphics` の値を指定していても、コンソールタイプは `text` に設定されます。

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

UEFI システムの EEPROM パラメータ

UEFI 対応システムの場合、パラメータは 2 つの場所に格納されます。Oracle Solaris 固有の変数は `bootenv.rc` ファイルに格納されます。UEFI 固有の変数は NVRAM ストアに設定されます。OBP を搭載した SPARC と異なり、Oracle Solaris 変数は UEFI ファームウェアで使用されません。UEFI 固有の変数を使用可能にするには、`-u` オプションを指定して `eeprom` コマンドを使用します。

ほとんどの UEFI 変数はバイナリ形式であり、人間が判読可能な形式に変換されません。変換が可能でない場合、`hexdump` が出力されます。

EEPROM パラメータの表示

EEPROM パラメータはプラットフォームによって異なります。たとえば、`boot-device` は SPARC プラットフォームのパラメータですが、x86 プラットフォームのパラメータではありません。使用しているシステムタイプで使用可能な EEPROM パラメータを表示するには、`eeprom` コマンドを引数なしで使用します。

例 20 すべての EEPROM パラメータの表示

次の例は、x86 ベースのシステムでの `eeprom` コマンドの出力です。

```
$ eeprom
keyboard-layout=Unknown
ata-dma-enabled=1
ataapi-cd-dma-enabled=1
ttyb-rts-dtr-off=false
ttyb-ignore-cd=true
ttya-rts-dtr-off=false
ttya-ignore-cd=true
ttyb-mode=9600,8,n,1,-
ttya-mode=9600,8,n,1,-
lba-access-ok=1
console=ttya
```

例 21 特定の EEPROM パラメータの表示

特定の EEPROM パラメータの値を表示するには、次のように `eeprom` コマンドにパラメータの名前を追加します。

```
$ /usr/sbin/eeprom console
console=ttya
```

例 22 全ての UEFI EEPROM パラメータの表示

次の例では、UEFI モードのシステムで、全ての UEFI パラメータを表示する方法を示します。このコマンドを使用するには、root 役割になる必要があります。

```
# eeprom -u
MonotonicCounter=0x1f2
OsaBootOptNum=0xffff
ConOut=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549440)
/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/VenPcAnsi()
ConIn=/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/
VenPcAnsi()
/PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x8,0x0)
BootOrder=Boot0000 Boot0001 Boot0002 Boot0003 Boot0004 Boot0005 Boot0006
Lang=eng
PlatformLang=en-US
Timeout=0x1
Boot0001=description:string=[UEFI]USB:USBIN:USB USB Hard Drive , flags:int=1, device_path:
\
string=/PciRoot(0x0)/Pci(0x1a,0x0)/USB(0x1,0x0)/USB(0x2,0x0)/
HD(1,MBR,0x004D5353,0x800,0x3b5800), \
optional_data:string=AMBO
Boot0002=description:string=[UEFI]PXE:NET0:Intel(R) Ethernet Controller 10 Gigabit X540-
AT2, \
flags:int=1, device_path:string=/PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0)/MAC(002128e77478),
\
optional_data:string=AMBO
Boot0003=description:string=[UEFI]PXE:NET1:Intel(R) Ethernet Controller 10 Gigabit X540-
AT2, \
flags:int=1, device_path:string=/PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x1)/MAC(002128e77479),
\
optional_data:string=AMBO
Boot0004=description:string=[UEFI]PXE:NET2:Intel(R) Ethernet Controller 10 Gigabit X540-
AT2, \
flags:int=1, device_path:string=/PciRoot(0x1)/Pci(0x1,0x0)/Pci(0x0,0x0)/MAC(002128e7747a),
\
optional_data:string=AMBO
Boot0005=description:string=[UEFI]PXE:NET3:Intel(R) Ethernet Controller 10 Gigabit X540-
AT2, \
flags:int=1, device_path:string=/PciRoot(0x1)/Pci(0x1,0x0)/Pci(0x0,0x1)/MAC(002128e7747b),
\
optional_data:string=AMBO
Boot0006=description:string=[UEFI]SAS:PCIE3:ATA HITACHI HDS7225SA81A, flags:int=1, \
device_path:string=/PciRoot(0x0)/Pci(0x3,0x0)/Pci(0x0,0x0) \
/MessagingPath(10,2c00b .... 12010100) \
/HD(1,GPT,BCB01265-4665-F1CA-8BF5-9C4FB95962FA,0x100,0x80000), optional_data:string=AMBO
Boot0000=description:string=Oracle Solaris s12_13, flags:int=1, device_path: \
string=HD(1,GPT,C7398875-60D2-A9E0-83EE-94DAA21B0383,0x100,0x80000),
file_path:string=/EFI/Oracle/grubx64.efi
USB_POINT=5139417f00000000
ConOutDev=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549440)
/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/VenPcAnsi()
ConInDev=/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/
VenPcAnsi()
/PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x8,0x0)
BootOptionSupport=0x1
ErrOutDev=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549440)
/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/VenPcAnsi()
ErrOut=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549440)
/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None)/VenPcAnsi()
PlatformLangCodes=en-US
S3PerfAdd=hexdump:989fd6aa00000000
```

```
LangCodes=eng
BootCurrent=Boot0000
```

例 23 特定の UEFI パラメータの表示

```
# eeprom -u Boot0000
Boot0000=description:string=Oracle Solaris s12_13, flags:int=1, device_path: \
string=HD(1,GPT,C7398875-60D2-A9E0-83EE-94DAA21B0383,0x100,0x80000), \
file_path:string=/EFI/Oracle/grubx64.efi
```

▼ SPARC: ブート属性を設定する方法

次の手順では、SPARC ベースのシステムでデフォルトのブートデバイスを設定する方法について説明します。X86 プラットフォームでは、UEFI Boot Manager などのファームウェアのタイプに対応するセットアップユーティリティによってブートデバイスを設定します。

注記 - X86 プラットフォームでは、UEFI Boot Manager などのファームウェアのタイプに対応するセットアップユーティリティによってブートデバイスを設定します。

1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. **ブート属性を指定します。**

```
# eeprom attribute=value
```

3. **属性が設定されていることを確認します。**

```
# eeprom attribute
```

この出力には、変更された属性の新しい eeprom 値が表示されます。

例 24 auto_boot パラメータの設定

auto_boot パラメータを false に設定するには、root 役割を使用して次のコマンドを入力します。

```
# eeprom auto-boot?=false
```

例 25 カーネルブート引数の設定

boot-args パラメータに値を指定して、カーネルブート引数を設定できます。たとえば、次のコマンドを入力して、システムがカーネルデバッガをブートすることを指定します。

```
# eeprom boot-args=-k
```

例 26 コンソールデバイスのパラメータの設定

Oracle Solaris コンソール設定をグラフィックモードに切り替えるには、次のコマンドを使用します。

```
# eeprom console=graphics
```

例 27 UEFI 対応システムのパラメータの設定

次の例では、UEFI 対応システムでブート順序を変更する方法を示します。

```
# eeprom -u BootOrder="Boot0005 Boot0001 Boot0002 Boot0003 Boot0004 Boot0000"
```

▼ UEFI EEPROM パラメータを削除する方法

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. UEFI EEPROM パラメータを削除します。

この例では、attribute という名前のカスタムパラメータが削除されます。

```
# eeprom -u -d attribute
```

3. 属性が削除されていることを確認します。

```
# eeprom -u attribute
eeprom: read: attribute doesn't exist
```

x86: SMF によるシャットダウンアニメーションの管理

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
```

システムのブート

次の手順では、さまざまな状態にシステムをブートする方法 (実行レベルブートとも呼ばれる) を説明しています。

このセクションでは次の手順について説明します。

- 79 ページの「実行レベルの動作」
- 81 ページの「システムをマルチユーザー状態 (実行レベル 3) にブートする方法」
- 83 ページの「システムをシングルユーザー状態 (実行レベル S) にブートする方法」
- 87 ページの「システムを対話式でブートする方法」

実行レベルの動作

システムの実行レベル (*init* 状態としても知られる) は、ユーザーが利用できるサービスおよび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 5 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	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

実行レベル	init 状態	タイプ	目的
6	リブート状態	リブート	<p>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。</p> <p>オペレーティングシステムを停止し、<code>/etc/inittab</code> ファイルの <code>initdefault</code> エントリに定義されている状態にリブートします。</p> <p>SMF サービス <code>svc:/system/boot-config:default</code> はデフォルトで有効になっています。 <code>config/fastreboot_default</code> プロパティーが <code>true</code> に設定されている場合に <code>init 6</code> を指定すると、システムの特定の機能に応じて、特定のファームウェア初期化手順およびテスト手順が省略されます。96 ページの「リブートプロセスの高速化」を参照してください。</p>

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 6 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
5	<code>milestone/single-user:default</code>
2	<code>milestone/multi-user:default</code>
3	<code>milestone/multi-user-server:default</code>

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 11.3 でのシステムサービスの管理』の「システムブート時のサービスの起動に関する問題を調査する方法」を参照してください。

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 28 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	以前の実行レベルを識別します

▼ 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 になっているシステムを実行レベル 3 にブートします。この手順で、SPARC プラットフォームまたは x86 プラットフォームのいずれかに適用する情報については、それに応じて言及されています。

1. **root** 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. プラットフォームに応じて、次のいずれかを実行します。

■ **SPARC** プラットフォームの場合:

a. システムを **ok PROM** プロンプトにします。

```
# init 0
```

b. システムを実行レベル 3 にブートします。

```
ok boot
```

■ **X86** プラットフォームの場合、システムを実行レベル 3 にリブートします。

```
# reboot
```

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

3. システムが**実行レベル 3** になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

例 29 SPARC: マルチユーザー状態 (実行レベル 3) へのシステムのブート

次の例に、ブートプロセスの起動後に、SPARC ベースのシステムを実行レベル 3 にブートして生成されるメッセージを示します。

```
ok boot
Probing system devices
Probing memory
ChassisSerialNumber FN62030249
Probing I/O buses
.
.
.
.
OpenBoot 4.30.4.a, 8192 MB memory installed, Serial #51944031.
Ethernet address 0:3:ba:18:9a:5f, Host ID: 83189a5f.
Rebooting with command: boot
Boot device: /pci@1c,600000/scsi@2/disk@0,0:a File and args:
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
misc/forthdebug (455673 bytes) loaded
Hardware watchdog enabled
```

```

Hostname: portia-123
NIS domain name is solaris.example.com

portia-123 console login: NIS domain name is solaris.example.com

```

例 30 x86: マルチユーザー状態 (実行レベル 3) へのシステムのブート

次の例に、ブートプロセスの起動後に、x86 ベースのシステムを実行レベル 3 にブートした場合のメッセージを示します。高速リブート機能はこのリリースでデフォルトであるため (x86 プラットフォームの場合)、reboot コマンドでシステムをブートすると、システムの高速リブートが開始されます。つまり、BIOS または UEFI ファームウェアがバイパスされます。さらに、システムのブート時に、GRUB メニューが表示されません。システムのファームウェアにアクセスするか、ブート時に GRUB メニューを編集する必要がある場合は、reboot コマンドと -p オプションを使用します。101 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。

```

~# reboot
Apr 23 13:30:29 system-04 reboot: initiated by ... on /dev/console
Terminated
system-04% updating /platform/i86pc/boot_archive
updating /platform/i86pc/amd64/boot_archive

system-04 console login: syncing file systems... done
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Hostname: system-04

system-04 console login: <username>
Password: xxxxxx
Last login: Mon Apr 23 11:06:05 on console
Oracle Corporation      SunOS 5.11      11.3      May 2015
# who -r
  run-level 3  Apr 23 13:31      3      0  S

```

▼ 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

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を実行する目的で、システムをシングルユーザー状態にブートします。

1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「[割り当てられている管理権利の使用](#)」を参照してください。
2. プラットフォームに応じて、次のいずれかを実行します。
 - SPARC プラットフォームの場合:

- a. システムを **ok PROM** プロンプトにします。

```
# init 0
```

- b. システムをシングルユーザー状態にブートします。

```
ok boot -s
```

- c. 次のメッセージが表示されたら、**root** のパスワードを入力します。

```
SINGLE USER MODE
```

```
Root password for system maintenance (control-d to bypass): xxxxxx
```

■ **x86** プラットフォームの場合:

- a. システムの標準リブートを実行します。

```
# reboot -p
```

高速リブート機能はデフォルトで有効にされているため、システムのリブート時に **-p** オプションを指定して、ブート時に GRUB メニューを表示できるようにする必要があります。**-p** オプションを指定する必要がないように、高速リブート機能を無効にするには、[100 ページの「デフォルトの高速リブートの動作を変更する」](#)を参照してください。

- システムで「**Press Any Key to Reboot**」プロンプトが表示された場合は、任意のキーを押して、システムをリブートします。または、このプロンプトで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。

- b. **GRUB** メニューが表示されたら、変更するブートエントリを選択し、**e** を入力してそのエントリを編集します。

- c. 矢印キーを使用して、**\$multiboot** 行に移動し、行の末尾に **-s** と入力します。

- d. **GRUB** 編集メニューを終了し、編集したばかりのエントリをブートするには、**Control-X** を押します。**UEFI** ファームウェアを搭載するシステムでシリアルコンソールを使用していない場合、**F10** キーを押してもエントリがブートします。

ブート時に、GRUB メニューを編集する詳細については、[43 ページの「ブート時に GRUB メニューを編集してカーネル引数を追加する」](#)を参照してください。

3. 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
```

4. 実行レベル S への変更が必要な保守タスクを実行します。

5. システムをリブートします。

例 31 SPARC: シングルユーザー状態 (実行レベル S) へのシステムのブート

次の例に、ブートプロセスの起動後に、SPARC ベースのシステムを実行レベル S にブートして生成されるメッセージを示します。

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 122 system services are now being stopped.
Mar  5 10:30:33 system1 syslogd: going down on signal 15
svc.startd: Killing user processes.
umount: /ws busy
umount: /home busy
Mar  5 17:30:50 The system is down. Shutdown took 70 seconds.
syncing file systems... done
Program terminated
{1c} ok boot -s

SC Alert: Host System has Reset
NOV 17 21:46:59 ERROR: System memory downgraded to 2-channel mode from 4-channel mode
NOV 17 21:47:00 ERROR: Available system memory is less than physically installed memory
NOV 17 21:47:00 ERROR: System DRAM Available: 008192 MB Physical: 016384 MB
Sun Fire T200, No Keyboard
.
.
Ethernet address 0:14:4f:1d:e8:da, Host ID: 841de8da.

ERROR: The following devices are disabled:
      MB/CMP0/CH2/R0/D0

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a
File and args: -s

SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015 Oracle and/or its affiliates. All rights reserved.
NOTICE: Hypervisor does not support CPU power management
Booting to milestone "milestone/single-user:default".
Hostname: system1
Requesting System Maintenance Mode
SINGLE USER MODE
Enter root password (control-d to bypass): xxxxxx
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode

Mar  5 10:36:14 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.3      ay 2015
root@system1:~# who -r
run-level S Mar  5 10:35      S      0  0
root@tsystem1:~#
```

例 32 x86: シングルユーザー状態 (実行レベル S) へのシステムのブート

次の例に、ブートプロセスの起動後に、x86 ベースのシステムを実行レベル S にブートした場合のメッセージを示します。

```

root@system-04:~# init 0
root@system-04:~# svc.startd: The system is coming down. Please wait.
svc.startd: 129 system services are now being stopped.
Apr 23 13:51:28 system-04 syslogd: going down on signal 15
svc.startd: Killing user processes.
umount: /home busy
Apr 23 13:51:36 The system is down. Shutdown took 26 seconds.
syncing file systems... done
Press any key to reboot.
.
.
.LSI Corporation MPT SAS BIOS
MPTBIOS-6.26.00.00 (2008.10.14)
Copyright 2000-2008 LSI Corporation.

Initializing..|Press F2 to runS POPUP (CTRL+P on Remote Keyboard)
Press F12 to boot from the network (CTRL+N on Remote Keyboard)
System Memory : 8.0 GB , Inc.
Auto-Detecting Pri Master..ATAPI CDROM                                0078
Ultra DMA Mode-2
.
.
.
GNU GRUB version 1.99,5.11.0.175.1.0.0.14.0

*****
*Oracle Solaris 11.3*
*
*
*
*
*
*
*
*
*
*
*
*****

Use the * and * keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

GNU GRUB version 1.99,5.11.0.175.1.0.0.14.0

*****
* setparams 'Oracle Solaris 11.3'*
*
* insmod part_msdos*
* insmod part_sunpc*
* insmod part_gpt*
* insmod zfs*
* search --no-floppy --fs-uuid --set=root cd03199c4187a7d7*
* zfs-bootfs /ROOT/s11.3/@/ zfs_bootfs*
* set kern=/platform/i86pc/kernel/amd64/unix*
* echo -n "Loading ${root}/ROOT/s11.3 /@$kern: "*
* $multiboot /ROOT/s11.3@$kern $kern -B $zfs_bootfs -s*
* set gfxpayload="1024x768x32;1024x768x16;800x600x16;640x480x16;640x480x1\

```

```

* 5;640x480x32"
*****
Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for
a command-line or ESC to discard edits and return to the GRUB menu.

Booting a command list

Loading hd0,msdos1,sunpc1/ROOT/s11.3/@/platform/i86pc/kernel/amd64/unix: 0
%...done.
Loading hd0,msdos1,sunpc1/ROOT/s113/@/platform/i86pc/amd64/boot_archive:
0%...
.
.
.
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 201, Oracle and/or its affiliates. All rights reserved.
NOTICE: kmem_io_2G arena created
Booting to milestone "milestone/single-user:default".
Hostname: system-04
Requesting System Maintenance Mode
SINGLE USER MODE

Enter user name for system maintenance (control-d to bypass): root
Enter root password (control-d to bypass): xxxxxxx
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode

May  8 11:13:44 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.3      May 2015
You have new mail.

root@system-04:~# who -r
.          run-level S  Apr 23 14:49      S      0  0

```

▼ システムを対話式でブートする方法

対話式でのシステムのブートは、元のファイルが破損しているか、システムがブートしないために、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。

次の手順では、ブート環境が1つしかないシステムの対話式のブート時に、代替の `/etc/system` ファイルを指定する方法について説明します。または、代替ブート環境をブートできます。

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` のファイル名を `/boot/solaris/filelist.ramdisk` ファイルに追加します。

```

# 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. プラットフォームに応じて、次のいずれかを実行します。

■ SPARC プラットフォームの場合:

- a. システムを **ok PROM** プロンプトにします。

```
# init 0
```

- b. システムを対話式でブートします。

```
ok boot -a
```

■ x86 プラットフォームの場合:

- a. システムの標準リブートを実行します。

```
# reboot -p
```

- b. **GRUB** メニューが表示されたら、対話式でブートするブートエントリを選択し、**e** を入力してそのエントリを編集します。

- c. `$multiboot` 行の末尾に **-a** と入力します。

- d. **GRUB** 編集メニューを終了し、編集したばかりのエントリをブートするには、**Control-X** を押します。UEFI ファームウェアを搭載するシステムがあり、シリアルコンソールを使用していない場合、**F10** キーを押すことによってもエントリがブートします。

3. 代替ファイルシステムを要求されたら、作成したバックアップファイルを指定し、**Return** キーを押します。例:

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

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。

4. リタイアストアプロンプトで、**Return** キーを押すか、`/dev/null` を指定してバイパスします。

注記 - `/etc/devices/retire_store` ファイルは、障害管理アーキテクチャー (FMA) によってリタイアされるデバイスのバッキングストアです。システムはこれらのデバイスを使用しなくなりました。必要に応じて、`/etc/devices/retire_store` の代替ファイルを提供できます。ただし、復旧のために `/dev/null` を指定することは、`/etc/devices/retire_store` ファイルの内容に関係なく、システムをブートするもっとも有益な選択肢です。

5. システムのブート後に、`/etc/system` ファイルの問題を修正します。

6. システムをリブートします。

```
# reboot
```

例 33 SPARC: システムを対話式でブートする

次の例では、システムのデフォルト (角かっこ [] で示す) が受け入れられます。

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 121 system services are now being stopped.
Apr 22 00:34:25 system-28 syslogd: going down on signal 15
svc.startd: Killing user processes.
umount: /home busy
Apr 22 06:34:37 The system is down. Shutdown took 18 seconds.
syncing file systems... done
Program terminated
{11} ok boot -a

SC Alert: Host System has Reset

Sun Fire T200, No Keyboard
Copyright (c) 1998, 2015, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #74139288.
Ethernet address 0:14:4f:6b:46:98, Host ID: 846b4698.

Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a File and args: -a
Name of system file [/etc/system]: /etc/system.bak
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Retire store [/etc/devices/retire_store] (/dev/null to bypass): Press Return

system-28 console login:
```

例 34 x86: システムを対話式でブートする

次の例では、x86 ベースのシステムが対話式でブートされます。

```
root@system-04:~# reboot -p
Apr 23 15:37:04 system-04 reboot: initiated by user1 on /dev/consoleTerminated
system-04% syncing file systems... done
rebooting...
.
.
.LSI Corporation MPT SAS BIOS
MPTBIOS-6.26.00.00 (2008.10.14)
Copyright 2000-2008 LSI Corporation.

Initializing..|Press F2 to runS POPUP (CTRL+P on Remote Keyboard)
Press F12 to boot from the network (CTRL+N on Remote Keyboard)
System Memory : 8.0 GB , Inc.
Auto-Detecting Pri Master..ATAPI CDROM 0078
Ultra DMA Mode-2
GNU GRUB version 1.99,5.11.0.175.1.0.0.14.0

*****
*Oracle Solaris 11.3 *
* *
* *
* *
*****
```

```

*
*
*
*
*
*
*
*
*
*
*****

```

Use the * and * keys to select which entry is highlighted.
 Press enter to boot the selected OS, 'e' to edit the commands
 before booting or 'c' for a command-line.

GNU GRUB version 1.99,5.11.0.175.1.0.0.15.1

```

+-----+
| setparams 'Oracle Solaris 11.3' |
| | |
| insmod part_msdos |
| insmod part_sunpc |
| insmod part_gpt |
| insmod zfs |
| search --no-floppy --fs-uuid --set=root cd03199c4187a7d7 |
| zfs-bootfs /ROOT/s11.3/@/ zfs_bootfs |
| set kern=/platform/i86pc/kernel/amd64/unix |
| echo -n "Loading ${root}/ROOT/s11./@$kern: " |
| $multiboot /ROOT/s11.3/@/$kern $kern -B $zfs_bootfs -a |
| set gfxpayload="1024x768x32;1024x768x16;800x600x16;640x480x16;640x480x1\ |
+-----+

```

Minimum Emacs-like screen editing is supported. TAB lists
 completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for
 a command-line or ESC to discard edits and return to the GRUB menu.

Booting a command list

```

Loading hd0,msdos1,sunpc1/ROOT/s11.3/@/platform/i86pc/kernel/amd64/unix: 0
%...done.
Loading hd0,msdos1,sunpc1/ROOT/s11.3/@/platform/i86pc/amd64/boot_archive:
0%...
.
.
Name of system file [/etc/system]: /etc/system.bak
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.: 0
Retire store [/etc/devices/retire_store] (/dev/null to bypass): Press Return

NOTICE: kmem_io_2G arena created
Hostname: system-04

system-04 console login:

```

代替のオペレーティングシステムまたはブート環境からのブート

このセクションでは次の手順について説明します。

- [SPARC: 代替のオペレーティングシステムまたはブート環境からブートする方法](#)
- [x86: 代替のオペレーティングシステムまたはブート環境からブートする方法](#)

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す。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。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。大域または非大域ゾーンでのユーティリティの使用など、ブート環境の管理に関する詳細は、『[Oracle Solaris 11.3 ブート環境の作成と管理](#)』を参照してください。

x86 のみ: 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、GRUB メニューの作成に使用される `grub.cfg` ファイルは、プールの最上位データセットに見つかります。これは、プールと同じ名前を持つデータセットです。プールにはそのようなデータセットが常に 1 つだけ存在します。このデータセットは、GRUB 構成ファイルおよびデータなど、プール規模のデータに適しています。システムがブートされたあと、このデータセットは、ルートファイルシステム内の `/pool-name` にマウントされます。

x86 のみ: プール内に複数のブート可能データセット (つまり、ルートファイルシステム) が存在することがあります。プール内のデフォルトのルートファイルシステムはプールの `bootfs` プロパティによって識別されます。`grub.cfg` ファイルに存在する GRUB メニューエントリに、`zfs-bootfs` コマンドによって特定の `bootfs` が指定されていない場合、デフォルトの `bootfs` ルートファイルシステムが使用されます。各 GRUB メニューエントリは異なる `zfs-bootfs` コマンドを使用するように指定できます。これにより、プール内の任意のブート可能 Oracle Solaris インスタンスを選択できます。詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

▼ SPARC: 代替のオペレーティングシステムまたはブート環境からブートする方法

1. `root` 役割になります。

『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. システムを **ok PROM** プロンプトにします。

```
# init 0
```

3. (オプション) **boot** コマンドに **-L** オプションを使用して、使用可能なブート環境の一覧を表示します。

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

```
Select environment to boot: [1 - 2]:
```

5. システムをブートするには、画面に表示された指示に従います。

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

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

例:

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

6. システムをブートしたあと、アクティブなブート環境を確認します。

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

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

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

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

```
# df -lk
```

例 35 SPARC: 代替のブート環境からのブート

次の例は、**boot -Z** コマンドを使用して SPARC ベースのシステムで代替のブート環境からブートする方法を示しています。

```
# init 0
root@system-28:~# svc.startd: The system is coming down. Please wait.
svc.startd: 126 system services are now being stopped.
Jul  3 22:11:33 system-28 syslogd: going down on signal 15
svc.startd: Killing user processes.
umount: /home busy
Jul  3 22:11:50 The system is down. Shutdown took 23 seconds.
syncing file systems... done
Program terminated
{1c} ok boot -L
```

```
SC Alert: Host System has Reset
```

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

```
Copyright (c) 1998, 2015, Oracle and/or its affiliates. All rights reserved.  
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #74139288.  
Ethernet address 0:14:4f:6b:46:98, Host ID: 846b4698.
```

```
Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a File and args: -L  
1 Oracle Solaris 11.3 SPARC  
2 s11.3_backup  
3 s11.3_backup2  
Select environment to boot: [ 1 - 3 ]: 3
```

```
To boot the selected entry, invoke:  
boot [<root-device>] -Z rpool/ROOT/s11.3_backup2
```

```
Program terminated  
{0} ok boot -Z rpool/ROOT/s11.3_backup2
```

SC Alert: Host System has Reset

```
Sun Fire T200, No Keyboard  
Copyright (c) 1998, 2015, Oracle and/or its affiliates. All rights reserved.  
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #74139288.  
Ethernet address 0:14:4f:6b:46:98, Host ID: 846b4698.
```

```
Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a \  
File and args: -Z rpool/ROOT/s11.3_backup2  
SunOS Release 5.11 Version 11.3 64-bit  
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.  
WARNING: consconfig: cannot find driver for  
screen device /pci@780/pci@0/pci@8/pci@0/TSI,mko@0  
Loading smf(5) service descriptions: Loading smf(5)  
service descriptions: Hostname: system-28  
.
```

system-28 console login: Jul 3 22:39:05 system-28

▼ x86: 代替のオペレーティングシステムまたはブート環境からブートする方法

1. root 役割になります。
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. システムの標準リブートを実行します。

```
# reboot -p
```
3. GRUB メニューが表示されたら、ブートする代替のブート環境またはオペレーティングシステムに移動します。
4. 代替オペレーティングシステムからブートするには、Control-X を押します。

UEFI ファームウェアを搭載するシステムがあり、シリアルコンソールを使用していない場合、F10 キーを押しても代替のオペレーティングシステムがブートします。

例 36 reboot コマンドを使用して代替のブート環境からブートする

次の例に示すように、ブートエントリ番号を指定して `reboot` コマンドを使用し、代替のブートエントリをブートできます。

```
# bootadm list-menu
the location of the boot loader configuration files is: /rpool/boot/grub
default 1
timeout 30
0 s11.s.backup
1 Oracle Solaris 11.s B14
# reboot 1
Apr 23 16:27:34 system-04 reboot: initiated by userx on /dev/consoleTerminated
system-04% syncing file systems... done
SunOS Release 5.11 Version 11.s 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.

Hostname: system-04

system-04 console login:
```

システムのリブート

このセクションでは次の手順について説明します。

- [95 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」](#)
- [96 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」](#)
- [96 ページの「リブートプロセスの高速化」](#)

通常、システムは電源投入時またはシステムクラッシュのあとにリブートします。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。`init 6` コマンドは停止方法 (SMF または `rc.d`) を必要とします。一方、`reboot` コマンドでは必要としないため、`reboot` コマンドはシステムをリブートするためのより信頼できる方法になります。詳細は、[init\(1M\)](#) および [reboot\(1M\)](#) を参照してください。

`reboot` は次の操作を実行します。

- カーネルを再起動する
- ディスクで `sync` 操作を実行する
- マルチユーザーブートを開始する

`root` ユーザーはいつでも `reboot` コマンドを使用できますが、複数のユーザーが居るシステム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サービスが失われることを警告するために、最初に `shutdown` コマンドを使用します。詳細は、[第3章「システムのシャットダウン」](#) 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

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。79 ページの「[実行レベルの動作](#)」を参照してください。

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。init 6 コマンドは、オペレーティングシステムを停止し、`/etc/inittab` ファイルの `initdefault` エントリに定義されている状態にリブートします。

注記 - Oracle Solaris 11 リリースから、SMF サービス `svc:/system/boot-config:default` はデフォルトで有効になっています。 `config/fastreboot_default` プロパティが `true` に設定されている場合 (すべての x86 ベースのシステムの場合) に `init 6` コマンドは、システムの特定の機能に応じて、特定のファームウェア初期化段階およびテスト段階を省略します。SPARC ベースのシステムでは、このプロパティはデフォルトで `false` に設定されますが、プロパティを手動で有効にできます。96 ページの「[リブートプロセスの高速化](#)」を参照してください。

1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「[割り当てられている管理権利の使用](#)」を参照してください。
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 37 `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) にリブートする

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) にリブートします。

```
~# init s
```

```
~# svc.startd: The system is coming down for administration. Please wait.
Jul 20 16:59:37 system-04 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): root
Enter root password (control-d to bypass): xxxxxx
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode

Jul 20 17:11:24 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.3      May 2015
You have new mail.
~# who -r
.          run-level S  Jul 20 17:11      S      1  3
```

▼ reboot コマンドを使用してシステムをリブートする方法

この手順を使用して、実行中のシステムをマルチユーザー状態 (実行レベル 3) にリブートします。

注記 - x86 プラットフォームで、reboot コマンドを使用して、システムの高速リブートを開始し、BIOS または UEFI ファームウェアと特定のブートプロセスをバイパスします。高速リブート機能が有効になっている x86 ベースシステムの標準リブートを実行するには、reboot コマンドで -p オプションを使用します。101 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 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. システムをリブートします。

```
# reboot
```

リブートプロセスの高速化

Oracle Solaris の高速リブート機能は、SPARC プラットフォームと x86 プラットフォームの両方でサポートされています。高速リブート機能では、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。

高速リブート機能のサポートは、新しい `boot-config` サービス `svc:/system/boot-config:default` によって強化されています。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。デフォルトでこのプロパティは、x86 ベースのシステムでは `true` に設定され、SPARC ベースのシステムでは `false` に設定されています。

x86 ベースのシステムで、システムの高速リブートは、システムファームウェア (BIOS または UEFI) とブートローダープロセスをバイパスします。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 x86 プラットフォームでは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。

SPARC ベースシステムの高速リブート機能は、x86 ベースシステムとは異なる方法で動作します。SPARC プラットフォームでの高速リブートのサポートに関する次の追加情報に注目してください。

- 高速リブートは sun4u システムではサポートされていません。
- 高速リブートは sun4v システムでサポートされています。ただし、SPARC ベースのシステムの高速リブートは、x86 ベースのシステムの高速リブートとは異なります。SPARC sun4v システムでは、高速リブートは、最小限のハイパーバイザによって開始される再起動で、x86 ベースのシステムの高速リブートと同じ基本パフォーマンスを発揮します。
- SPARC ベースのシステムの高速リブートの動作はデフォルトでは無効にされています。SPARC ベースのシステムの高速リブートを開始するには、`reboot` コマンドで `-f` オプションを使用する必要があります。または、高速リブートをデフォルトの動作にするには、`config/fastreboot_default` プロパティを `true` に設定します。手順については、[100 ページの「デフォルトの高速リブートの動作を変更する」](#)を参照してください。
- SPARC ベースシステムでは、`boot-config` サービスに、`action_authorization` および `value_authorization` として `solaris.system.shutdown` 権限も必要です。

x86: quiesce 関数について

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트리ポイントである `quiesce` のデバイスドライバ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つ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。

注記 - すべてのデバイスドライバで `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手順については、[140 ページの「高速リブートが機能しない可能性のある状況」](#) および [126 ページの「高速リブートをサポートしないシステムのブートアーカイブの自動更新障害を解決する方法」](#) を参照してください。

▼ システムの高速リブートを開始する方法

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. システムのプラットフォームに応じて、次のいずれかを実行します。

■ SPARCベースのシステムでは、次のコマンドを入力します。

```
# reboot -f
```

■ x86 ベースのシステムでは、次のいずれかのコマンドを入力します。

```
# reboot
```

```
# init 6
```

これらのいずれかのコマンドを実行すると、システムが `grub.cfg` ファイルのデフォルトのエントリにリブートします。

x86: 新しくアクティブにしたブート環境へのシステムの高速リブートを開始する

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 38 x86: 新しくアクティブにしたブート環境へのシステムの高速リブートを開始する

次の例では、`2013-06-10-be` という名前のブート環境をアクティブにして、高速リブートされるようにしています。

```
# beadm activate 2013-06-10-be
# reboot
```

例 39 x86: 代替ブート環境を指定する間のシステムの高速リブート開始

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'
```

たとえば、次のように、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します。

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

例 40 x86: 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

次のように、zfsbe3 ブート環境へのシステムの高速リブートを開始します。

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

例 41 x86: 新しいカーネルへのシステムの高速リブートの開始

次のように、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します。

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

例 42 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高速リブートの開始

次のように、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します。

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

例 43 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高速リブートの開始

次のように、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高速リブートを開始します。

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

デフォルトの高速リブートの動作を変更する

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` を通じて実装されます。`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にシステムの自動高速リブートが有効になります。`config/fastreboot_default` プロパティを `true` に設定すると、システムは高速リブートを自動的に実行するため、`reboot -f` コマンドを使用する必要はありません。デフォルトでこのプロパティの値は、x86 ベースのシステムでは `true` に設定され、SPARC ベースのシステムでは `false` に設定されています。

例 44 x86: `boot-config` サービスのプロパティの構成

`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`

これらのプロパティは、`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ません。

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

例 45 SPARC: `boot-config` サービスのプロパティの構成

次の例に、`boot-config` SMF サービスプロパティを `true` に設定して、SPARC ベースのシステムで高速リブートをデフォルトの動作にする方法を示します。

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

```
# svcadm refresh svc:/system/boot-config:default
```

プロパティの値を `true` に設定すると、リブートプロセスが高速化するため、高速リブート機能をサポートするシステムで特定の POST テストをバイパスできます。プロパティが `true` に設定されている場合、`reboot` コマンドで `-f` オプションを使用しなくてもシステムの高速リブートを実行できます。

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
```


◆◆◆ 第 5 章

システムのネットワークからのブート

この章では、SPARC および x86 ベースの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、およびタスク関連の情報を提供します。この章で SPARC または x86 ベースのシステムのみにも適用される情報については、その旨が示されています。

この章の内容は次のとおりです:

- [SPARC: システムのネットワークからのブート](#)
- [108 ページの「システムのネットワークからのブート」](#)

システムのブートに関する概要情報は、[第1章「システムのブートとシャットダウンの概要」](#)を参照してください。

Oracle Solaris をインストールする目的で、システムをネットワークからブートする場合については、『[Oracle Solaris 11.3 システムのインストール](#)』を参照してください。

SPARC: システムのネットワークからのブート

このセクションでは次の手順について説明します。

- [104 ページの「ネットワークブートプロセス」](#)
- [104 ページの「システムをネットワークからブートするための要件」](#)
- [105 ページの「OpenBoot PROM へのネットワークブート引数の設定」](#)
- [107 ページの「DHCP を使用して自動ブートするための NVRAM 別名の設定」](#)
- [107 ページの「システムをネットワークからブートする方法」](#)

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
- 復旧目的のため

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

この Oracle Solaris リリースでの DHCP の動作に関する一般的な情報と、DHCP サーバーを設定するための具体的な情報については、『[Oracle Solaris 11.3 での DHCP の作業](#)』を参照してください。

SPARC: ネットワークブートプロセス

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

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

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

SPARC: システムをネットワークからブートするための要件

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

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

ブートクライアントのネットワークインタフェースの構成に必要な情報は、DHCP サーバーから提供されます。自動インストーラ (AI) サーバーを設定している場合は、そのサーバーも DHCP サーバーにすることができます。または、別の DHCP サーバーを設定できます。詳細は、『[Oracle Solaris 11.3 での DHCP の作業](#)』を参照してください。

- また、`tftp` サービスを提供するブートサーバーも必要です。

SPARC: OpenBoot PROM へのネットワークブート引数の設定

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

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

PROM が必要な情報には次のものが含まれます。

- ブートクライアントの IP アドレス

注記 - WAN ブートには IPv6 アドレスのサポートが含まれません。

- ブートファイルの名前
- ブートファイルイメージを提供するシステムの IP アドレス

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

ネットワークブートで使用する構文は次のとおりです。

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

`protocol` 使用するアドレス検索プロトコルを指定します。

`key=value` 構成パラメータを属性ペアとして指定します。

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

パラメータ	説明
<code>tftp-server</code>	TFTP サーバーの IP アドレス。
<code>file</code>	TFTP を使ってダウンロードするファイル、または WAN ブートの URL。
<code>host-ip</code>	ブートクライアントの IP アドレス (ドット区切り 10 進表記)。
<code>router-ip</code>	デフォルトのルーターの IP アドレス (ドット区切り 10 進表記)。
<code>subnet-mask</code>	サブネットマスク (ドット区切り 10 進表記)

パラメータ	説明
client-id	DHCP クライアント識別子: DHCP サーバーによって許可される任意の一意の値に設定できます。AI クライアントの場合、この値は、クライアントの 16 進数ハードウェアアドレスの前に、Ethernet ネットワークを示す文字列 01 を付けた値に設定する必要があります。たとえば、16 進 Ethernet アドレス 8:0:20:94:12:1e を持つ Oracle Solaris クライアントには、クライアント ID 0108002094121E を使用します。
hostname	DHCP トランザクションに使用するホスト名
http-proxy	HTTP プロキシサーバー指定 (<i>IPADDR[: PORT]</i>)
tftp-retries	TFTP の最大リトライ回数
dhcp-retries	DHCP の最大リトライ回数

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

始める前に システムをネットワークからブートするために必要なすべての予備的タスクを完了します。詳細については、[104 ページの「システムをネットワークからブートするための要件」](#)を参照してください。

1. ネットワークからブートするシステム上で、**root** 役割になります。
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. **network-boot-arguments** パラメータに適切な値を指定します。

```
# eeprom network-boot-arguments="protocol,hostname=hostname"
```

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

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

3. システムを **ok PROM** プロンプトにします。

```
# init 0
```

4. ネットワークからシステムをブートします。

```
ok boot net
```

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

SPARC: DHCP を使用して自動ブートするための NVRAM 別名の設定

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

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

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

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

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



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

▼ SPARC: 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。104 ページの「システムをネットワークからブートするための要件」を参照してください。
 - Oracle Solaris をインストールするためにシステムをネットワーク経由でブートする場合は、最初に AI クライアントイメージをダウンロードし、そのイメージに基づいてインストールサービスを作成します。手順については、『Oracle Solaris 11.3 システムのインストール』のパート 3、「インストールサーバーを使用したインストール」を参照してください。

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 必要に応じて、システムを ok PROM プロンプトにします。

```
# init 0
```

3. 「install」フラグを付けずにネットワークからシステムをブートします。

```
ok boot net:dhcp
```

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

```
ok boot net
```

x86: システムのネットワークからのブート

このセクションでは、次の情報について説明しています。

- [109 ページの「システムをネットワークからブートするための要件」](#)
- [110 ページの「GRUB 2 PXE ブートイメージがインストールされる場所」](#)
- [111 ページの「UEFI および BIOS ファームウェアを搭載するシステムのネットワークからのブート」](#)
- [112 ページの「システムをネットワークからブートする方法」](#)

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、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。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。ネットワークアダプタファームウェアが Preboot eXecution Environment (PXE) 仕様をサポートする x86 ベースのシステムを使用して、Oracle Solaris をブートできます。GRUB 2は、Oracle Solaris カーネルをロードし、ブートプロセスを続行するために使用される PXE Network Bootstrap (NBP) プログラムです。

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。

ブート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。AI サーバーを設定している場合は、そのシステムも DHCP サーバーにできます。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 11.3 での DHCP の作業』](#)を参照してください。

x86: システムをネットワークからブートするための要件

ネットワークから x86 ベースのシステムをブートするときは、次の情報に注意してください。

- 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。
- Oracle Solaris のネットワークブートでは、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関係なく、ネットワーク経由でブートプログラムをロードするメカニズムを提供する PXE ファームウェアインタフェースを使用します。このファームウェアは、BIOS ファームウェアを搭載するシステムの場合に `pxegrub2` および 64 ビット UEFI ファームウェアを搭載するシステムの場合に `grub2netx86.efi` という名前の特別に構築された GRUB 2 イメージであるブートプログラムのロードを担当します。これらのファイルには、ネットワークでパケットを転送するために、Trivial File Transfer Protocol (TFTP)、DHCP、User Datagram Protocol (UDP)、インターネットプロトコル (IP)、および Universal Network Device Interface (UNDI) ファームウェアインタフェース (BIOS システムの場合) または Simple Network Protocol (SNP) インタフェース (UEFI システムの場合) のいずれかを使用するミニドライバの基本実装が含まれます。
- GRUB 2 は、GRUB Legacy の PXE ベースのネットワークブートと同様のメカニズムを使用します。GRUB 2 の PXE ブートイメージには、GRUB を初期化するために必要なコードとモジュール、ZFS からブートするために必要なファイルシステムモジュール、および多くの有益な GRUB コマンドが含まれています。ネットワーク経由でモジュールをロードすると、ネットワークリソースに不必要な負荷がかかり、必須のコマンドを使用できないような障害に PXE ブートプロセスがさらされることがあるため、GRUB コマンドを実装するモジュールが、TFTP サーバー上に残される代わりに、GRUB 2 PXE イメージに組み込まれています。
- GRUB 2 ブートイメージには、GRUB Legacy に存在するのと同じ検索アルゴリズムを実装する、埋め込みの `grub.cfg` ファイルが含まれています。このアルゴリズムは、TFTP サーバー上のいくつかの場所で、オペレーティングシステムのブートに使用する `grub.cfg` ファイルを検索します。
- GRUB Legacy と同様に、GRUB 2 の PXE ブートイメージは、TFTP サーバーのルートディレクトリにインストールされます。PXE ブートイメージの名前は、自動インストーラ (AI) の構成方法によって異なります。AI ドキュメントに従って、該当する DHCP `BootFile` マクロには、PXE ブートイメージの名前が含まれます。
- `installadm` コマンドは、無条件に BIOS および UEFI PXE イメージを TFTP サーバー上の適切な場所にコピーするように変更されました。さらに、DHCP サーバーは、UEFI ファームウェアを実行しているシステムに、PXE ブート時に正しい GRUB 2 (UEFI) `BootFile` オプションが与えられるように、ブートクライアントによって適切なシステムアーキテクチャタグが送られた場合に、適切な

BootFile マクロを返すことができる必要もあります。この情報は、DHCP サーバーが DHCPPOFFER を送信した場合に提供されます。

インストールされた Oracle Solaris インスタンスで、PXE ブートイメージは /boot/grub/pxegrub2 ファイル (BIOS 対応のイメージの場合) および /boot/grub/grub2netx64.efi ファイル (64 ビット UEFI 対応のイメージの場合) に保存されます。

AI を使用して Oracle Solaris をインストールするためにネットワークからシステムをブートする場合、詳細については、『[Oracle Solaris 11.3 システムのインストール](#)』を参照してください。

DHCP サーバーは、DHCP クラス PXEClient に次の情報で応答できる必要があります。

- ファイルサーバーの IP アドレス
- ブートファイルの名前。これは BIOS ファームウェアを搭載するシステムの場合 pxegrub2 で、UEFI ファームウェアを搭載するシステムの場合 grub2netx64.efi です。

ネットワークから PXE ブートを実行するシーケンスは次のようになります。

1. ファームウェアが、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. ファームウェアが DHCP リクエストを送信します。
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. ファームウェアは、TFTP を使用して、pxegrub2 (または grub2netx64.efi) をダウンロードし、次に GRUB 2 イメージを実行します。
5. TFTP を使用して GRUB 構成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: GRUB 2 PXE ブートイメージがインストールされる場所

GRUB Legacy と同様に、GRUB 2 の PXE ブートイメージは、TFTP サーバーのルートディレクトリにインストールされます。ブートイメージの名前は、AI の構成方法によって異なります。該当する DHCP BootFile オプションには PXE ブートイメージの名前が含まれます。AI イメージが GRUB 2 ベースの場合、BIOS と UEFI の両方のファームウェアタイプが自動的にサポートされます。特殊な引数は必要ありません。

インストールされた Oracle Solaris インスタンスで、BIOS 対応のイメージと UEFI 対応のイメージの両方の PXE ブートイメージが AI イメージのルートディレクトリの boot/

grub、たとえば `/export/auto_install/my_ai_service/boot/grub` に保存されます。

このディレクトリには、次の内容が格納されます。

```
bash-4.1$ cd grub/
bash-4.1$ ls
grub_cfg_net i386-pc splash.jpg x86_64-efi
grub2netx64.efi pxegrub2 unicode.pf2
```

GRUB 2 モジュールには、ファームウェア固有のサブディレクトリがあり、それらは、BIOS ファームウェアを搭載するシステムの場合 `i386-pc` ディレクトリ内で、64 ビット UEFI システムの場合 `x64_64-efi` ディレクトリ内にあります。ただし、これらのディレクトリ内のファイルは、ネットワークブート時に使用されません (モジュールは GRUB 2 イメージに組み込まれ、TFTP 経由で転送されません)。

注記 - `installadm` コマンドによって管理されていない DHCP サーバーを使用している場合、クライアントアーキテクチャー識別子に基づいて `BootFile` を設定するという `installadm` コマンドでアクセス可能な DHCP サーバーを構成する通常の方法に従って、DHCP サーバーを構成する必要があります。管理者の支援のため、`installadm` コマンドは、手動で構成された DHCP サーバーに設定するクライアントアーキテクチャーブートファイルパスを出力します。

x86: UEFI および BIOSファームウェアを搭載するシステムのネットワークからのブート

ブート可能なネットワークアダプタには、PXE 仕様に準拠したファームウェアが含まれます。PXE ファームウェアは、アクティブになるとネットワーク上で DHCP 交換を実行し、DHCP サーバーが TFTP サーバーからの DHCP 応答に含めた `BootFile` マクロをダウンロードします (TFTP サーバーも DHCP 応答に含まれる)。Oracle Solaris の場合、この `BootFile` マクロ、`pxegrub2` (BIOS ファームウェアを搭載するシステムの場合)、または `grub2netx64.efi` (64 ビット UEFI ファームウェアを搭載するシステムの場合) は GRUB 2 です。GRUB は `unix` カーネルとブートアーカイブのダウンロードに進み、両方をメモリーにロードします。その時点で、Oracle Solaris カーネルに制御が渡されます。

UEFI ファームウェアが搭載されたシステムでのネットワークブートプロセスは、BIOS ファームウェアが搭載されたシステムでのプロセスと非常に似ていますが、UEFI ファームウェアが搭載されたシステムでは DHCP リクエストが若干異なり、UEFI システム用に返される `BootFile` マクロをカスタマイズするのに十分な情報が DHCP サーバーに提供されます。UEFI ファームウェアが搭載されたシステムで必要なのは UEFI ブートアプリケーションで、そうでない場合は DHCP サーバーから `BootFile` マクロとして返される BIOS 対応のブートプログラムではありません。`BootFile` マクロ (`grub2netx64.efi` または同等のもの) で指定された UEFI プー

トアプリケーション (GRUB) の UEFI クライアントへのダウンロードが完了すると、次にブートローダー (GRUB) が実行されます。BIOS ネットワークブートのプロセスと同様に、GRUB は unix カーネルおよびブートアーカイブを DHCP で指定された TFTP サーバーからダウンロードし、次に両方をメモリーにロードして、最後に制御を unix カーネルに渡します。

▼ x86: 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。109 ページの「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 11.3 システムのインストール』のパート 3、「インストールサーバーを使用したインストール」を参照してください。
1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。
 2. **BIOS からシステムのリブートを実行します。**

```
# reboot -p
```

高速リブート機能がデフォルトで有効にされているシステムで、-p オプションを指定していなければ、リブート時にファームウェアがバイパスされます。このオプションを指定すると、システムの標準 (低速リブート) が有効になるため、システムのファームウェアユーティリティーにアクセスして、PXE ブートおよびインストールを指定できます。高速リブートの詳細については、96 ページの「リブートプロセスの高速化」を参照してください。
 3. **BIOS または UEFI ファームウェアにネットワークからブートするように指示します。**
 - システムが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からブートする場合は、BIOS または UEFI ファームウェア画面が表示されたらすぐにそのシーケンスを入力します。
たとえば、BIOS ファームウェアを搭載するシステムで F12 キーを押して、セットアップユーティリティーに入ります。
 - ネットワークブートのためにファームウェア設定を手動で変更する必要がある場合は、ファームウェア設定ユーティリティーにアクセスするためのキーストローク

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。

4. **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てから **Return** キーを押し、そのイメージをブートしてインストールします。

システムは、選択した Oracle Solaris インストールイメージのネットワークからのブートとインストールに進みます。インストールの実行には、数分かかることがあります。AI インストールの実行については、『[Oracle Solaris 11.3 システムのインストール](#)』のパート 3, 「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。

ブートプールを含むシステムの管理

ブートプールからブートするシステムの管理に関連したトピックの一覧を次に示します。

- [116 ページの「BE をブートする機能を保持する方法」](#)
- [117 ページの「すべてのブートプールデータセットに対する削除動作の変更」](#)
- [117 ページの「BE をブート可能にする方法」](#)
- [118 ページの「フォールバックイメージを更新する方法」](#)
- [119 ページの「Oracle Solaris 11.3 の OpenBoot プロパティ」](#)
- [119 ページの「os-root-device 変数」](#)

ファームウェアからアクセス不可能なストレージデバイスからのブートの概要

ファームウェアからアクセス不可能なストレージデバイスからブートできるシステム (SPARC M7 シリーズサーバーなど) 上には、ブートプールが作成されます。これらは、ブートプロセスに手順を追加しなければ、ファームウェアが識別できないデバイスです。たとえば、IPoIB を使用している iSCSI デバイスからのブートでは、ファームウェアからアクセス不可能なストレージを使用します。詳細は、[143 ページの「ブートプロセスの変更点」](#)を参照してください。

ブートプール、ブートプールデータセット、およびフォールバックイメージの管理

ハードウェアではアクセスできないストレージデバイスからブートするシステムでは、`bootadm boot-pool` コマンドを使用してブートプールを管理できます。各オプションには、特定のルートプールを指定するための `-p rpool` オプションを含めることができます。詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。次のオプションを指定できます。

add	ブートプールに新しいデバイスを追加します。
list	ブートプールの構成設定に関する情報を表示します。
remove	ブートプールからデバイスをただちに削除します。
resync	ブートプールを再同期し、特定のブート環境のためのブート可能データセットを作成します。
set	ブートプールのパラメータを変更します。現在は、 <code>eviction_algorithm</code> パラメータのみを変更できます。

デフォルトでは、最後にブートされた BE に関連付けられているブートプールデータセットはブートプール内に残ります。ブートプールの領域が不足すると、最近ブートされていないデータセットが削除されます。ブートプールデータセットが自動的に削除されないようにする場合は、[116 ページの「BE をブートする機能を保持する方法」](#)を参照してください。また、すべてのブートプールデータセットに対するシステムのデフォルトの動作を変更することもできます。詳細は、[117 ページの「すべてのブートプールデータセットに対する削除動作の変更」](#)を参照してください。

注記 - このプロセスによって削除されるのは、BE に関連付けられたデータセットだけです。BE は影響を受けません。

▼ BE をブートする機能を保持する方法

デフォルトでは、ブートプールのメモリーが不足しても、最後にブートされた BE に関連付けられているデータセットは保持されます。この手順では、BE のブートプールデータセットを保持して、その BE をブート可能なままにします。

1. 保持する BE を識別します。

```
# beadm list
BE          Flags Mountpoint Space  Policy Created
--          - - - - -
BE1         - -           6.13M  static 2014-10-09 17:21
BE2         - -           52.86M static 2015-01-03 16:22
BE3         NR /           313.1M static 2015-02-04 17:36
```

2. BE のポリシーを変更します。

この例では、BE2 という名前のブート環境が保持されます。

```
# beadm set-policy -p noevict BE2
```

3. ポリシーが変更されたことを確認します。

BE2 エントリへの `noevict` ポリシーの追加は、BE2 が保持されることを示します。

```
# beadm list
BE          Flags Mountpoint Space  Policy      Created
--          -
BE1         -      -           6.13M  static      2014-10-09 17:21
BE2         -      -           52.86M noevict,static 2015-01-03 16:22
BE3         NR     /           313.1M static       2015-02-04 17:36
```

すべてのブートプールデータセットに対する削除動作の変更

BE データセットは最後のブートの順序で保持されるため、最近ブートされた BE に関連付けられているデータセットは保持されます。デフォルトでは、ブートプールがいっぱいになると、最近ブートされていないデータセットが削除されます。この動作を変更するには、次のコマンドを入力します。

```
# bootadm boot-pool set eviction_algorithm=none
```

これにより、データセットがブートプールから削除されなくなります。ただし、ブートプールがいっぱいになると、そのブートプールに情報を追加する操作は失敗します。ブートプールに情報を追加する操作には、次のものが含まれます。

- 新しい BE の作成 (多くの場合、pkg 操作によって実行されます)
- データセットがそのブートプール内にはない BE のアクティブ化
- BE に関するポリシーの noevict への変更

▼ BE をブート可能にする方法

ファームウェアからアクセス不可能なストレージデバイスからシステムがブートされない場合は、すべての BE がブート可能です。ただし、IPoIB を使用している (たとえば、SPARC M7 シリーズサーバー上の) iSCSI ブートデバイスからブートすると、BE に関連付けられたデータセットがブートプールから削除されている可能性があります。この場合、その BE はブート可能でなくなっています。次の手順を使用すると、データセットを復元し、BE をブート可能にできます。

1. ブート不可能な BE を識別します。

この例では、BE1 のフラグのリストに ! が含まれています。これは、BE1 がブート可能でないことを示しています。ここで、N フラグは BE3 が現在ブートされていることを示し、R フラグは、次のリブート中に BE3 が使用されることを示しています。

```
# beadm list
BE          Flags Mountpoint Space  Policy      Created
--          -
BE1         !-    -           6.13M  static      2014-10-09 17:21
BE2         -     -           52.86M static      2015-01-03 16:22
```

```
BE3          NR      /          313.1M  static 2015-02-04 17:36
```

2. **BE をアクティブ化します。**

このコマンドは、この BE をリブートでアクティブになるように設定し、ブート用にブートプールデータセットを構成します。

```
# beadm activate BE1
```

3. **BE がブート可能であることを確認します。**

この例では、BE1 は現在、その BE のフラグのリストに !がないためブート可能になっています。

```
# beadm list
BE          Flags Mountpoint Space  Policy      Created
--          -
BE1         R      -          6.13M  static      2014-10-09 17:21
BE2         -      -          52.86M  noevict     2015-01-03 16:22
BE3         N      /          313.1M  static      2015-02-04 17:36
```

4. **(オプション) 次のリブートでブートされる BE を再設定します。**

次回 OS がリブートしたときに BE1 からブートしたくない場合は、アクティブな BE を再設定します。

```
# beadm activate BE3
# beadm list
BE          Flags Mountpoint Space  Policy      Created
--          -
BE1         -      -          6.13M  static      2014-10-09 17:21
BE2         -      -          52.86M  noevict     2015-01-03 16:22
BE3         NR     /          313.1M  static      2015-02-04 17:36
```

▼ フォールバックイメージを更新する方法

フォールバックミニルートがインストールされているサービスプロセッサ (SP) を備えたシステムでは、フォールバックイメージを更新するには、まずそのイメージを HTTP サーバーにダウンロードする必要があります。その後、HTTP サーバー上のフォールバックイメージが SP にロードされます。

1. **MOS からイメージをダウンロードします。**

<https://support.oracle.com> の Web ページから開始します。

- a. 「パッチと更新版」タブを選択します。
- b. 右側にある「パッチ検索」領域で、「製品またはファミリー(拡張検索)」をクリックします。
- c. 「製品」フィールドに部分的な製品名を入力します。

- d. 一致する製品名のリストから製品を選択します。
 - e. 「リリース」ドロップダウンメニューで適切なリリースを選択します。
 - f. 「リリース」プルダウンメニューから、システムにインストールされている OS バージョンに対応するフォールバックイメージを選択します。
 - g. そのフォールバックイメージを HTTP サーバーにダウンロードします。
2. ILOM プロンプトで、現在のイメージのバージョンをチェックして、最新のバージョンがまだインストールされていないことを確認します。

```
-> show /SP/firmware/host/miniroot version

/SP/firmware/host/miniroot
Properties:
  version = fallback-5.11-0.175.2.9.0.5.0
```

3. ILOM プロンプトで、フォールバックイメージを更新します。

```
-> load -source http://webserver.example.com/fallback/fallback.pkg
```

Oracle Solaris 11.3 の OpenBoot プロパティ

新しい OpenBoot プロパティは、システムのブート中に使用できるデバイスに関する情報を提供します。これらのプロパティは自動的に保持されます。

これらのプロパティは読み取り専用であり、`/chosen` を使用して表示できます。

<code>boot-pool-list</code>	ブートプールを構成し、かつブート時に Oracle Solaris が使用するデバイスである、OpenBoot からアクセス可能なストレージデバイスへのデバイスパスを一覧表示します
<code>tboot-list</code>	フォールバックイメージを含むストレージデバイスを一覧表示します

os-root-device 変数

`os-root-device` NVRAM 変数は、ルートプールのデバイスおよびルートファイルシステムを定義します。この変数を表示するには、OpenBoot プロンプトで `printenv` コマンドを使用するか、シェルプロンプトで `eeprom` コマンドを使用します。この変数は自動的に保持されるため、通常、手動での介入は必要ありません。ファームウェア

からアクセス不可能なストレージデバイス上に格納されるルートプールを定義するには、次のキーワードを使用できます。

- `osroot-path`
- `osroot-type`
- `osroot-iscsi-initiator-id`
- `osroot-iscsi-target-ip`
- `osroot-iscsi-port`
- `osroot-iscsi-partition`
- `osroot-iscsi-lun`
- `osroot-iscsi-target-name`
- `osroot-subnet-mask`
- `osroot-host-ip`

例 46 `os-root-device` 変数のキーワードデータの表示

この例では、`os-root-device` 変数に関連付けられたデータを表示します。この出力では、テキストを読みやすくするために、1行に1つのキーワードのペアが表示されています。通常、出力は1つの長い文字列です。

```
# eeprom os-root-device
os-root-device=osroot-type=ZFS/iSCSI/IPv4/IPoIB;
osroot-iscsi-port=3260;
osroot-iscsi-target-ip=168.168.1.2;
osroot-iscsi-partition=a;
osroot-iscsi-initiator-id=iqn.1986-03.com.sun:01:0010e05db261.550b268b;
osroot-iscsi-lun=5;
osroot-iscsi-target-name=iqn.1986-03.com.sun:02:fb3685b9-883d-460a-b817-8ea0d8c023dc;
osroot-subnet-mask=255.255.255.0;osroot-host-ip=168.168.1.156;
osroot-path=/pci@314/pci@1/pciex15b3,1003@0:port=1,pkey=FFFF,protocol=ip
```

▼ 通常のブートを有効にする方法

ファームウェアからアクセス不可能なストレージデバイスからブートするようにシステムが構成されると、そのシステムは引き続きその構成でブートします。別のデバイスを使用してブートする場合は、`os-root-device` 変数を変更します。

1. **`os-root-device` プロパティをチェックします。**

```
# eeprom os-root-device
```

2. **`os-root-device` プロパティをクリアします。**

```
# eeprom os-root-device=
```

3. **プロパティが変更されたことを確認します。**

```
# eeprom os-root-device
```

次の手順 ファームウェアからアクセス不可能なストレージデバイスからシステムをリポートする必要がある場合は、ブートプールからシステムをブートします。その時点からシステムが引き続き iSCSI デバイスを使用してブートするように、`os-root-device` 変数はブートプロセス中に自動的に更新されます。

システムのブートに関するトラブルシューティング

この章では、システムのブートを妨げる問題や、復旧の目的でシステムをシャットダウンし、リブートする必要がある問題のトラブルシューティング方法について説明します。この章で SPARC または x86 ベースのシステムのみに適用される情報については、その旨が示されています。

この章の内容は次のとおりです。

- 123 ページの「Oracle Solaris ブートアーカイブの管理」
- 127 ページの「復旧目的のシステムのシャットダウンおよびブート」
- 134 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」
- 137 ページの「カーネルデバッグ (kmdb) を有効にしてシステムをブートする」
- 139 ページの「高速リブートに関する問題のトラブルシューティング」
- 141 ページの「ブートとサービス管理機構の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管理

このセクションでは、次の情報について説明しています。

- 124 ページの「ブートアーカイブの内容を一覧表示する方法」
- 124 ページの「boot-archive SMF サービスの管理」
- 125 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」
- 126 ページの「高速リブートをサポートしないシステムの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概要については、16 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。

x86 プラットフォームで、`bootadm` コマンドは、ブートローダーの管理に加えて、SPARC および x86 Oracle Solaris ブートアーカイブの両方を保守するための次のタスクを実行する場合にも使用します。

- 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。

▼ ブートアーカイブの内容を一覧表示する方法

1. `root` 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

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
```

`list-archive` 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します。

boot-archive 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うち破壊され、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 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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。

▼ 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

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

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
```

注記 - 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: 高速リブートをサポートしないシステムのブートアーカイブの自動更新障害を解決する方法

システムのリブートのプロセス中に、システムが高速リブート機能をサポートしていない場合、ブートアーカイブの自動更新が失敗することがあります。この問題により、システムが同じブート環境からリブートできなくなることがあります。

この場合、次のような警告が表示され、システムは保守モードに入ります。

```
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.
```

svc:/system/boot-config:default SMF サービスには、`auto-reboot-safe` プロパティー (デフォルトで `false` に設定) が含まれます。このプロパティーを `true` に設定すると、システムのファームウェアとデフォルトの GRUB メニューエントリの両方が現在のブートデバイスからブートするように設定されます。次の手順で説明するように、失敗したブートアーカイブの自動更新をクリアできるように、このプロパティーの値を変更できます。

1. root 役割になります。

『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. システムをリブートします。

```
# reboot
```

3. アクティブな BIOS または UEFI 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
```

復旧目的のシステムのシャットダウンおよびブート

このセクションでは次の手順について説明します。

- [128 ページの「復旧を目的としてシステムを停止する方法」](#)
- [129 ページの「復旧目的でのシステムの停止とリブートの方法」](#)
- [130 ページの「シングルユーザー状態でブートして、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」](#)
- [131 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」](#)
- [133 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 GRUB 構成の問題を解決する方法」](#)

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。
- カーネルデバッガを使ってシステムをブートします。

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。

- シングルユーザー状態に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 AI 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、失われた root パスワードを回復します。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。
- **x86 のみ:** 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。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。

▼ SPARC: 復旧を目的としてシステムを停止する方法

1. `shutdown` または `init 0` コマンドを使用して、システムを `ok PROM` プロンプトにします。

2. ファイルシステムを同期させます。

```
ok sync
```

3. 該当する `boot` コマンドを入力して、ブートプロセスを起動します。

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

4. システムが指定した実行レベルになっていることを確認します。

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

5. 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。

例 47 システムの電源切断

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

```
# shutdown -g0 -i0 -y
# svc.startd: The system is coming down. Please wait.
svc.startd: 91 system services are now being stopped.
Jun 12 19:46:57 wgs41-58 syslogd: going down on signal 15
svc.startd: The system is down.
syncing file systems...done
```

```

Program terminated
r)ebboot o)k prompt, h)alt?
# 0

ok #.
->

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

->

```

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

例 48 システムの電源投入

次の例は、サービスプロセッサを使用するシステムの電源を投入する方法を示しています。最初に Oracle ILOM にログインする必要があります。<http://download.oracle.com/docs/cd/E19166-01/E20792/z40002fe1296006.html#scrolltoc> を参照してください。

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

```

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

->

```

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

▼ x86: 復旧目的でのシステムの停止とリブートの方法

1. root 役割になります。
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. キーボードやマウスが動作している場合は、`init 0` と入力して、システムを停止します。
init 0
3. 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。

▼ シングルユーザー状態でブートして、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. root 役割になります。
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. プラットフォームに応じて、次のいずれかを実行します。
 - SPARC プラットフォームの場合:
 - a. システムを ok PROM プロンプトにします。

```
# init 0
```
 - b. システムをシングルユーザー状態にブートします。

```
ok boot -s
```
 - x86 プラットフォームの場合:
 - a. reboot コマンドの -p オプションによって、稼働中のシステムをリブートします。

```
# reboot -p
```
 - b. GRUB メニューが表示されたら、該当するブートエントリを選択し、e を入力してそのエントリを編集します。
 - c. 矢印キーを使用して、\$multiboot 行に移動し、行の末尾に -s と入力します。
 - GRUB 編集メニューを終了し、編集したばかりのエントリをブートするには、Control-X を押します。UEFI ファームウェアを搭載するシステムがあり、シリアルコンソールを使用していない場合、F10 キーを押すことによってもエントリがブートします。
3. /etc/passwd ファイルのシェルエントリを修正します。

```
# vi /etc/password
```

4. 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 11.3 での ZFS ファイルシステムの管理](#)』の「[ZFS ルートプール内のディスクの交換](#)」を参照してください。

1. 次のいずれかのオプションを使用して、Oracle Solaris メディアからブートします。

- **SPARC: テキストインストール - インストールメディアまたはネットワークからブートして、テキストインストール画面からシェルオプション (オプション 3) を選択します。**

- **SPARC: 自動インストール - 次のコマンドを使用して、シェルに出られるインストールメニューから直接ブートします。**

```
ok boot net:dhcp
```

- **x86: 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。**

- **x86: 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シェルオプション (オプション 3) を選択します。**

- **x86: 自動インストール - ネットワーク上の AI サーバーからブートします。この方法では、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シェルオプション (オプション 3) を選択します。**

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. ブート環境をマウントポイント /a にマウントします。

```
# beadm mount solaris-instance|be-name /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. シングルユーザー状態にシステムをリブートし、root パスワードの入力を求められたら、Return キーを押します。

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。

参照 メディアからシステムをブートする必要がある問題が GRUB 構成に発生した場合は、この手順にある x86 プラットフォーム用と同じ手順に従います。

▼ x86: 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 GRUB 構成の問題を解決する方法

x86 ベースのシステムがブートしない場合、問題の原因は、破損したブートローダーまたは失われているか破壊された GRUB メニューにある可能性があります。これらの種類の状況では次の手順を使用します。

注記 - 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 11.3 での ZFS ファイルシステムの管理](#)』の「[ZFS ルートプール内のディスクの交換](#)」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シェルオプション (オプション 3) を選択します。
- **自動インストール - ネットワーク上の AI 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シェルオプション (オプション 3) を選択します。

2. ルートプールをインポートします。

```
# zpool import -f rpool
```

3. GRUB 構成の問題を解決するには、次のいずれかの操作を実行します。

- システムがブートせず、エラーメッセージが表示されない場合、ブートローダーが破損している可能性があります。問題を解決するには、[55 ページの「bootadm install-bootloader コマンドを使用して GRUB 2 をインストールする」](#)を参照してください。
- GRUB メニューが失われている場合、ブート時に、「grub.cfg を開けません」というエラーメッセージが表示されます。この問題を解決するには、[34 ページの「GRUBメニューを手動で再生成する方法」](#)を参照してください。

- GRUB メニューが破損している場合、システムがブート時に GRUB メニューを解析しようとして、別のエラーメッセージが表示されることがあります。34 ページの「GRUBメニューを手動で再生成する方法」も参照してください。
4. 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

このセクションでは次の手順について説明します。

- [SPARC: 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](#)
- [x86: 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](#)

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 機能を使用できません。

システムのクラッシュダンプについては、『Oracle Solaris 11.3 でのシステム管理のトラブルシューティング』の「システムのクラッシュダンプの構成」を参照してください。

▼ SPARC: 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

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

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

```
> n
ok sync
```

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

3. システムが実行レベル 3 でブートしていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

例 49 SPARC: halt -d コマンドを使用してシステムのクラッシュダンプとリポートを強制実行する

この例は、halt -d コマンドを使用して、SPARC ベースの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する方法を示しています。

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

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

000002a1008f7860 genunix:kadmin+438 (b4, 0, 0, 0, 5, 0)
%l0-3: 0000000000000000 0000000000000000 0000000000000004 0000000000000004
%l4-7: 000000000000003cc 0000000000000010 0000000000000004 0000000000000004
000002a1008f7920 genunix:uadmin+110 (5, 0, 0, 6d7000, ff00, 4)
%l0-3: 0000030002216938 0000000000000000 0000000000000001 000004237922872
%l4-7: 00000423791e770 0000000000004102 000003000449308 0000000000000005

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

.
.
Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: kernel/sparcv9/unix
configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: jupiter
The system is coming up. Please wait.
NIS domain name is example.com

.
.
System dump time: Wed Jul 21 14:13:41 2013
Jul 21 14:15:23 jupiter savecore: saving system crash dump
in /var/crash/jupiter/*.0
Constructing namelist /var/crash/jupiter/unix.0
Constructing corefile /var/crash/jupiter/vmcore.0
100% done: 5339 of 5339 pages saved

.
.
.
```

▼ x86: クラッシュダンプを強制してシステムをリポートする方法

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グがロードされている必要があります。

注記 - カーネルデバグ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. カーネルデバグにアクセスします。

デバ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 50 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ポートを強制実行する

この例は、halt -d コマンドを使用して、x86 ベースの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_syscall132+101 ()

syncing file systems... done
dumping to /dev/dsk/c1t0d0s1, offset 107675648, content: kernel
NOTICE: adpu320: bus reset
```

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

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

カーネルデバッガ (kldb) を有効にしてシステムをブートする

このセクションでは次の手順について説明します。

- [137 ページの「カーネルデバッガ \(kldb\) を有効にしてシステムをブートする方法」](#)
- [138 ページの「カーネルデバッガ \(kldb\) を有効にしてシステムをブートする方法」](#)

システムの問題をトラブルシューティングする必要がある場合は、カーネルデバッガでシステムを実行するとたいへん役立ちます。カーネルデバッガはシステムの停止を調査するのに役立ちます。たとえば、カーネルデバッガがアクティブな状態で、カーネルを実行しているときに停止が発生した場合、デバッガに割り込んでシステムの状態を調べることができます。さらに、システムパニックが発生した場合、システムがリブートする前に、パニックを調査できます。このようにして、問題の原因となっている可能性のあるコードのセクションを知ることができます。

次の手順では、カーネルデバッガを有効にしてブートすることにより、システムの問題をトラブルシューティングするための基本手順について説明します。

▼ SPARC: カーネルデバッガ (kldb) を有効にしてシステムをブートする方法

この手順では、SPARC ベースのシステムでカーネルデバッガ (kldb) をロードする方法を示します。

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

1. システムを停止します。これにより、`ok` プロンプトが表示されます。システムを正常に停止するには、`halt` コマンドを使用します。

2. `boot -k` と入力して、カーネルデバッグのロードをリクエストします。Return キーを押します。
3. カーネルデバッグにアクセスします。
デバッグに入るための方法は、システムへのアクセスに使用するコンソールのタイプによって異なります。

- ローカル接続されたキーボードを使用している場合は、キーボードの種類に応じて **Stop-A** または **L1-A** を押します。
- シリアルコンソールを使用している場合は、そのシリアルコンソールタイプに適した方法を使用してブレークを送信します。

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

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

例 51 SPARC: カーネルデバッグ (kmdb) を有効にしてシステムをブートする

次の例に、カーネルデバッグ (kmdb) を有効にして SPARC ベースのシステムをブートする方法を示します。

```
ok boot -k
Resetting...

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

▼ x86: カーネルデバッグ (kmdb) を有効にしてシステムをブートする方法

この手順では、カーネルデバッグ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。
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、`$multiboot` 行を選択します。

4. **GRUB 編集メニューで、 \$multiboot 行の末尾に -k を入力します。**
カーネルが実行する前に、デバッガでシステムが停止 (ブレイク) するように指示するには、-d オプションと -k オプションを含めます。
 5. **GRUB 編集メニューを終了し、編集したばかりのエントリをブートするには、Control-X を押します。UEFI ファームウェアを搭載するシステムがあり、シリアルコンソールを使用していない場合、F10 キーを押すことによってもエントリがブートします。**
-k を入力すると、デバッガ (kmdb) がロードされたあと、オペレーティングシステムが直接ブートします。
 6. **カーネルデバッガにアクセスします。**
デバッガ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ガ (kmdb) 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、デバッガ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ガと対話する機会が与えられます。

カーネルデバッガ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 kmdb を使用して、システムと対話する詳細については、[kmdb\(1\)](#)のマニュアルページを参照してください。

x86: 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。

このセクションでは、次の情報について説明しています。

- [140 ページの「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」](#)
- [140 ページの「高速リブートが機能しない可能性のある状況」](#)

高速リブート機能をサポートしない x86 ベースのシステムで Oracle Solaris ブートアーカイブを手動で更新する必要がある場合は、[126 ページの「高速リブートをサポー](#)

[トしないシステムのブートアーカイブの自動更新障害を解決する方法](#)を参照してください。

x86: 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対して依存関係を持つため、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 を一時的に変更できます。

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

x86: 高速リブートが機能しない可能性のある状況

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 構成を処理できない。
- 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)
```

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\(9S\)](#)

ブートとサービス管理機構の問題のトラブルシューティング

システムをブートするときに発生する可能性のある問題は、次のとおりです。

- ブート時にサービスが開始しない。
サービス管理機構 (SMF) サービスの起動に問題があると、システムがブート時に停止することがあります。この種類の問題をトラブルシューティングする場合には、システムをブートするのにどのサービスも開始する必要はありません。詳細は、『[Oracle Solaris 11.3 でのシステムサービスの管理](#)』の「[システムブート時のサービスの起動に関する問題を調査する方法](#)」を参照してください。
- ブート時に `system/filesystem/local:default` SMF サービスが失敗する。
システムのブートに必要なローカルファイルシステムは、`svc:/system/filesystem/local:default` サービスによってマウントされます。それらのすべてのファイルシステムをマウントできない場合、サービスは保守状態になります。システムの起動は続行し、`filesystem/local` に依存していないサービスが起動します。その後、依存関係によって、`filesystem/local` をオンラインにする必要があるサービスは起動しません。この問題の回避策は、システムの起動を続行させる代わりに、サービスの失敗後ただちに、`sulogin` プロンプトが表示されるようにシステムの構成を変更することです。詳細は、『[Oracle Solaris 11.3 でのシステムサービスの管理](#)』の「[ブート中にローカルファイルシステムサービスが失敗した場合に、シングルユーザーログインを強制する方法](#)」を参照してください。

インストール後のブートに関する問題

以前に MBR パーティション分割を使用していた GPT ラベル付きディスクにインストールしたあと、そのディスクから x86 ファームウェアがブートできない場合があります。これは、BIOS の実装で、BIOS モードでブートするには、少なくとも 1 台のハードディスクにブート可能としてマークされた MBR パーティションが少なくとも 1 つ存在する必要がある場合に発生します。この問題は、UEFI 仕様では、EFI 保護 MBR パーティションをアクティブとしてマークすることが禁止されているために発生します。この状況を解決するには、ブートを試みる前に、`fdisk` コマンドを使用してパーティションのステータスをアクティブに設定します。



Oracle Solaris ブートプロセス

この付録の内容は、次のとおりです。

- [143 ページの「ブートプロセスの変更点」](#)
- [144 ページの「フォールバックイメージからのブート」](#)

ブートプロセスの変更点

ブートプロセスは、異なるタイプのデバイスを使用できるように拡張されています。以前のリリースでは、ブートプロセスは次のように実行されました。

1. ブートプロセスがブートストラップで始まります。これは、ブートプログラムをロードする自動的な手順です。
2. ブートアーカイブがロードされます。
3. ルートプールがインポートされ、ルートファイルシステムがマウントされます。
4. `init` が実行されます。これが次に、すべてのサービスを起動する `svc.startd` を起動します。

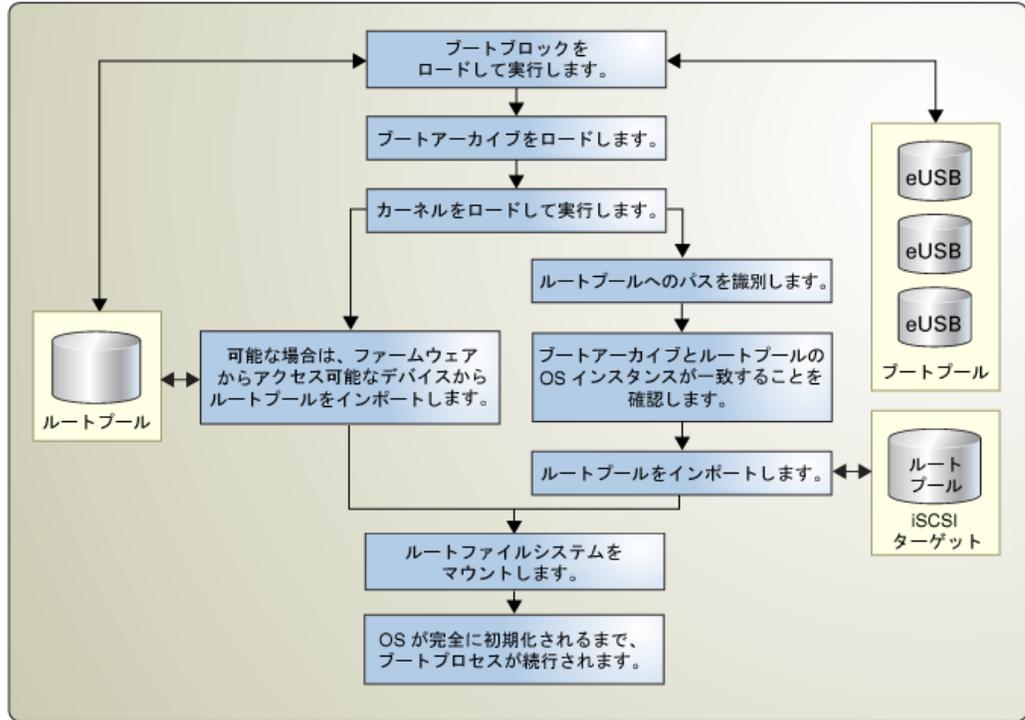
ブートプールがルートプールから分離されている、ファームウェアからアクセス不可能なストレージデバイスからブートするシステムでは、ブートプログラムとブートアーカイブがロードされたあとのブートプロセスは次のように実行されます。

1. ブートプールが識別され、インポートされます。
2. ルートプールへのパスが識別されます。ソフトウェアは、ブートアーカイブとルートファイルシステム内の OS インスタンスが一致するか確認します。

次の図は、この一連の手順を示しています。

注記 - SPARC M7 シリーズサーバーでは、ブートプールは 1 つまたは複数の eUSB デバイスで構成されます。SPARC T7 シリーズサーバーでは、ブートプールは 1 つの eUSB デバイスで構成されます。

図 1 Oracle Solaris ブートプロセス



フォールバックイメージからのブート

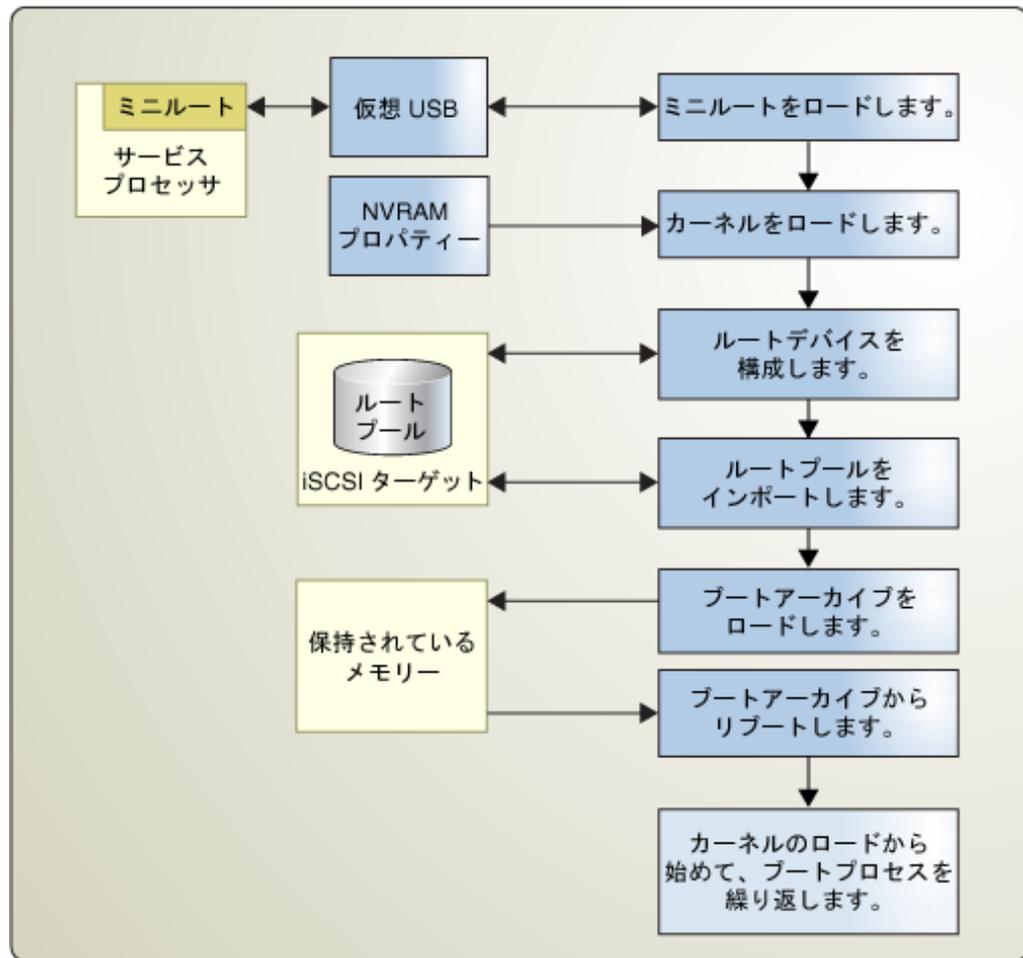
ファームウェアからアクセス不可能なストレージデバイスからのブートに加えて、一部の新しいサーバーでは、ブートプールが使用できない場合にフォールバックイメージを使用する機能が提供されます。フォールバックプロセスは、1つまたは複数のサービスプロセッサ上に格納されている、ミニルートと呼ばれる Oracle Solaris OS の小さなバージョンに依存します。これらのサーバーには、フォールバックイメージが事前に構成されています。フォールバックイメージを更新する必要がある場合は、118 ページの「[フォールバックイメージを更新する方法](#)」を参照してください。

フォールバックブートプロセス中に、ミニルートからカーネルがロードされると、ルートデバイスが構成されます。この構成には、iSCSI デバイス上のルートプールの場所へのポインタが含まれています。次に、ルートプールがインポートされ、保持されているメモリーにブートアーカイブがロードされます。その後、システムがブート

アーカイブからリブートされ、カーネルのロードから始めて、ブートプロセスが繰り返されます。ただし、2 番目のブートでのカーネルはフォールバックイメージ内のミニルートからではなく、ブートアーカイブからロードされます。

このプロセスを示したのが次の図です。

図 2 フォールバックブートプロセス



索引

か

- カーネルデバッグ (kmdb)
 - 高速リブートでの開始, 99
 - システムのブート, 138
- カーネルの選択
 - 高速ブートでの開始, 99
- カーネルブート引数
 - 設定, 77
- 開始
 - 高速リブート, 98
- 確認
 - 実行レベル (方法), 81
- クラッシュダンプとリブートの強制的な実行
 - halt -d, 136
 - トラブルシューティング, 134
- 構成ファイル
 - GRUB 2
 - grub.cfg, 23
- 高速リブート
 - quiesce 関数, 97
 - アクティブになったブート環境への開始, 98
 - 開始, 98
 - 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, 140
 - 問題のトラブルシューティング, 139
- 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, 140
- 高度な GRUB 管理, 54
- コンソールデバイス
 - 設定, 78
- コンソール EEPROM パラメータ
 - 表示, 75

さ

- 削除
 - GRUB メニューエントリ, 42
 - UEFI パラメータ, 78
- システムシャットダウンコマンド, 60
- システムのクラッシュダンプとリブート
 - 強制的な実行, 134
- 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, 60
 - ガイドライン, 60
- システムのリブート
 - クラッシュダンプの強制的な実行, 134
- 実行レベル
 - 0 (電源切断レベル), 79
 - 1 (シングルユーザーレベル), 79
 - 2 (マルチユーザーレベル), 79
 - 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, 80
 - 3 (NFS を持つマルチユーザー), 79
 - s または S (シングルユーザーレベル), 79
- 確認 (方法), 81
 - システムをマルチユーザー状態にブートする, 81
 - 使い分け, 81
 - 定義, 79
 - デフォルトの実行レベル, 79
- 実行レベルとマイルストーンの使い分け, 81
- シングルユーザー状態
 - 高速ブートでの開始, 99
 - システムをブートする
 - 実行レベル S, 83
- シングルユーザーレベル 参照 実行レベル s または S

正常なシャットダウン, 60
設定

- EEPROM パラメータ, 77
- カーネルブート引数, 77
- コンソールデバイス, 78
- ブート時にブート属性を, 43
- ブートデバイス順序, 78

早期システムパニック
デバッグ
高速リブート, 140

た

代替ブート環境

- 高速リブートの開始, 99

タスクの比較

- GRUB Legacy と比較した GRUB 2, 27

チェーンローダーエントリ

- GRUB メニュー, 52

追加

- GRUB メニューエントリ, 41

ディスクの選択

- 高速ブートでの開始, 99

データセットの選択

- 高速ブートでの開始, 99

デバイスドライバ

- quiesce 関数, 97

デバイスの命名スキーム

- GRUB 2, 24

デバイス命名スキーム

- GRUB 2, 24

デバッグ

- 高速リブートに関する問題, 139
- 高速リブートによる早期システムパニック, 140

デフォルトの実行レベル, 79

トラブルシューティング

- kmdb コマンドとブート, 138
- クラッシュダンプの強制的な実行, 134
- 高速リブート, 139, 140

は

パーティション命名スキーム

- GRUB 2, 24

パニック

- 高速リブートのデバッグ, 140

パラメータ

- UEFI 対応システム上での変更, 75

表示

- EEPROM パラメータ, 75
- UEFI EEPROM パラメータ, 76

ブート

- ガイドライン, 12
- 対話式 (方法), 88
- ネットワークからの x86 システム, 108

ブート環境

- 高速リブートの開始, 98
- 代替ブート環境の高速リブートの開始, 99

ブートする

- 実行ベル S に, 83
- 実行レベル 3 (マルチユーザー) に, 81

ブート属性 (x86 プラットフォーム)

- ブート時に変更, 43

ブートデバイス順序

- 設定, 78

ブート動作

- GRUB メニューでの変更方法, 93

変更

- UEFI パラメータ, 75

編集

- ブート時に GRUB メニューを, 43

保守

- GRUB メニュー, 35
- 同じシステム上での GRUB 2 と GRUB Legacy, 53

ま

マイルストーン

- 使い分け, 81

マルチユーザーブート, 81

マルチユーザーレベル 参照 実行レベル 3

A

auto_boot EEPROM パラメータ

- 設定, 77

B

banner コマンド (PROM), 70
 boot-args EEPROM パラメータ
 設定, 77
 bootadm generate-menu
 grub.cfg ファイルの再生成, 34
 bootadm set-menu
 例, 35
 bootadm コマンド
 GRUB の管理, 30
 GRUB メニューエントリの削除, 42
 GRUB メニューエントリの設定, 38
 GRUB メニューエントリの追加, 41
 サブコマンド, 31
 BootOrder EEPROM パラメータ
 設定, 78

C

console EEPROM パラメータ
 設定, 78
 custom.cfg
 GRUB 構成
 カスタマイズ, 47

E

eeeprom コマンド
 UEFI パラメータの削除, 78
 -u オプション, 75
 概要, 74
 パラメータの設定, 77
 パラメータの表示, 75
 EEPROM パラメータ
 1 つ設定, 77
 1 つ表示, 75
 すべての UEFI パラメータの表示, 76
 すべて表示, 75
 EEPROM 引数
 カーネルブート引数の設定, 77

G

GRand Unified Bootloader 参照 GRUB

grub.cfg ファイル
 再生成, 34
 説明, 23
 GRUB 2
 GRUB Legacy からのアップグレード, 48
 GRUB Legacy 情報が変換される方法, 52
 GRUB Legacy との比較, 27
 GRUB Legacy の保守, 53
 概要, 21
 構成ファイル, 23
 命名スキーム, 24
 GRUB 2 と GRUB Legacy の比較, 27
 GRUB 2 へのアップグレード
 GRUB Legacy 情報が変換される方法, 52
 GRUB 2 メニュー
 チェーンロード, 52
 GRUB
 管理の概要, 21
 構成のカスタマイズ, 47
 高度な管理, 54
 GRUB Legacy
 GRUB 2 との比較, 27
 GRUB 2 の保守, 53
 GRUB 2 へのアップグレード, 48
 GRUB 2 への変換, 52
 GRUB Legacy から GRUB 2 へのアップグレード,
 48
 GRUB Legacy から GRUB 2 への変換, 52
 GRUB 構成のカスタマイズ
 custom.cfg, 47
 GRUB のインストール
 高度な GRUB 管理, 54
 GRUB の管理
 bootadm コマンド, 30
 概要, 21
 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, 93
 GRUB メニュー
 手動での再生成, 34
 ブート時に編集, 43
 保守, 35
 GRUB メニューエントリ
 削除, 42
 属性の設定, 38
 追加, 41

GRUB メニューエントリの属性を設定する (方法), 38
GRUB メニューでカーネルの使用法を変更, 93
GRUB メニューの手動での再生成, 34
GRUB を管理するためのコマンド
 bootadm, 30
GRUBClient
 x86 ベースのネットワークブート, 108

H

halt -d
 クラッシュダンプとリブートの強制的な実行, 136
halt コマンド, 61

I

init コマンド
 説明, 61
init 状態 参照 実行レベル

K

kldb コマンド, 138
kldb の有効化
 トラブルシューティング, 138

P

poweroff コマンド, 61
PXEClient
 x86 ベースのネットワークブート, 108

Q

quiesce 関数
 高速リブートの実装, 97

R

reboot コマンド, 61

S

shutdown コマンド
 システムのシャットダウン (方法), 62
 説明, 61
sync コマンド, 134
sync コマンドによるファイルシステムの同期化, 134

U

UEFI EEPROM パラメータ
 1 つ表示, 77
 すべて表示, 76
UEFI 対応システム
 パラメータの変更, 75

W

who コマンド, 81

X

x86 プラットフォーム
 ブート時に GRUB メニューを編集, 43
x86 プラットフォーム上のネットワークブート, 108