



Solaris 10 11/06 インストールガイド (Solaris Live Upgrade とアップグレードの計画)



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-7815-10
2006年11月

Sun Microsystems, Inc. (以下米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。それらに限定されるものではありません。

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

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者によって開発された素材を含んでいることがあります。

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社の商標、登録商標もしくは、サービスマークです。

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

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

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn8 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。Copyright(C) OMRON Co., Ltd. 1995-2006. All Rights Reserved. Copyright(C) OMRON SOFTWARE Co., Ltd. 1995-2006 All Rights Reserved.

「ATOK for Solaris」は、株式会社ジャストシステムの著作物であり、「ATOK for Solaris」にかかる著作権、その他の権利は株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK」および「推測変換」は、株式会社ジャストシステムの登録商標です。

「ATOK for Solaris」に添付するフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

「ATOK for Solaris」に含まれる郵便番号辞書(7桁/5桁)は日本郵政公社が公開したデータを元に制作された物です(一部データの加工を行なっています)。

Unicode は、Unicode, Inc. の商標です。

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

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

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものへの輸出および再輸出は厳しく禁止されています。

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

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

原典: Solaris 10 11/06 Installation Guide: Solaris Live Upgrade and Upgrade Planning

Part No: 819-6396-10

Revision A

目次

はじめに	11
パート I Solaris Live Upgrade によるアップグレード	15
1 Solaris インストールの計画についての参照先	17
計画とシステム要件についての参照先	17
2 Solaris Live Upgrade (概要)	19
Solaris Live Upgrade の紹介	19
Solaris Live Upgrade の実行手順	20
ブート環境の作成	22
RAID-1 ボリュームファイルシステムを持つブート環境の作成	27
ブート環境のアップグレード	33
ブート環境のアクティブ化	36
元のブート環境へのフォールバック	37
ブート環境の保守	39
3 Solaris Live Upgrade (計画)	41
Solaris Live Upgrade の要件	41
Solaris Live Upgrade のシステム要件	41
Solaris Live Upgrade のインストール	42
Solaris Live Upgrade のディスク容量の要件	45
RAID-1 ボリューム(ミラー)を作成する場合の Solaris Live Upgrade の要件	45
パッケージまたはパッチによるシステムのアップグレード	46
lucreate コマンドを使用したファイルシステムの作成のための指針	47
ファイルシステムのスライスを選択するための指針	48
ルート (/) ファイルシステムのスライスを選択するための指針	48

ミラー化されたファイルシステムのスライスを選択するための指針	49
スワップファイルシステムのスライスを選択するための指針	50
共有可能なファイルシステムのスライスを選択するための指針	51
新しいブート環境の内容のカスタマイズ	52
ブート環境間でのファイルの同期	53
/etc/lu/syncList へのファイルの追加	53
ブート環境間での強制的な同期	54
x86: GRUB メニューを使ったブート環境のアクティブ化	55
リモートシステムからの Solaris Live Upgrade の使用	56
4 Solaris Live Upgrade によるブート環境の作成 (作業)	57
Solaris Live Upgrade インタフェースの概要	57
Solaris Live Upgrade メニューの使用 (CUI)	58
作業マップ: Solaris Live Upgrade のインストールとブート環境の作成	59
Solaris Live Upgrade のインストール	59
Solaris Live Upgrade に必要なパッチのインストール	60
▼ 必要なパッチをインストールするには	61
▼ pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法	61
▼ Solaris インストールプログラムを使用して Solaris Live Upgrade をインストールする方 法	61
Solaris Live Upgrade の起動と停止 (キャラクタユーザーインタフェース)	63
▼ Solaris Live Upgrade メニューを起動する	63
▼ Solaris Live Upgrade のメニューを終了するには	64
新しいブート環境の作成	64
▼ ブート環境を作成する (キャラクタユーザーインタフェース)	64
▼ ブート環境をはじめて作成する (コマンド行インタフェース)	69
▼ ブート環境を作成しファイルシステムをマージする (コマンド行インタフェース) ...	72
▼ ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)	74
▼ ブート環境を作成しスワップを再構成する (コマンド行インタフェース)	76
▼ リストを使用してブート環境を作成しスワップを再構成する (コマンド行インタ フェース)	78
▼ ブート環境を作成し共有可能ファイルシステムをコピーする (コマンド行インタ フェース)	80
▼ 別々のソースから単一のブート環境を作成 (コマンド行インタフェース)	81
▼ Solaris フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェ ース)	83
▼ RAID-1 ボリューム (ミラー) を持つブート環境の作成 (コマンド行インタフェ ース)	83

ス)	85
▼ ブート環境の作成と内容のカスタマイズ (コマンド行インタフェース)	91
5 Solaris Live Upgrade によるアップグレード (作業)	95
作業マップ: ブート環境のアップグレード	95
ブート環境のアップグレード	96
アップグレードのガイドライン	96
▼ ブート環境のオペレーティングシステムイメージをアップグレードする (キャラクター ユーザーインタフェース)	97
▼ ブート環境のオペレーティングシステムイメージをアップグレードする (コマンド行 インタフェース)	98
▼ 複数の CD を使用してオペレーティングシステムイメージをアップグレードする (コ マンド行インタフェース)	100
▼ ブート環境のオペレーティングシステムイメージにパッケージを追加する (コマンド 行インタフェース)	102
▼ ブート環境のオペレーティングシステムイメージにパッチを追加する (コマンド行イ ンタフェース)	103
▼ ブート環境にインストールされているパッケージの情報を取得する (コマンド行イン タフェース)	104
JumpStart プロファイルを使用したアップグレード	105
▼ Solaris Live Upgrade で使用されるプロファイルを作成する	105
▼ Solaris Live Upgrade で使用されるプロファイルをテストする	110
▼ プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行イ ンタフェース)	111
ブート環境への Solaris フラッシュアーカイブのインストール	112
▼ ブート環境への Solaris フラッシュアーカイブのインストール (キャラクターユーザ ーインタフェース)	114
▼ ブート環境へ Solaris フラッシュアーカイブをインストールする (コマンド行インタ フェース)	115
▼ プロファイルを使用した Solaris フラッシュアーカイブのインストール (コマンド行イ ンタフェース)	116
▼ プロファイルキーワードを使用した Solaris フラッシュアーカイブのインストール (コ マンド行インタフェース)	118
ブート環境のアクティブ化	119
ブート環境をアクティブ化するための要件と制限	120
▼ ブート環境のアクティブ化 (キャラクターユーザーインタフェース)	121
▼ ブート環境をアクティブにする (コマンド行インタフェース)	123
▼ ブート環境をアクティブにしてファイルを同期させる (コマンド行インタフェ ース)	124

x86: GRUB メニューを使ったブート環境のアクティブ化	125
▼ x86: GRUB メニューを使ってブート環境をアクティブ化する (コマンド行インタフェース)	126
6 障害回復: 元のブート環境へのフォールバック (作業)	129
SPARC: 元のブート環境へのフォールバック (コマンド行インタフェース)	129
▼ SPARC: 新しいブート環境のアクティブ化に成功した場合のフォールバック	130
▼ SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック	130
▼ SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする	131
x86: 元のブート環境へのフォールバック	132
▼ x86: 新しいブート環境のアクティブ化に成功した場合の GRUB メニューを使ったフォールバック	133
▼ x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューを使ったフォールバック	134
▼ x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューと DVD または CD を使ったフォールバック	136
7 Solaris Live Upgrade ブート環境の管理 (作業)	139
Solaris Live Upgrade 管理作業の概要	139
すべてのブート環境のステータスの表示	140
▼ すべてのブート環境のステータスを表示する (キャラクタユーザーインタフェース)	141
▼ すべてのブート環境のステータスを表示する (コマンド行インタフェース)	141
以前に構成されたブート環境の更新	142
▼ 以前に構成されたブート環境を更新する (キャラクタインタフェース)	142
▼ 以前に構成されたブート環境を更新する (コマンド行インタフェース)	143
スケジュールされた処理 (作成/アップグレード/コピー) の取り消し	144
▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (キャラクタインタフェース)	144
▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (コマンド行インタフェース)	144
ブート環境の比較	145
▼ ブート環境を比較する (キャラクタユーザーインタフェース)	145
▼ ブート環境を比較する (コマンド行インタフェース)	146
非アクティブブート環境の削除	146
▼ 非アクティブブート環境を削除する (キャラクタユーザーインタフェース)	147
▼ 非アクティブブート環境を削除する (コマンド行インタフェース)	147

アクティブブート環境の名前の表示	148
▼ アクティブブート環境の名前を表示する (キャラクターインターフェイス)	148
▼ アクティブブート環境の名前を表示する (コマンド行インターフェイス)	148
ブート環境の名前の変更	149
▼ 非アクティブブート環境の名前を変更する (キャラクターインターフェイス)	149
▼ 非アクティブブート環境の名前を変更する (コマンド行インターフェイス)	150
ブート環境名に関連付ける説明の作成または変更	150
▼ テキストを使用してブート環境名の説明を作成または変更する方法	151
▼ ファイルを使用してブート環境名の説明を作成または変更する方法	151
▼ テキストで記述された説明からブート環境名を確認する方法	152
▼ ファイル内の説明からブート環境名を確認する方法	152
▼ 名前からブート環境説明を確認する方法	153
ブート環境の構成の表示	153
▼ 非アクティブブート環境の構成を表示する (キャラクターインターフェイス)	153
▼ ブート環境の構成を表示する (コマンド行インターフェイス)	154
8 x86: GRUB メニューの menu.lst ファイルの検出 (作業)	155
x86: GRUB メニューの menu.lst ファイルの検出 (作業)	155
▼ GRUB メニューの menu.lst ファイルの検出	156
▼ アクティブな menu.lst ファイルが別のブート環境にある場合の GRUB メニューの menu.lst ファイルの検出	156
▼ Solais Live Upgrade ブート環境がマウントされている場合の GRUB メニューの menu.lst ファイルの検出	157
▼ 使用しているシステムに x86 ブートパーティションがある場合の GRUB メニューの menu.lst ファイルの検出	158
9 Solaris Live Upgrade (例)	161
Solaris Live Upgrade によるアップグレードの例 (コマンド行インターフェイス)	161
必要なパッチをインストールするには	162
アクティブブート環境で Solaris Live Upgrade をインストールする方法	163
ブート環境を作成する方法	164
非アクティブブート環境をアップグレードする方法	164
ブート環境がブート可能か確認する方法	164
非アクティブブート環境をアクティブにする方法	164
(省略可能) ソースブート環境へフォールバックする方法	165
RAID-1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インターフェイス)	168

既存のボリュームから Solaris ボリュームマネージャー RAID-1 ボリュームへの移行例 (コマンド行インタフェース)	172
空のブート環境を作成して Solaris フラッシュアーカイブをインストールする例 (コマンド行インタフェース)	172
空のブート環境を作成する方法	173
新しいブート環境へ Solaris フラッシュアーカイブをインストールする方法	174
新しいブート環境をアクティブ化する方法	174
Solaris Live Upgrade によるアップグレードの例 (キャラクタユーザーインタフェース) ...	174
アクティブブート環境で Solaris Live Upgrade をインストールする方法	175
必要なパッチをインストールするには	176
ブート環境を作成する方法	177
非アクティブブート環境をアップグレードする方法	177
非アクティブブート環境をアクティブにする方法	178
10 Solaris Live Upgrade (コマンドリファレンス)	179
Solaris Live Upgrade のコマンド	179
パート II 付録	181
A 問題発生時の解決方法 (作業)	183
ネットワークインストールの設定に関する問題	183
システムのブートに関する問題	184
メディアからのブート時のエラーメッセージ	184
メディアからのブート時の一般的な問題	185
ネットワークからのブート時のエラーメッセージ	186
ネットワークからのブート時の一般的な問題	189
Solaris OS の初期インストール	190
▼ x86: IDE ディスクの不良ブロックの検査	191
Solaris OS のアップグレード	192
アップグレード時のエラーメッセージ	192
アップグレード時の一般的な問題	194
▼ 問題発生後にアップグレードを継続する方法	195
x86: GRUB を使用する場合の Solaris Live Upgrade に関する問題	196
▼ Veritas VxVm の実行中に Solaris Live Upgrade を使用してアップグレードするとシステムパニックが発生する	198
x86: 既存のサービスパーティションが存在しないシステムでは、デフォルトでサービ	

パーティションが作成されない	200
▼ ネットワークインストールイメージまたは Solaris Operating System DVD からのソフトウェアのインストール	200
▼ Solaris SOFTWARE - 1 CD または ネットワークインストールイメージからのインストール	201
B その他の SVR4 パッケージ要件 (リファレンス)	203
稼働中の OS に対する変更の防止	203
絶対パスの使用	203
pkgadd -R コマンドの使用	204
\$PKG_INSTALL_ROOT と \$BASEDIR の相違点の概要	204
スクリプト作成のガイドライン	205
ディスクレスクライアントの互換性維持	206
パッケージの検証	206
インストール中およびアップグレード中のユーザー操作の回避	207
ゾーンのパッケージパラメータの設定	208
背景情報	212
C アップグレード時のパッチアナライザの使用 (作業)	213
Solaris Update リリースへのアップグレード	213
▼ analyze_patches スクリプトを実行する方法	214
▼ パッチアナライザの出力を確認する方法	215
用語集	217
索引	233

はじめに

このマニュアルでは、SPARC® および x86 アーキテクチャーベースの、ネットワークに接続されたシステムとネットワークに接続されていないシステムの両方で、Solaris™ オペレーティングシステム (OS) をインストールおよびアップグレードする方法を説明します。

このマニュアルには、システムハードウェアや周辺装置を設定する方法は記載されていません。

注 - このリリースでは、SPARC および x86 系列のプロセッサアーキテクチャー (UltraSPARC®, SPARC64, AMD64, Pentium, Xeon EM64T) を使用するシステムをサポートします。サポートされるシステムについては、Solaris 10 Hardware Compatibility List (<http://www.sun.com/bigadmin/hcl>) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関連する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- 「x64」は、AMD64 または EM64T システムに関する 64 ビット特有の情報を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、Solaris 10 Hardware Compatibility List を参照してください。

対象読者

このマニュアルは、Solaris OS のインストールを担当するシステム管理者を対象としています。このマニュアルには、次の 2 種類の情報が含まれています。

- ネットワーク環境で複数の Solaris マシンを管理するエンタープライズシステム管理者向けの上級 Solaris インストール情報
- Solaris のインストールやアップグレードをときどき行うシステム管理者向けの基本 Solaris インストール情報

関連マニュアル

表 P-1 に、Solaris ソフトウェアをインストールする際に参考となる関連情報の一覧を示します。

表 P-1 関連情報

インフォメーション	説明
『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』	Solaris オペレーティングシステムのインストールやアップグレードの計画について解説されています。このマニュアルでは、すべての Solaris インストールプログラムの概要が説明されています。また、インストールに関連する次の機能の概要について解説されています。 <ul style="list-style-type: none"> ■ GRUB (オープンソースのブートローダー) ■ Solaris ゾーン区分技術 ■ インストール時に作成できる RAID-1 ボリューム
『Solaris 10 11/06 インストールガイド (基本編)』	グラフィカルユーザーインタフェース (GUI) を使用した、OS の基本インストールについて解説されています。
『Solaris 10 11/06 インストールガイド (ネットワークインストール)』	ローカルエリアネットワーク経由で Solaris のリモートインストールを実行する方法や、広域ネットワーク経由でセキュリティー保護されたインストールを実行する方法について説明します。
『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』	無人でのカスタム JumpStart™ インストールを実行するために、必要なファイルとディレクトリを作成する方法が解説されています。
『Solaris 10 11/06 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』	Solaris OS を複数のシステムにインストールするために使用する Solaris フラッシュアーカイブを作成する方法が解説されています。
『Solaris のシステム管理 (デバイスとファイルシステム)』の第 24 章「ファイルシステムのバックアップと復元 (概要)」	システムファイルのバックアップ方法など、システム管理タスクについて解説されています。
『Solaris ご使用にあたって』	Solaris リリースに関する、バグ、既知の問題、サポートが中止されたソフトウェア、パッチなどが解説されています。
SPARC: 『Solaris Sun ハードウェアマニュアル』	サポート対象の Sun ハードウェアについて解説されています。
『Solaris パッケージリスト』	Solaris OS に含まれるパッケージの一覧と説明です。
『Solaris (x86 Platform) Hardware Compatibility List』	サポート対象ハードウェアの情報とデバイス構成が解説されています。
	サポート対象ハードウェア情報とデバイス構成を含む Web ベースのデータベース。Sun Microsystems およびその他のベンダーの提供する SPARC および x86 システムが含まれます。

マニュアル、サポート、およびトレーニング

Sun の Web サイトでは、次のサービスに関する情報も提供しています。

- マニュアル (<http://jp.sun.com/documentation/>)
- サポート (<http://jp.sun.com/support/>)
- トレーニング (<http://jp.sun.com/training/>)

表記上の規則

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

表 P-2 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep '^#define \ XV_VERSION_STRING'

コード例は次のように表示されます。

- C シェル

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

- C シェルのスーパーユーザー

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

- Bourne シェルおよび Korn シェル

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

- Bourne シェルおよび Korn シェルのスーパーユーザー

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

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

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

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

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

パート I

Solaris Live Upgrade によるアップグレード

このパートでは、Solaris Live Upgrade を使って非アクティブブート環境を作成しアップグレードする方法について説明します。このブート環境は、アクティブブート環境に切り替えることができます。

◆ ◆ ◆ 第 1 章

Solaris インストールの計画についての参照先

このマニュアルでは、Solaris Live Upgrade プログラムを使用して Solaris オペレーティングシステムをアップグレードする方法について説明します。このマニュアルでは、Solaris Live Upgrade を使用する上で知っておく必要のある情報をすべて提供しますが、操作を開始する前にインストールマニュアルコレクション内の計画マニュアルを参照しておくことも役立ちます。次に示す参照先には、システムをアップグレードする前に役立つ情報が記載されています。

計画とシステム要件についての参照先

『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』には、システム要件と、ファイルシステムを計画するうえでのガイドラインやアップグレードの計画などの計画の概要が記載されています。次の一覧に、この計画マニュアルの章構成を示します。

計画マニュアルの章の説明	参照
この章では、Solaris インストールプログラムの新機能について説明します。	『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の第2章「Solaris インストールの新機能」
この章では、Solaris OS のインストールやアップグレードを実施する前に決定すべき内容について説明します。たとえば、ネットワークインストールイメージやDVDメディアをどのようなときに使用するかを判断するために必要な情報や、すべての Solaris インストールプログラムについての説明を記載しています。	『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の第3章「Solaris のインストールおよびアップグレード(ロードマップ)」
この章では、Solaris OS のインストールやアップグレードに伴うシステム要件について説明します。また、ディスク容量の計画に関しての一般的な指針や、スワップ空間のデフォルトの割り当てについても説明します。アップグレードの制限についても説明します。	『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の第4章「システム要件、ガイドライン、およびアップグレード(計画)」

計画マニュアルの章の説明	参照
<p>この章には、システムのインストールやアップグレードに必要な情報の収集に役立つチェックリストが含まれています。ここで提供される情報は、対話式インストールの実行時に役に立ちます。このチェックリストでは、対話式インストールを行うために必要なすべての情報が得られます。</p>	<p>『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の第5章「インストールやアップグレードの前に収集すべき情報 (計画)」</p>
<p>これらの章は、Solaris OS のインストールまたはアップグレードに関連するいくつかの技術の概要を説明します。これらの技術に関連するガイドラインと要件も含まれています。これらの章には、GRUB ベースのブート、Solaris ゾーン区分技術、およびインストール時に作成できる RAID-1 ボリュームについての情報が含まれています。</p>	<p>『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』のパート II 「GRUB、Solaris ゾーン、および RAID-1 ボリュームに関連するインストールについて」</p>

Solaris Live Upgrade (概要)

この章では、Solaris Live Upgrade の実行手順について説明します。

注-このマニュアルでは「スライス」という用語を使用しますが、一部の Solaris のマニュアルとプログラムでは、スライスのことを「パーティション」と呼んでいる場合があります。

Solaris Live Upgrade の紹介

Solaris Live Upgrade を使用すると、稼働中のシステムを停止することなくシステムをアップグレードできます。現在のブート環境を動作させたまま、ブート環境のコピーを作成し、それをアップグレードできます。アップグレードする代わりに、Solaris フラッシュアーカイブをブート環境にインストールすることもできます。環境をアップグレードしても、アーカイブをインストールしても、元のシステム構成は影響を受けずに支障なく機能します。準備ができたところでシステムをリブートすると、新しいブート環境がアクティブになります。障害が発生した場合は、リブートするだけで元のブート環境に戻ることができます。このように切り替えが可能なので、テストや評価処理のためにサービスを停止する必要がなくなります。

Solaris Live Upgrade を使用すると、現在動作しているシステムに影響を与えずに、ブート環境のコピーを作成して、次のような作業を行うことができます。

- システムをアップグレードします。
- 現在のブート環境のディスク構成を、新しいブート環境のディスク構成（ファイルシステムのタイプ、サイズ、および配置）に変更します。
- 異なるイメージを持つ複数のブート環境を保守します。たとえば、現在のパッチを持つブート環境を作成すると同時に、Update リリースを持つ別のブート環境を作成できます。

Solaris Live Upgrade を使用するには、システム管理についての基礎的な事柄を理解しておく必要があります。ファイルシステムの管理、マウント、ブート、スワップの管理な

ど、システム管理作業に関する基本的な情報については、『Solaris のシステム管理 (デバイスとファイルシステム)』を参照してください。

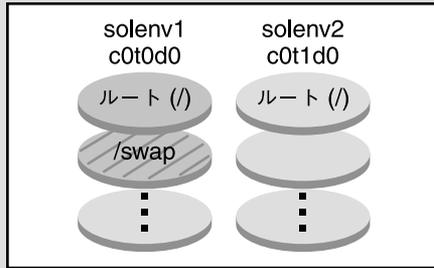
Solaris Live Upgrade の実行手順

次に、現在のブート環境のコピーを作成してこのコピーをアップグレードし、アクティブなブート環境になるように切り替える作業の概要を示します。元のブート環境に切り替えるフォールバックの手順についても説明します。図 2-1 に、この Solaris Live Upgrade 処理の全体を示します。

Solaris Live Upgrade の実行手順

- ① ブート環境を作成する。

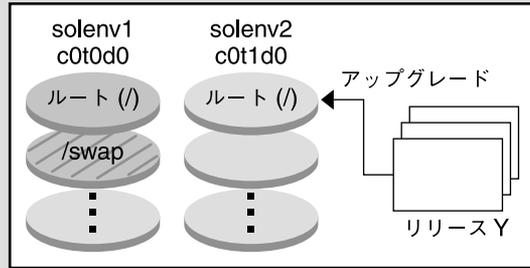
```
# lucreate -c solenv1 \
-m /:/dev/dsk/c0t1d0s0:ufs \
-n solenv2
```



- ② 非アクティブブート環境をアップグレードする。

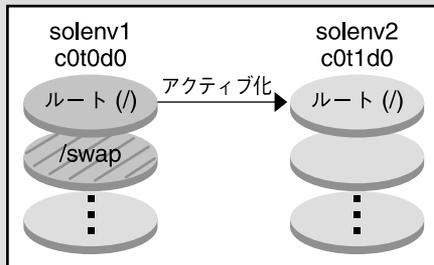
標準アップグレードの場合

```
(a) # luupgrade -u -n solenv2 \
-s /net/installmachine/export/Solaris/OS_image
```



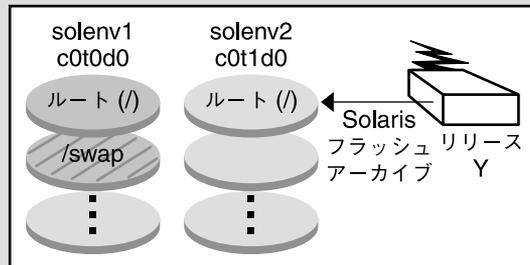
- ③ リポートして非アクティブブート環境をアクティブ化する。

```
# luactivate solenv2
# init 6
```



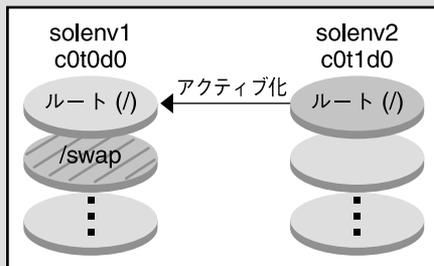
Solaris フラッシュアーカイブの場合

```
(b) # luupgrade -f -n solenv2 \
-s /net/installmachine/export/Solaris/Release_Y \
-a /net/server/archive/Release_Y
```



- ④ (任意) 元のブート環境へフォールバックする。

```
# luactivate solenv1
# init 6
```



- ⑤ (任意) 非アクティブ環境を削除する。

```
# ludelete solenv2
```

図 2-1 Solaris Live Upgrade の実行手順

次の節で、Solaris Live Upgrade の実行手順について説明します。

1. 物理スライスまたは論理ボリューム上での新しいブート環境の作成
 - 22 ページの「ブート環境の作成」
 - 27 ページの「RAID-1 ボリュームファイルシステムを持つブート環境の作成」
2. 33 ページの「ブート環境のアップグレード」
3. 36 ページの「ブート環境のアクティブ化」
4. 37 ページの「元のブート環境へのフォールバック」

ブート環境の作成

ブート環境を作成すると、クリティカルファイルシステムをアクティブなブート環境から新しいブート環境にコピーできます。必要であれば、ディスクを編成し直して、ファイルシステムをカスタマイズし、クリティカルファイルシステムを新しいブート環境にコピーします。

ファイルシステムのタイプ

Solaris Live Upgrade では、次の 2 種類のファイルシステムを区別します。クリティカルファイルシステムと共有可能ファイルシステムです。次の表に、これらのファイルシステムのタイプを示します。

ファイルシステムのタイプ	説明	例と詳細
クリティカルファイルシステム	クリティカルファイルシステムは、Solaris OS に必須のファイルシステムです。これらのファイルシステムは、アクティブなブート環境と非アクティブなブート環境の <code>vfstab</code> において別々のマウントポイントを持ちます。これらのファイルシステムは、必ずソースブート環境から非アクティブブート環境にコピーされます。クリティカルファイルシステムのことを「共有不能」と呼ぶこともあります。	<code>root (/)</code> 、 <code>/usr</code> 、 <code>/var</code> 、 <code>/opt</code> などがクリティカルファイルシステムの例です。
共有可能ファイルシステム	共有可能なファイルシステムとは、 <code>/export</code> のように、アクティブなブート環境と非アクティブなブート環境の両方の <code>vfstab</code> において同じマウントポイントを持つユーザー定義ファイルのことです。したがって、アクティブなブート環境内の共有ファイルを更新すると、非アクティブなブート環境のデータも更新されます。新しいブート環境を作成するとき、共有可能なファイルシステムはデフォルトで共有されます。しかし、コピー先のスライスを指定した場合、そのファイルシステムは(共有されずに)コピーされます。	たとえば、 <code>/export</code> が共有可能ファイルシステムの例です。 共有可能なファイルシステムについての詳細は、51 ページの「共有可能なファイルシステムのスライスを選択するための指針」を参照してください。

ファイルシステムのタイプ	説明	例と詳細
スワップ	スワップは、特殊な共有可能ファイルシステムです。ほかの共有可能なファイルシステムと同様に、すべてのスワップスライスはデフォルトで共有されます。しかし、スワップ用のディレクトリを指定した場合、スワップスライスは (共有されずに) コピーされます。	スワップを再構成する手順については、次の節を参照してください。 <ul style="list-style-type: none"> ■ 「ブート環境を作成する (キャラクターインターフェイス)」の手順 9 ■ 76 ページの「ブート環境を作成しスワップを再構成する (コマンド行インターフェイス)」

ファイルシステム上の RAID-1 ボリュームの作成

Solaris Live Upgrade では、ファイルシステム上に RAID-1 ボリューム (ミラー) を持つブート環境を作成できます。概要については、27 ページの「RAID-1 ボリュームファイルシステムを持つブート環境の作成」を参照してください。

ファイルシステムのコピー

新しいブート環境を作成するには、まず、クリティカルファイルシステムをコピーできる未使用のスライスが存在することを確認します。スライスが使用できないかあるいは最小限の要件を満たしていない場合は、新しいスライスをフォーマットする必要があります。

スライスを定義した後、ファイルシステムをディレクトリにコピーする前に、新しいブート環境上のファイルシステムを再構成できます。ファイルシステムを分割およびマージすることによって `vfstab` を簡単に編集でき、ファイルシステムを再構成することができます。ファイルシステムは、同じマウントポイントを指定して親ディレクトリにマージすることも、異なるマウントポイントを指定して親ディレクトリから分割することも可能です。

非アクティブブート環境でファイルシステムを構成した後、自動コピーを開始します。クリティカルファイルシステムは、指定された宛先ディレクトリにコピーされます。共有可能なファイルシステムは (それらの一部をコピーするように指定しない限り)、コピーされずに共有されます。ファイルシステムをアクティブなブート環境から非アクティブなブート環境にコピーする時、ファイルは新しいディレクトリにコピーされるので、アクティブなブート環境は変更されません。

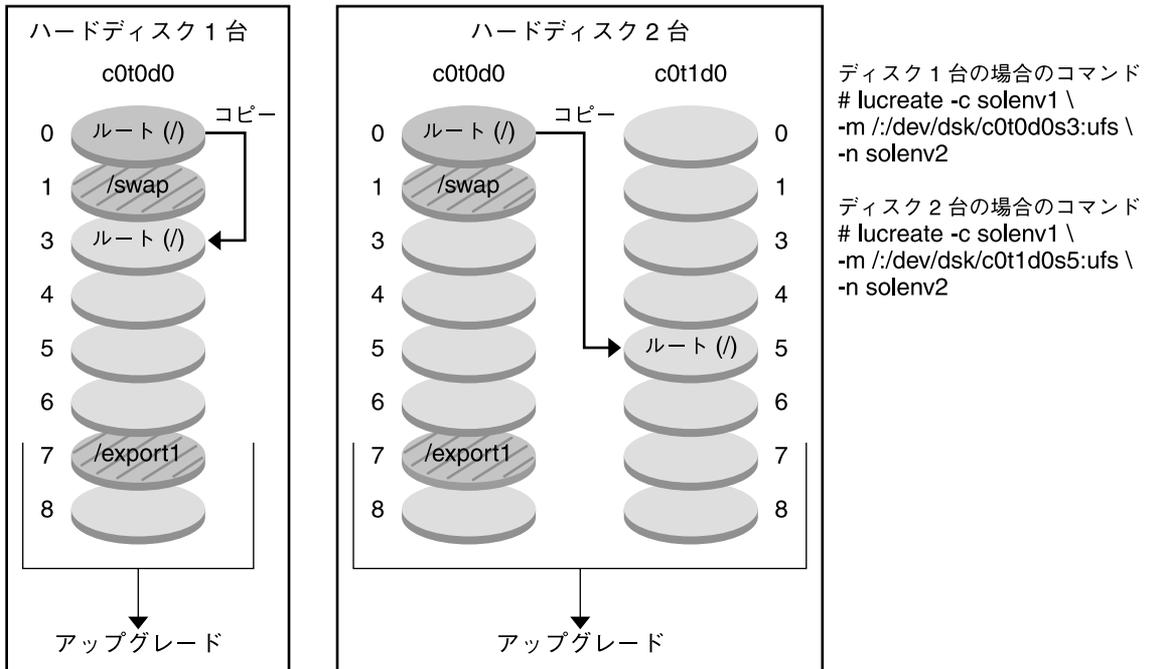
ファイルシステムの分割やマージの手順	<ul style="list-style-type: none">■ 「ブート環境を作成する(キャラクターインターフェイス)」の手順7または手順8■ 72ページの「ブート環境を作成しファイルシステムをマージする(コマンド行インターフェイス)」■ 74ページの「ブート環境を作成しファイルシステムを分割する(コマンド行インターフェイス)」
RAID-1 ボリュームファイルシステムを持つブート環境の作成の概要	27ページの「RAID-1 ボリュームファイルシステムを持つブート環境の作成」

新しいブート環境の作成の例

次の図に、さまざまな方法で新しいブート環境を作成する例を示します。

図 2-2 に、クリティカルファイルシステムのルート (/) をディスク上の別のスライスにコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境は、既存のスライス上にルート (/) ファイルシステムを持っています。新しいブート環境は、新しいスライス上にルート (/) ファイルシステムとまったく同じ複製を持ちます。ファイルシステム /swap および /export/home はアクティブなブート環境と非アクティブなブート環境で共有されます。

ブート環境の作成 -
スライスへのルート (/) ファイルシステムのコピー

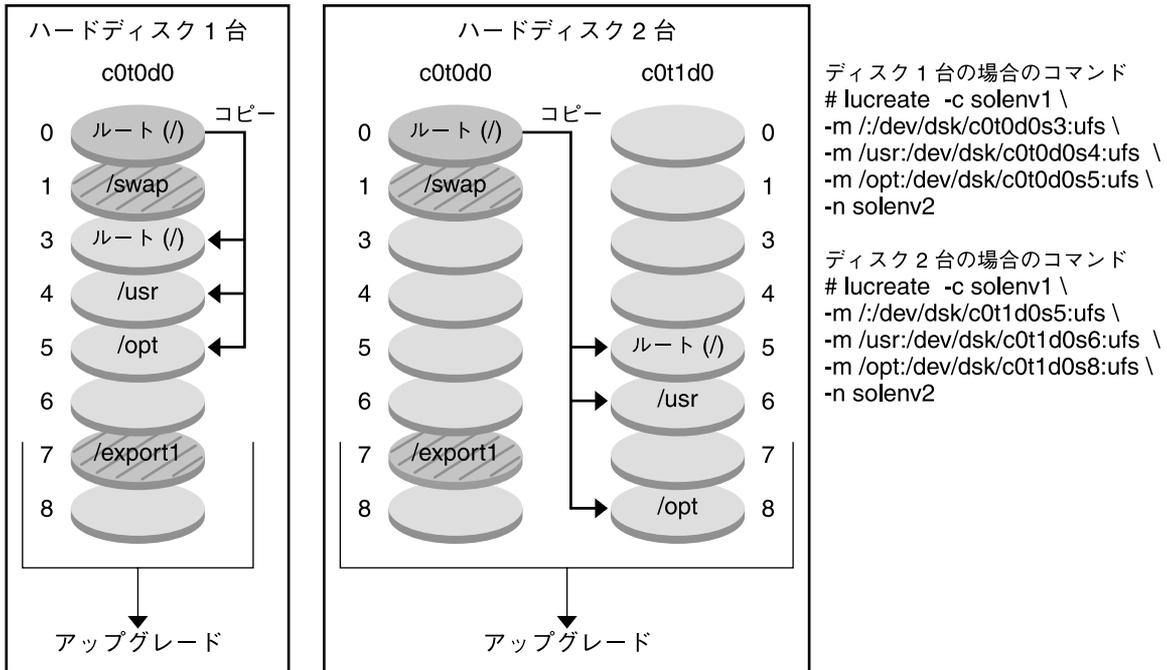


- 現在のリリース X
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース X
クリティカルなファイルシステムのルート (/)
- 共有ファイルシステム

図 2-2 非アクティブなブート環境の作成 - ルート (/) ファイルシステムのコピー

図 2-3 に、クリティカルファイルシステムを分割し、ディスク上の複数のスライスにコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境は、既存のスライス上にルート (/) ファイルシステムを持っています。このスライスでは、ルート (/) ファイルシステム内に、/usr、/var、および /opt ディレクトリがあります。新しいブート環境では、ルート (/) ファイルシステムは分割され、/usr と /opt は別のスライスに配置されています。ファイルシステム /swap と /export/home は両方のブート環境で共有されます。

ブート環境の作成 - ファイルシステムの分割

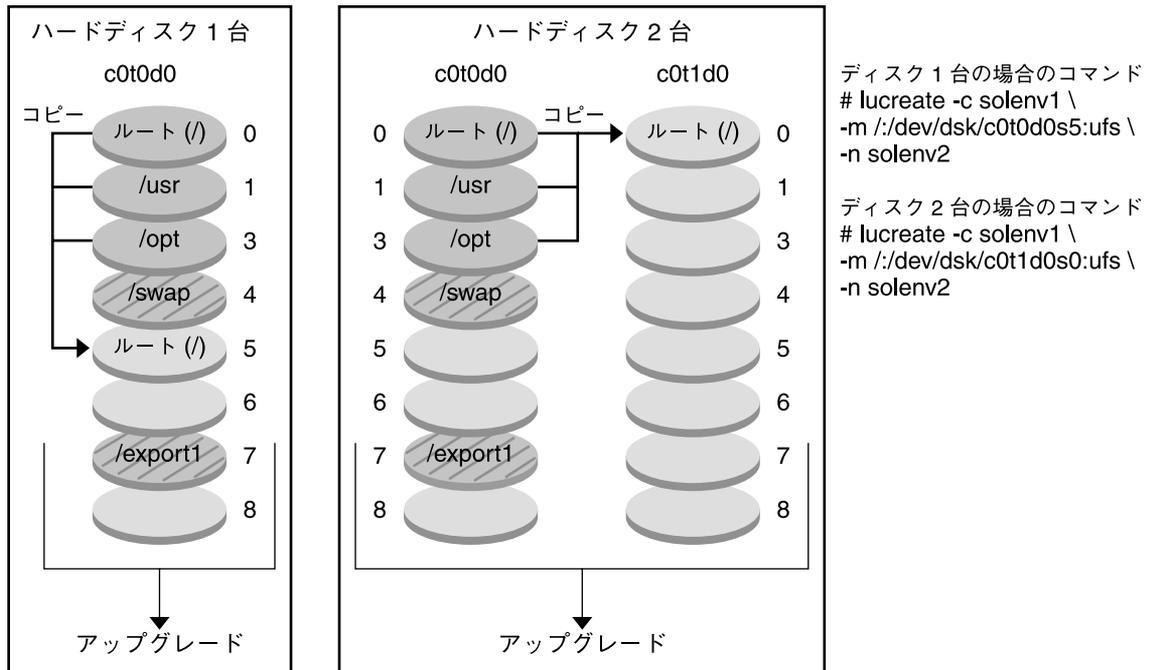


- 現在のリリース X
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース X
クリティカルなファイルシステムのルート (/) /usr /opt
- 共有ファイルシステム

図 2-3 非アクティブなブート環境の作成 - ファイルシステムの分割

図 2-4 に、クリティカルファイルシステムをマージし、ディスク上の複数のスライスにコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境には、ルート (/) ファイルシステム、/usr、/var、/opt があり、各ファイルシステムは別々のスライス上に配置されています。新しいブート環境では、/usr と /opt はルート (/) ファイルシステムと同一のスライス上にマージされます。ファイルシステム /swap と /export/home は両方のブート環境で共有されます。

ブート環境の作成 – ファイルシステムのマージ



- 現在のリリース X
クリティカルなファイルシステムのルート (/) /usr /opt
- 非アクティブなリリース Y
クリティカルなファイルシステムのルート (/)
- 共有ファイルシステム

図 2-4 非アクティブなブート環境の作成 - ファイルシステムのマージ

RAID-1 ボリュームファイルシステムを持つブート環境の作成

Solaris Live Upgrade は Solaris ボリュームマネージャーテクノロジーを使って、RAID-1 ボリュームにカプセル化されたファイルシステムを持つブート環境を作成できます。Solaris ボリュームマネージャーでは、ボリュームを使って確実にディスクやデータを管理できます。Solaris ボリュームマネージャーでは、連結、ストライプ、その他の複雑な構成が可能です。Solaris Live Upgrade では、これらの作業の一部を実行できます。たとえば、ルート (/) ファイルシステムの RAID-1 ボリュームを作成できます。

ボリュームを使用すると、複数のディスクにまたがるディスクスライスをグループ化して、OSで単一のディスクとして扱われるようにできます。Solaris Live Upgrade で作成できるのは、RAID-1 ボリューム(ミラー)内に単一スライスの連結を持つルート(/) ファイルシステムのブート環境だけです。これは、ブート用のスライスを1つだけ選択するようにブート PROM が制限されているためです。

Solaris Live Upgrade でボリュームを管理する方法

ブート環境を作成するとき、Solaris Live Upgrade を使って次の作業を行うことができます。

- 単一スライスの連結(サブミラー)を RAID-1 ボリューム(ミラー)から切り離します。必要な場合は、内容を保持して新しいブート環境の内容にすることができます。内容はコピーされないため、新しいブート環境を短時間で作成できます。ミラーから切り離されたサブミラーは、元のミラーの一部ではなくなります。サブミラーに対する読み取りや書き込みがミラーを介して実行されることはなくなります。
- ミラーを含んだブート環境を作成します。
- 新しく作成したミラーに単一スライスの連結を3つまで接続します。

lucreate コマンドの `-m` オプションを使って、新しいブート環境に対してミラーの作成、サブミラーの切り離し、およびサブミラーの接続を行うことができます。

注-現在のシステム上に VxVM ボリュームが構成されている場合は、lucreate コマンドを使用して新しいブート環境を作成できます。新しいブート環境にデータをコピーすると、Veritas ファイルシステム構成が失われ、新しいブート環境に UFS ファイルシステムが作成されます。

詳細な手順	85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」
インストール時の RAID-1 ボリューム作成の概要	『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の第8章「インストール時の RAID-1 ボリューム(ミラー)の作成(概要)」
Solaris Live Upgrade では使用できない Solaris ボリュームマネージャーの複雑な構成に関する詳細	『Solaris ボリュームマネージャの管理』の第2章「記憶装置管理の概念」

Solaris ボリュームマネージャーのタスクと Solaris Live Upgrade の対応

Solaris Live Upgrade では、Solaris ボリュームマネージャーのタスクの一部が管理されません。表 2-1 に、Solaris Live Upgrade で管理できる Solaris ボリュームマネージャーのコンポーネントを示します。

表 2-1 ボリュームクラス

用語	説明
連結	RAID-0 ボリューム。複数のスライスが連結された方式では、利用可能な最初のスライスがいっぱいになるまでそのスライスにデータが書き込まれます。そのスライスがいっぱいになると次のスライスに連続してデータが書き込まれます。ミラーに含まれている場合を除き、連結にはデータの冗長性はありません。
ミラー	RAID-1 ボリューム。「RAID-1 ボリューム」を参照してください。
RAID-1 ボリューム	同じデータのコピーを複数保持しているボリューム。RAID-1 ボリュームはミラーと呼ばれることもあります。RAID-1 ボリュームは、サブミラーと呼ばれる 1 つまたは複数の RAID-0 ボリュームから構成されます。
RAID-0 ボリューム	ストライプ方式または連結方式のボリューム。これらはサブミラーとも呼ばれます。ストライプや連結は、ミラーを構築する基本構成ブロックです。
状態データベース	状態データベースでは、Solaris ボリュームマネージャー構成の状態に関する情報がディスクに保存されます。状態データベースは、複製された複数のデータベースコピーの集まりです。各コピーは「状態データベースの複製」と呼ばれます。状態データベースは、既知の状態データベースの複製の格納場所と状態をすべて記録しています。
状態データベースの複製	状態データベースのコピー。複製により、データベース内のデータの有効性が保証されます。
サブミラー	「RAID-0 ボリューム」を参照してください。
ボリューム	システムで単一の論理デバイスとして扱われる、物理スライスやボリュームの集まり。アプリケーションやファイルシステムから見ると、ボリュームは物理ディスクと同じように機能します。一部のコマンド行ユーティリティでは、ボリュームはメタデバイスと呼ばれます。

Solaris Live Upgrade を使用して RAID-1 ボリュームを作成する例

次の例では、新しいブート環境の RAID-1 ボリュームを作成するためのコマンド構文を示します。

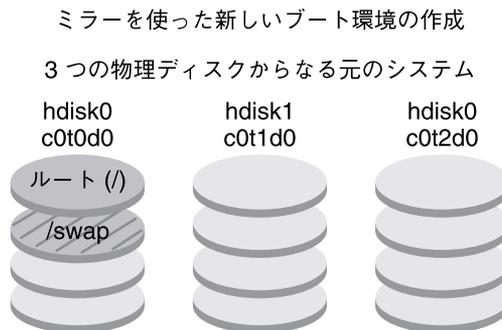
2つの物理ディスク上に RAID-1 ボリュームを作成する

図 2-5 は、2つの物理ディスク上に作成された RAID-1 ボリューム(ミラー)を持つ新しいブート環境を示しています。この新しいブート環境とミラーは、次のコマンドで作成されたものです。

```
# lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d31:attach -m /:/dev/dsk/c0t2d0s0,/dev/md/dsk/d32:attach \
-m -:/dev/dsk/c0t1d0s1:swap -m -:/dev/dsk/c0t2d0s1:swap
```

このコマンドは、次のような処理を実行します。

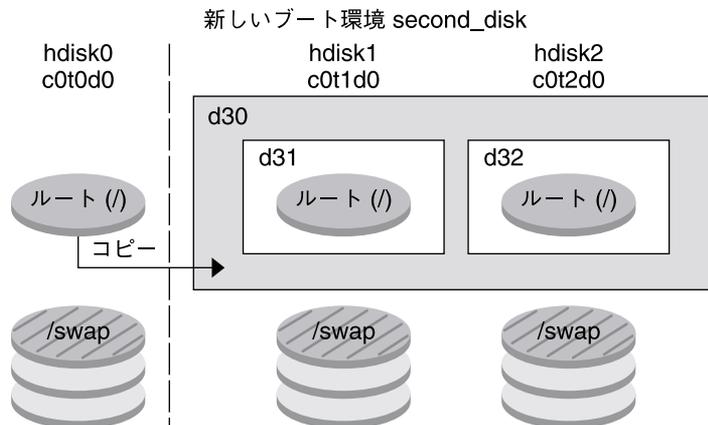
- 新しいブート環境 `second_disk` を作成する。
- ミラー `d30` を作成し、UFS ファイルシステムを構成する。
- 各物理ディスクのスライス 0 に単一デバイスの連結を作成する。これらの連結に `d31` および `d32` という名前を付ける。
- これら 2 つの連結をミラー `d30` に追加する。
- ルート (`/`) ファイルシステムをミラーにコピーする。
- 各物理ディスクのスライス 1 に、スワップ用のファイルシステムを構成する。



```

コマンド: lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d31:attach \
-m /:/dev/dsk/c0t2d0s0,/dev/md/dsk/d32:attach \
-m -:/dev/dsk/c0t1d0s1:swap -m -:/dev/dsk/c0t2d0s1:swap

```



- d30 – RAID-1 ボリューム (ミラー)
- d31 – 単一スライスの連結 (サブミラー)
- d32 – 単一スライスの連結 (サブミラー)

図2-5 ブート環境の作成とミラーの作成

ブート環境の作成と既存のサブミラーの使用

図 2-6 は、RAID-1 ボリューム(ミラー)を持つ新しいブート環境を示しています。この新しいブート環境とミラーは、次のコマンドで作成されたものです。

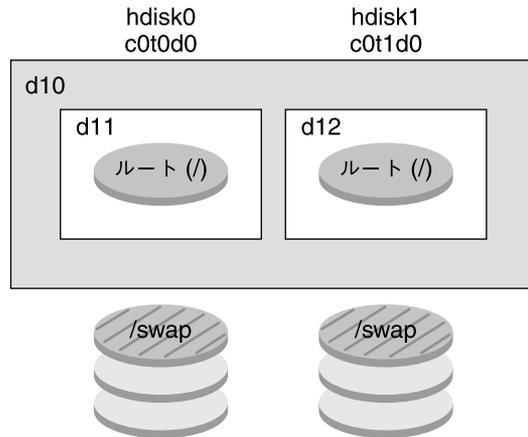
```
# lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

このコマンドは、次のような処理を実行します。

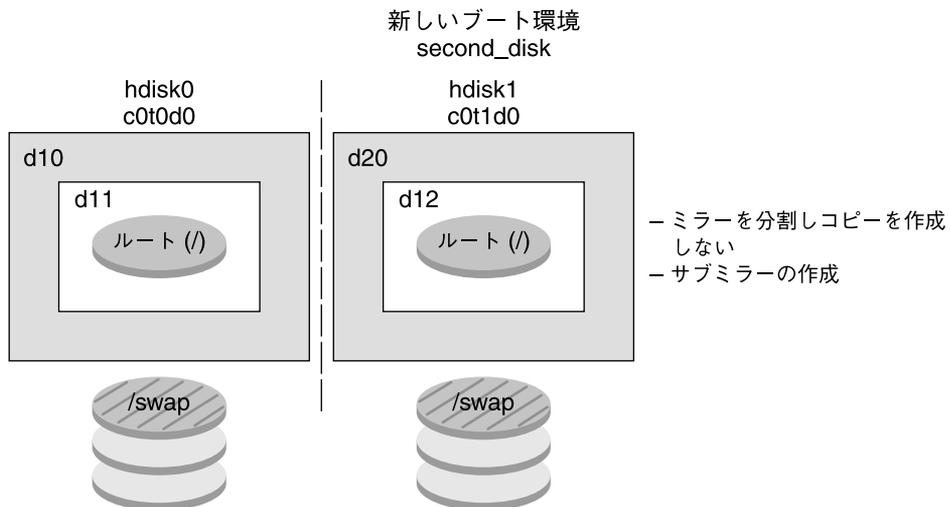
- 新しいブート環境 `second_disk` を作成する。
- ミラー `d10` を解除し、連結 `d12` を切り離す。
- 連結 `d12` の内容を保持する。ファイルシステムのコピーは行われない。
- 新しいミラー `d20` を作成する。これで、`d10` および `d20` という 2 つの 1 面ミラーが作成される。
- 連結 `d12` をミラー `d20` に接続する。

新しいブート環境の作成と既存のサブミラーの流用

2つの物理ディスクからなる元のシステム



コマンド: `lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \`
`-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve`



d10 - RAID-1 ボリューム (ミラー)
d11 - 単一スライスの連結 (サブミラー)
d12 - 単一スライスの連結 (サブミラー)
d20 - 新しい RAID-1 ボリューム (ミラー)

図2-6 ブート環境の作成と既存のサブミラーの使用

ブート環境のアップグレード

ブート環境の作成が完了したら、そのブート環境をアップグレードできます。アップグレード作業の過程で、ブート環境の任意のファイルシステムに RAID-1 ボリューム (ミラー) を持たせることができます。アップグレードを行なっても、アクティブなブート環境内のファイルには影響ありません。準備ができたところでこの新しいブート環境をアクティブ化し、このブート環境を現行のブート環境とします。

ブート環境のアップグレード手順

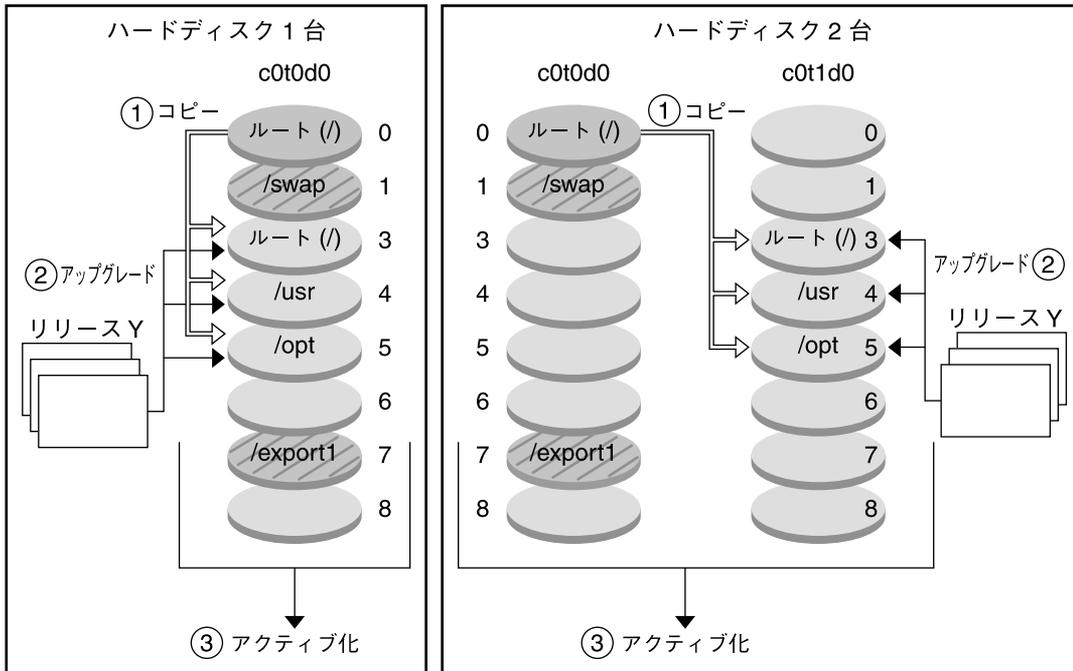
第 5 章

RAID-1 ボリュームファイルシステムを持つブート環境のアップグレードの例

168 ページの「RAID-1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース)」

図 2-7 に、非アクティブなブート環境のアップグレードの例を示します。

ブート環境のアップグレード



■ 現在のリリース X
クリティカルなファイルシステムのルート (/)

□ 非アクティブなリリース Y
クリティカルなファイルシステムのルート (/)
/usr /opt

▨ 共有ファイルシステム

① ディスク 1 台の場合のコマンド
lucreate -c solenv1 \
-m /:/dev/dsk/c0t0d0s3:ufs -m /usr:/dev/dsk/c0t0d0s4:ufs \
-m /opt:/dev/dsk/c0t0d0s5:ufs \
-n solenv2

① ディスク 2 台の場合のコマンド
lucreate -c solenv1 \
-m /:/dev/dsk/c0t1d0s3:ufs -m /usr:/dev/dsk/c0t1d0s4:ufs \
-m /opt:/dev/dsk/c0t1d0s5:ufs
-n solenv2

② # luupgrade -u -n solenv2 \
-s /net/installmachine/export/Solaris_10/OS_image

図 2-7 非アクティブなブート環境のアップグレード

アップグレードする代わりに、Solaris フラッシュアーカイブをブート環境にインストールすることもできます。Solaris フラッシュインストール機能を使用すると、Solaris OS の単一の参照用インストールを 1 台のシステム上に作成できます。このシステムはマスターシステムと呼ばれます。続いて、クローンシステムと呼ばれる多数のシステム上にこのインストールを複製できます。この場合、非アクティブなブート環境はクローンシ

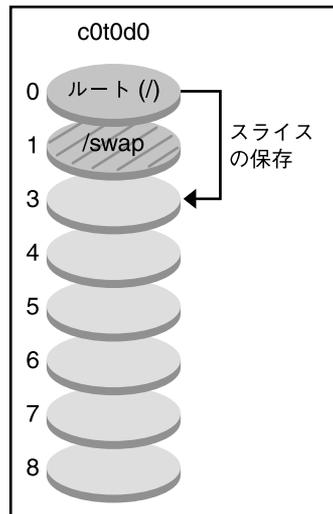
システムです。Solaris フラッシュアーカイブをシステムにインストールするとき、初期インストールの場合と同じように、アーカイブは既存のブート環境にあるすべてのファイルを置き換えます。

Solaris フラッシュアーカイブのインストール手順については、112 ページの「ブート環境への Solaris フラッシュアーカイブのインストール」を参照してください。

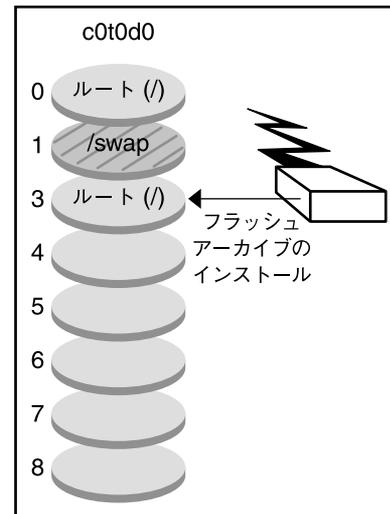
次の図に、非アクティブなブート環境における Solaris フラッシュアーカイブのインストールを示します。図 2-8 は、1 台のハードディスクを持つシステムを示しています。図 2-9 は、2 台のハードディスクを持つシステムを示しています。

Solaris フラッシュアーカイブのインストール - ハードディスク 1 台

① 空のブート環境の作成



② フラッシュアーカイブをインストールすることによるアップグレード

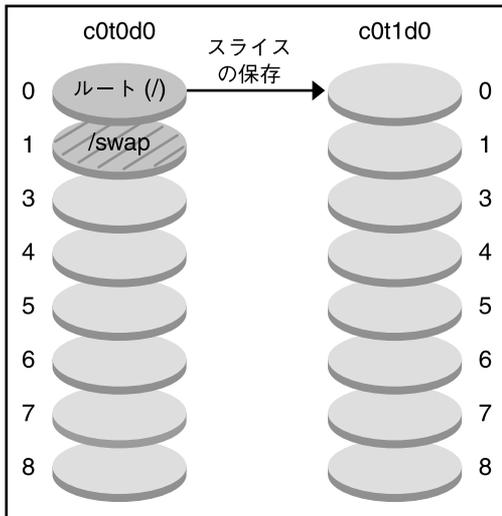
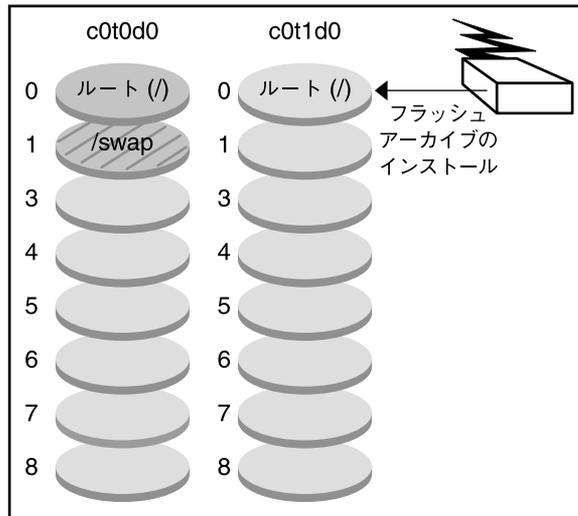


- | | |
|--|---|
| <input type="checkbox"/> 現在のリリース X
クリティカルなファイルシステムのルート (/) | コマンド
lucreate -s -\
-m /:/dev/dsk/c0t0d0s3:ufs -n solenv2 |
| <input type="checkbox"/> 非アクティブなリリース Y
クリティカルなファイルシステムのルート (/)
/usr /opt | # luupgrade -f -n solenv2 \
-s /net/installmachine/export \
/Solaris/OS_image \
-a /net/server/archive/Solaris |
| <input checked="" type="checkbox"/> 共有ファイルシステム | |

図 2-8 Solaris フラッシュアーカイブのインストール - ハードディスク 1 台

Solaris フラッシュアーカイブのインストール - ハードディスク 2 台

空のブート環境の作成

フラッシュアーカイブを
インストールすることによるアップグレード

- 現在のリリース X
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース X
クリティカルなファイルシステムのルート (/)
- 共有ファイルシステム

```

コマンド
# lucreate -s - \
-m /:/dev/dsk/c0t1d0s0 -n solenv2

# luupgrade -f -n solenv2 \
-s /net/installmachine/export \
/Solaris/OS_image \
-a /net/server/archive/Solaris

```

図 2-9 Solaris フラッシュアーカイブのインストール - ハードディスク 2 台

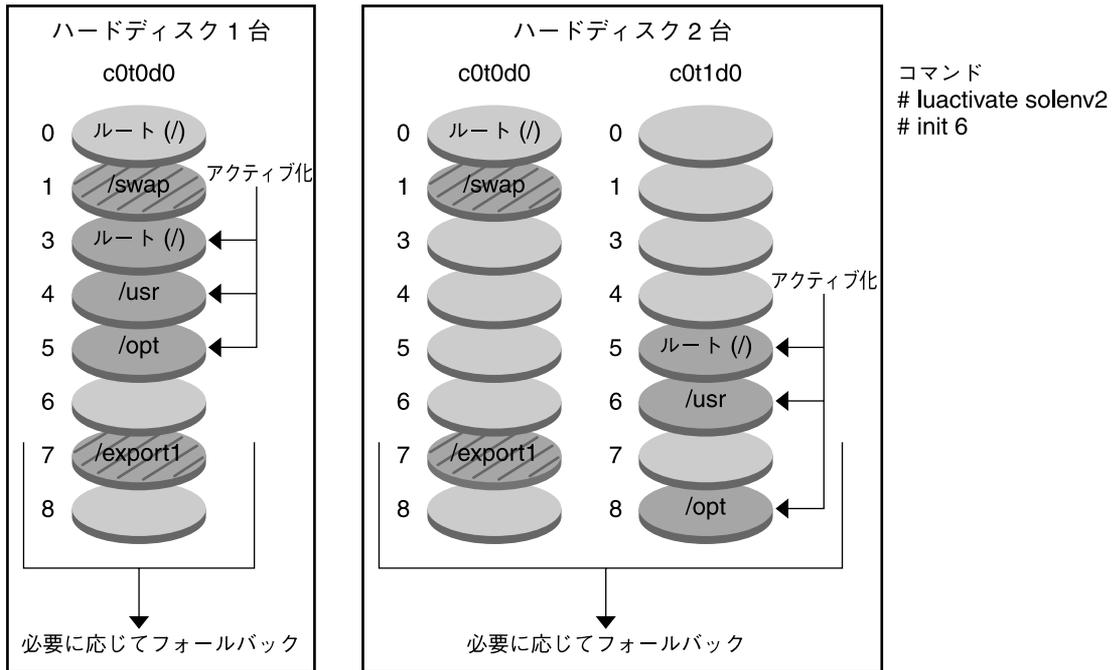
ブート環境のアクティブ化

新しいブート環境に切り替えてアクティブにする準備ができたなら、新しいブート環境をすばやくアクティブにしてリブートします。新たに作成したブート環境を初めて起動するとき、ブート環境間でファイルの同期がとられます。ここでいう「同期」とは、いくつかのシステムファイルやディレクトリを、直前にアクティブだったブート環境から、ブート中のブート環境へコピーすることです。システムをリブートすると、非アクティブなブート環境にインストールした構成がアクティブになります。この時点で、元のブート環境は非アクティブブート環境となります。

アクティブなブート環境と非アクティブなブート環境の同期についての情報 53 ページの「ブート環境間でのファイルの同期」

図 2-10 に、リブート後に非アクティブなブート環境からアクティブなブート環境に切り替わる様子を示します。

ブート環境のアクティブ化



- 現在のリリース Y
クリティカルなファイルシステムのルート (/) /usr /opt
 - 非アクティブなリリース X
クリティカルなファイルシステムのルート (/)
 - 共有ファイルシステム
- 図 2-10 非アクティブなブート環境のアクティブ化

元のブート環境へのフォールバック

問題が発生する場合は、アクティブ化とリブートを行なって元のブート環境にすぐにフォールバックできます。元のブート環境をバックアップして復元するよりも、フォールバックの方がはるかに時間がかかりません。ブートに失敗した新しいブート環境は保

存されるので、障害を解析できます。フォールバックを実行できるのは、`luactivate` を使用して新しいブート環境をアクティブにしたブート環境だけです。

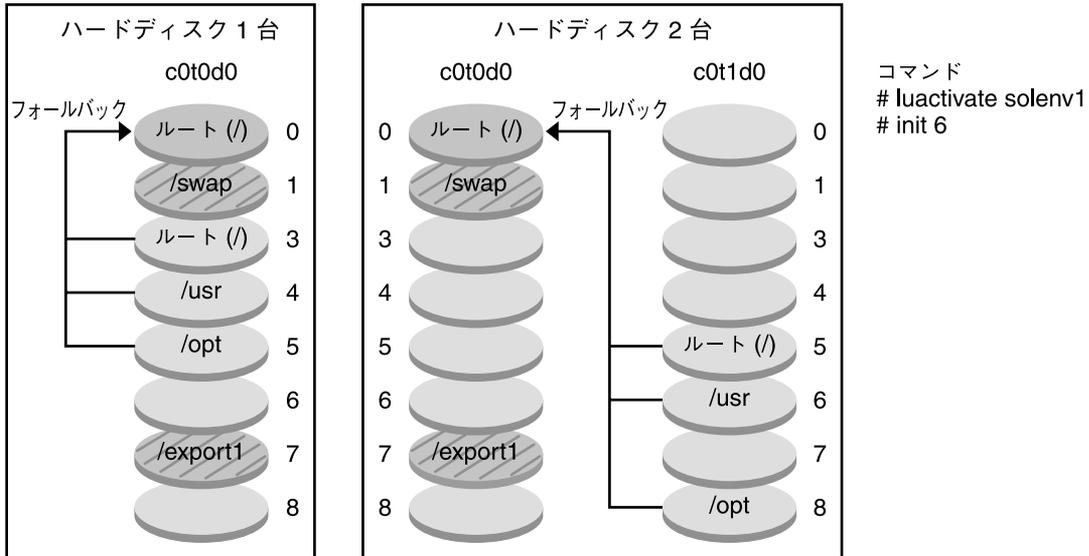
以前のブート環境にフォールバックするには、次の手順に従います。

問題	操作
新しいブート環境は正常にブートしたが、結果に満足できない。	<p><code>luactivate</code> コマンドに以前のブート環境の名前を指定して実行し、リポートします。</p> <p>x86 のみ - Solaris 10 1/06 以降のリリースでは、GRUB メニューにある元のブート環境を選択してフォールバックすることができます。元のブート環境と新しいブート環境は、GRUB ソフトウェアに基づいている必要があります。GRUB メニューからブートすると、古いブート環境と新しいブート環境の間でファイルは同期されません。ファイルの同期の詳細については、54 ページの「ブート環境間での強制的な同期」を参照してください。</p>
新しいブート環境がブートしない。	<p>フォールバックするブート環境をシングルユーザーモードでブートし、<code>luactivate</code> コマンドを実行し、リポートします。</p>
シングルユーザーモードでブートできない。	<p>次のいずれかの操作を実行します。</p> <ul style="list-style-type: none"> ■ DVD/CD メディアまたはネットインストールイメージからブートします ■ フォールバックするブート環境上のルート (/) ファイルシステムをマウントします ■ <code>luactivate</code> コマンドを実行し、リポートします

フォールバックの手順については、[第 6 章](#)を参照してください。

[図 2-11](#) に、リポートしてフォールバックしたときにブート環境が切り替わる様子を示します。

元のブート環境へのフォールバック



- 現在のリリース X
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース X
クリティカルなファイルシステムのルート (/)
- ▨ 共有ファイルシステム

図 2-11 元のブート環境へのフォールバック

ブート環境の保守

ブート環境のステータス確認、名前変更、削除など、さまざまな保守作業も行うことができます。保守作業の手順については、[第 7 章](#)を参照してください。

Solaris Live Upgrade (計画)

この章では、Solaris Live Upgrade のインストールと使用を開始する前に考慮すべき指針と要件を説明します。『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の「アップグレード計画」で、アップグレード全般に関する情報も確認してください。この章の内容は次のとおりです。

- 41 ページの「Solaris Live Upgrade の要件」
- 46 ページの「パッケージまたはパッチによるシステムのアップグレード」
- 47 ページの「lucreate コマンドを使用したファイルシステムの作成のための指針」
- 48 ページの「ファイルシステムのスライスを選択するための指針」
- 52 ページの「新しいブート環境の内容のカスタマイズ」
- 53 ページの「ブート環境間でのファイルの同期」
- 56 ページの「リモートシステムからの Solaris Live Upgrade の使用」

Solaris Live Upgrade の要件

Solaris Live Upgrade のインストールと使用を開始する前に、次の要件をよく理解してください。

Solaris Live Upgrade のシステム要件

Solaris Live Upgrade は Solaris ソフトウェアに含まれています。現在の OS に Solaris Live Upgrade パッケージをインストールする必要があります。アップグレード後の OS のリリース番号と同じリリース番号の Solaris Live Upgrade パッケージをインストールする必要があります。たとえば、OS を現在使用している Solaris 9 リリースから Solaris 10 11/06 リリースにアップグレードする場合、Solaris 10 11/06 リリースの Solaris Live Upgrade パッケージをインストールする必要があります。

表 3-1 に、Solaris Live Upgrade でサポートされるリリースを示します。

表 3-1 サポートされる Solaris リリース

現在のリリース	互換性のあるアップグレードリリース
Solaris 8 OS	Solaris 8、9、またはすべての Solaris 10 リリース
Solaris 9 OS	Solaris 9 またはすべての Solaris 10 リリース
Solaris 10 OS	すべての Solaris 10 リリース

Solaris Live Upgrade のインストール

Solaris Live Upgrade パッケージのインストールには、次を使用します。

- pkgadd コマンド。Solaris Live Upgrade パッケージは SUNWlur と SUNWluu です。この順序でインストールする必要があります。
- Solaris Operating System DVD、Solaris SOFTWARE - 2 CD、またはネットインストールイメージ上にあるインストーラ。

Solaris Live Upgrade を正しく操作するためには、次のパッチのインストールが必要な場合があります。

説明	参照先
<p>注意: Solaris Live Upgrade を正しく操作するためには、指定の OS バージョン用の特定のパッチリビジョンのセットがインストールされている必要があります。Solaris Live Upgrade をインストールまたは実行する前に、これらのパッチをインストールする必要があります。</p>	<p>http://sunsolve.sun.com で最新のパッチリストを確認してください。SunSolve の Web サイトで、infodoc 72099 を検索してください。</p>
<p>x86のみ - このパッチのセットがインストールされていない場合、Solaris Live Upgrade は失敗し、次のエラーメッセージが表示されることがあります。次のエラーメッセージが表示されなくても、必要なパッチがインストールされていない場合があります。Solaris Live Upgrade のインストールを試みる前に、SunSolve の infodoc に記載されたすべてのパッチがすでにインストール済みであることを必ず確認してください。</p>	
<pre>ERROR: Cannot find or is not executable: </sbin/biosdev>. ERROR: One or more patches required by Live Upgrade has not been installed.</pre>	
<p>infodoc 72099 に記載されたパッチは、随時変更される可能性があります。これらのパッチにより、Solaris Live Upgrade の欠陥が修正される可能性があると同時に、Solaris Live Upgrade が依存するコンポーネントの欠陥も修正される可能性があります。Solaris Live Upgrade で問題が発生した場合は、最新の Solaris Live Upgrade パッチがインストールされていることを確認してください。</p>	
<p>Solaris 8 または 9 OS を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。</p>	<p>Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、Java 2 Runtime Environment 推奨パッチクラスタをインストールします。このパッチクラスタは http://sunsolve.sun.com から入手できます。</p>

Solaris Live Upgrade ソフトウェアのインストール方法については、59 ページの「Solaris Live Upgrade のインストール」を参照してください。

必要なパッケージ

Solaris Live Upgrade で問題がある場合は、パッケージが不足している可能性があります。次の表に示されたパッケージが、使用している OS にインストールされていることを確認してください。これらは、Solaris Live Upgrade を使用する上で必要なパッケージです。

Solaris 10 リリースの場合:

- 次のソフトウェアグループのいずれかをインストールする場合、これらのソフトウェアグループには必要なすべての Solaris Live Upgrade パッケージが含まれていません。
 - 全体ディストリビューションと OEM サポート
 - 全体ディストリビューション
 - 開発者システムサポート
 - エンドユーザーシステムサポート
- 次のソフトウェアグループのいずれかをインストールする場合は、Solaris Live Upgrade を使用する上で必要なパッケージの一部が含まれていない可能性があります。
 - コアシステムサポート
 - 限定ネットワークシステムサポート

ソフトウェアグループについては、『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の「ソフトウェアグループごとの推奨ディスク容量」を参照してください。

表 3-2 Solaris Live Upgrade に必要なパッケージ

Solaris 8 リリース	Solaris 9 リリース	Solaris 10 リリース
SUNWadmap	SUNWadmap	SUNWadmap
SUNWadmC	SUNWadmC	SUNWadmlib-sysid
SUNWlibC	SUNWadmfw	SUNWadmr
SUNWbzip	SUNWlibC	SUNWlibC
SUNWgzip	SUNWgzip	Solaris 10 3/05 のみ: SUNWgzip
SUNWj2rt	SUNWj2rt	SUNWj5rt
注 - SUNWj2rt パッケージは、次の状況でのみ必要になります。	注 - SUNWj2rt パッケージは、次の状況でのみ必要になります。	注 - SUNWj5rt パッケージは、次の状況でのみ必要になります。
<ul style="list-style-type: none"> ■ Solaris Live Upgrade インストーラを実行して Solaris Live Upgrade パッケージを追加する場合 ■ アップグレードの際に CD メディアを使用する場合 	<ul style="list-style-type: none"> ■ Solaris Live Upgrade インストーラを実行して Solaris Live Upgrade パッケージを追加する場合 ■ アップグレードの際に CD メディアを使用する場合 	<ul style="list-style-type: none"> ■ Solaris Live Upgrade インストーラを実行して Solaris Live Upgrade パッケージを追加する場合 ■ アップグレードの際に CD メディアを使用する場合

システム上のパッケージを確認するには、次のコマンドを入力します。

```
% pkginfo package_name
```

Solaris Live Upgrade のディスク容量の要件

アップグレードの一般的なディスク容量の要件に従います。『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の第4章「システム要件、ガイドライン、およびアップグレード (計画)」を参照してください。

ブート環境の作成に必要なファイルシステムのサイズを見積もるには、新しいブート環境の作成を開始してください。サイズが計算されたところで、処理を中断できます。

新しいブート環境上のディスクをブートデバイスとして使用する必要があります。システムの中には、ブートデバイスとして機能するディスクを限定するものがあります。ブート制限が適用されるかどうかを確認するには、各システムのマニュアルを参照してください。

新しいブート環境を作成する前に、ディスクの準備が必要になることもあります。ディスクが正しくフォーマットされていることを次のように確認します。

- スライスがファイルシステムをコピーできるだけの十分な大きさであることを確認します。
- ブート環境間でコピーするのではなく、共有するディレクトリが入っているファイルシステムを確認します。ディレクトリを共有する場合、そのディレクトリを固有のスライスに配置して新しいブート環境を作成する必要があります。こうすることにより、ディレクトリは、将来のブート環境と共有可能なファイルシステムになります。独立したファイルシステムを作成して共有する方法についての詳細は、[51 ページ](#)の「共有可能なファイルシステムのスライスを選択するための指針」を参照してください。

RAID-1 ボリューム (ミラー) を作成する場合の Solaris Live Upgrade の要件

Solaris Live Upgrade は Solaris ボリュームマネージャーのテクノロジーを使用して、RAID-1 ボリューム (ミラー) を備えたファイルシステムを持つブート環境のコピーを作成します。Solaris Live Upgrade では、Solaris ボリュームマネージャーのすべての機能が実装されるわけではありませんが、Solaris ボリュームマネージャーの次のコンポーネントが必要になります。

表 3-3 Solaris Live Upgrade と RAID-1 ボリュームに必要なコンポーネント

要件	説明	参照先
状態データベースを1つ以上、状態データベースの複製を3つ以上作成する必要があります。	状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。状態データベースは、複製された複数のデータベースコピーの集まりです。各コピーは「状態データベースの複製」と呼ばれます。状態データベースのコピーを作成することで、単一点障害によるデータ損失を防ぐことができます。	状態データベースの作成については、『Solaris ボリュームマネージャの管理』の第6章「状態データベース(概要)」を参照してください。
Solaris Live Upgrade では、ルート (/) ファイルシステムに単一スライスの連結を持つ RAID-1 ボリューム(ミラー)だけがサポートされません。	連結は RAID-0 ボリュームです。複数のスライスが連結された方式では、利用可能な最初のスライスがいっぱいになるまでそのスライスにデータが書き込まれます。そのスライスがいっぱいになると次のスライスに連続してデータが書き込まれます。RAID-1 ボリュームに含まれている場合を除き、連結にはデータの冗長性はありません。 RAID-1 ボリュームは、最大3つの連結から構成されます。	ミラー化されたファイルシステムの作成のガイドラインについては、49 ページの「ミラー化されたファイルシステムのスライスを選択するための指針」を参照してください。

パッケージまたはパッチによるシステムのアップグレード

Solaris Live Upgrade を使ってパッケージやパッチをシステムに追加できます。Solaris Live Upgrade を使用すると、システムのダウンタイムはリブートの時間だけですみます。luupgrade コマンドを使って、パッチやパッケージを新しいブート環境に追加できます。luupgrade コマンドを使用する場合は、Solaris フラッシュアーカイブを使用してパッチやパッケージをインストールすることもできます。



注意 - Solaris Live Upgrade でアップグレードしたりパッケージやパッチの追加・削除を行ったりするには、パッケージやパッチが SVR4 パッケージガイドラインに準拠していません。Sun のパッケージはこのガイドラインに準拠していますが、サードパーティーベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、アップグレード時にパッケージ追加ソフトウェアの障害が発生するか、アクティブブート環境が改変されてしまう可能性があります。

パッケージの要件については、[付録 B](#) を参照してください。

インストールの種類	説明	参照先
ブート環境へのパッチの追加	新しいブート環境を作成してから、 <code>-t</code> オプションを指定して <code>luupgrade</code> コマンドを実行します。	103 ページの「ブート環境のオペレーティングシステムイメージにパッチを追加する (コマンド行インタフェース)」
ブート環境へのパッケージの追加	<code>-p</code> オプションを指定して <code>luupgrade</code> コマンドを実行します。	102 ページの「ブート環境のオペレーティングシステムイメージにパッケージを追加する (コマンド行インタフェース)」
Solaris Live Upgrade を使った Solaris フラッシュアーカイブのインストール	アーカイブには、新しいパッケージやパッチがすでに追加されているブート環境の完全なコピーが格納されています。このコピーを複数のシステムにインストールできます。	<ul style="list-style-type: none"> ■ Solaris フラッシュアーカイブの詳しい作成手順については、『Solaris 10 11/06 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第3章「Solaris フラッシュアーカイブの作成 (作業)」を参照してください。 ■ Solaris Live Upgrade を使って Solaris フラッシュアーカイブをインストールする方法については、112 ページの「ブート環境への Solaris フラッシュアーカイブのインストール」を参照してください。

lucreate コマンドを使用したファイルシステムの作成のための指針

lucreate コマンドに `-m` オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。`-m` オプションを使ってファイルシステムを作成する場合、次の項目に留意してください。

- `-m` オプションを1個指定して、新しいブート環境のルート (`/`) ファイルシステムを作成する必要があります。`-m` オプションを指定しないで `lucreate` を実行すると、「Configuration」メニューが表示されます。「Configuration」メニューでは、新しいマウントポイントにファイルを変更して新しいブート環境をカスタマイズできます。
- 現在のブート環境にクリティカルファイルシステムがある場合、このファイルシステムは、`-m` オプションで指定しなくても新しく作成されたファイルシステムの上位2番目のファイルシステムにマージされます。
- 新しいブート環境には、`-m` オプションで指定されたファイルシステムだけが作成されます。現在のブート環境に複数のファイルシステムがあり、新しいブート環境にも同じ数のファイルシステムを作成する場合は、ファイルシステムごとに1個ずつ `-m` オプションを指定します。

たとえば、`-m` オプションを1回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のすべてのファイルシステムが、`-m` で指定されたファイルシステムにマージされます。`-m` オプションを2回使用すると、ファイル

システムが2つ作成されます。ルート (/) ファイルシステム、 /opt ファイルシステム、 /var ファイルシステムがある場合、これらを新しいブート環境に作成するには、それぞれに -m オプションを1個ずつ指定します。

- マウントポイントが重複しないようにしてください。たとえば、ルート (/) ファイルシステムを2つ作成することはできません。

ファイルシステムのスライスを選択するための指針

ブート環境のファイルシステムを作成する場合の規則は、Solaris OSのファイルシステムを作成する場合と同じです。Solaris Live Upgradeでは、クリティカルファイルシステムに無効な構成を作成できてしまいます。たとえば、lucreate コマンドを入力して、ルート (/) と /kernel に別々のファイルシステムを作成することができますが、このようにルート (/) ファイルシステムを分割するのは誤りです。

ディスクスライスを作成するときは、スライスがオーバーラップしないように注意してください。スライスのオーバーラップがあると、新しいブート環境を作成したつもりでも、アクティブにした後ブートすることができません。こうしたオーバーラップは、ファイルシステムの破損の原因となります。

Solaris Live Upgrade を正しく機能させるには、アクティブブート環境の `vfstab` ファイルの内容が有効で、ルート (/) ファイルシステムのエントリが少なくとも1つは含まれている必要があります。

ルート (/) ファイルシステムのスライスを選択するための指針

非アクティブブート環境を作成する場合は、ルート (/) ファイルシステムがコピーされるスライスを確認する必要があります。ルート (/) ファイルシステムのスライスを選択する場合は、次の項目に留意してください。スライスは、次の条件を満たしていなければなりません。

- システムをブートできるスライスである。
- 推奨されている最小サイズ以上である。
- アクティブなルート (/) ファイルシステムとは異なる物理ディスクでも同じディスクでもかまわない。
- VxVM (Veritas Volume Manager) のボリュームにすることができる。現在のシステム上に VxVM ボリュームが構成されている場合は、lucreate コマンドを使用して新しいブート環境を作成できます。新しいブート環境にデータをコピーすると、Veritas ファイルシステム構成が失われ、新しいブート環境に UFS ファイルシステムが作成されます。

ミラー化されたファイルシステムのスライスを選択するための指針

新しく作成するブート環境には、物理ディスクスライス、Solaris ボリュームマネージャのボリューム、および Veritas Volume Manager のボリュームを自由に組み合わせて使用できます。新しいブート環境にコピーされるクリティカルファイルシステムには、次のような種類があります。

- 物理スライス。
- RAID-1 ボリューム (ミラー) に含まれる単一スライスの連結。ルート (/) ファイルシステムが置かれているスライスは、RAID-1 ボリュームでもかまいません。
- RAID-0 ボリュームに含まれる単一スライスの連結。ルート (/) ファイルシステムが置かれているスライスは、RAID-0 ボリュームでもかまいません。

新しいブート環境を作成する際、`lucreate - m` コマンドは、次の 3 種類のデバイスを認識します。

- 物理スライス (`/dev/dsk/cwtxdysz`)
- Solaris ボリュームマネージャのボリューム (`/dev/md/dsk/dnum`)
- Veritas Volume Manager のボリューム (`/dev/vx/dsk/volume_name`)。現在のシステム上に VxVM ボリュームが構成されている場合は、`lucreate` コマンドを使用して新しいブート環境を作成できます。新しいブート環境にデータをコピーすると、Veritas ファイルシステム構成が失われ、新しいブート環境に UFS ファイルシステムが作成されます。

注 - Veritas VxVM のアップグレードで問題が生じる場合は、198 ページの「Veritas VxVm の実行中に Solaris Live Upgrade を使用してアップグレードするとシステムパニックが発生する」を参照してください。

RAID-1 ボリューム (ミラー) ファイルシステムを作成するための一般的な指針

次の指針を使用して、RAID-1 ボリュームが使用中または再同期中でないかどうか、あるいは Solaris Live Upgrade ブート環境が使用しているファイルシステムがボリュームに含まれていないかどうかを確認してください。

ボリュームの命名に関する指針については、『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の「カスタム JumpStart と Solaris Live Upgrade を行うときの RAID ボリューム名の要件とガイドライン」を参照してください。

ボリュームのステータスの確認

ミラーやサブミラーが保守を必要としている場合や使用中である場合、コンポーネントを切り離すことはできません。新しいブート環境を作成して `detach` キーワードを使用す

る前に、`metastat` コマンドを実行してください。`metastat` コマンドは、ミラーが再同期の処理中かどうか、または使用中かどうかを確認します。詳細は、`metastat(1M)` のマニュアルページを参照してください。

ボリュームの切り離しとミラーの再同期

`detach` キーワードを使ってサブミラーを切り離す場合、`lucreate` コマンドは、デバイスが再同期の処理中かどうかを確認します。デバイスが再同期中である場合、サブミラーを切り離すことはできず、エラーメッセージが表示されます。

再同期処理とは、次のような問題のあとで、あるサブミラーから別のサブミラーにデータをコピーする処理のことです。

- サブミラーの障害。
- システムのクラッシュ。
- オフラインであったサブミラーがオンラインに復帰。
- 新しいサブミラーの追加。

再同期処理の詳細は、『Solaris ボリュームマネージャの管理』の「RAID-1 ボリューム (ミラー) の再同期」を参照してください。

Solaris ボリュームマネージャのコマンドの使用

非アクティブなブート環境のボリュームを操作するには、Solaris ボリュームマネージャのコマンドではなく `lucreate` コマンドを使用します。Solaris ボリュームマネージャソフトウェアにはブート環境に関する考慮はありませんが、`lucreate` コマンドでは、ブート環境を誤って破棄しないように確認が行われます。たとえば、`lucreate` では、Solaris ボリュームマネージャのボリュームの上書きや削除が防止されます。

ただし、Solaris ボリュームマネージャを使って複雑な連結、ストライプ、ミラーなどを作成した場合、それらのボリュームコンポーネントの操作には Solaris ボリュームマネージャを使用する必要があります。Solaris Live Upgrade では、これらのコンポーネントを認識して使用できます。Solaris ボリュームマネージャのコマンドでボリュームコンポーネントを作成、変更、または破棄する前に、`lustatus` コマンドまたは `lufslist` コマンドを実行してください。これらのコマンドを使用すると、Solaris Live Upgrade ブート環境で使用されているファイルシステムがどの Solaris ボリュームマネージャボリュームに置かれているかを確認できます。

スワップファイルシステムのスライスを選択するための指針

ここでは、スワップスライスの構成に関する推奨事項と例を示します。

新しいブート環境のスワップの構成

`lucreate` コマンドの `-m` オプションを使って、3 通りの方法でスワップスライスを構成できます。

- スワップスライスを指定しないと、現在のブート環境のスワップスライスが新しいブート環境用に構成されます。
- スワップスライスを1つ以上指定すると、それらのスワップスライスだけが新しいブート環境で使用されます。この場合指定したスワップスライスは2つのブート環境の間で共有されません。
- スワップスライスを共有すると同時に、新しいスライスを追加することもできます。

3通りのスワップ構成の例を次に示します。現在のブート環境では、ルート (/) ファイルシステムが `c0t0d0s0` 上に構成されています。スワップファイルシステムは `c0t0d0s1` 上に構成されています。

- 次の例では、スワップスライスを指定していません。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。`c0t0d0s1` 上のスワップが、現在のブート環境と新しいブート環境の間で共有されます。

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs
```

- 次の例では、スワップスライスを指定しています。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。新しいスワップファイルシステムが `c0t1d0s1` 上に作成されます。現在のブート環境と新しいブート環境の間でスワップスライスは共有されません。

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs -m -:/dev/dsk/c0t1d0s1:swap
```

- 次の例では、スワップスライスを1つ追加すると同時に、別のスワップスライスを2つのブート環境で共有しています。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。新しいスワップスライスが `c0t1d0s1` 上に作成されます。`c0t0d0s1` 上のスワップスライスが、現在のブート環境と新しいブート環境の間で共有されます。

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs -m -:shared:swap -m -:/dev/dsk/c0t1d0s1:swap
```

スワップの使用中に起きるブート環境作成の失敗

現在のブート環境以外のブート環境によってスワップスライスが使用されている場合、ブート環境の作成は失敗します。`-s` オプションを使って作成されたブート環境の場合、代替ソースブート環境だけはスワップスライスを使用していてもかまいませんが、それ以外のブート環境が使用してはいけません。

共有可能なファイルシステムのスライスを選択するための指針

Solaris Live Upgrade は、スライスの内容を指定した新しいブート環境のスライスにコピーします。容量とコピーにかかる時間を節約する場合は、そのスライス上に複数のブート環境で共有できるだけの大きなファイルシステムを用意することもできます。ルート (/) や `/var` など、OS に欠かせないクリティカルファイルシステムは必ずコピーしてください。

い。/homeなどの非クリティカルファイルシステムは、複数のブート環境で共有できます。共有可能なファイルシステムは、ユーザーによって定義され、アクティブブート環境と新しいブート環境の両方の個々のスワップスライス上に存在していなければなりません。必要に応じて、複数の方法でディスクを再構成できます。

ディスクの再構成	例	参照先
新しいブート環境を作成する前にディスクスライスを作成し直し、そのスライス上に共有可能なファイルシステムを配置することができます。	たとえば、ルート(/)ファイルシステム、/var、/homeがすべて同じスライス上にある場合、ディスクを再構成して/homeを固有のスライスに配置できます。デフォルトの設定では、新しいブート環境を作成すると、/homeはアクティブブート環境と新しいブート環境で共有されます。	format(1M)
ディレクトリを共有する場合、そのディレクトリを固有のスライスに配置する必要があります。こうすることにより、ディレクトリは、そのほかのブート環境と共有可能なファイルシステムになります。lucreate -m コマンドを実行すると、新しいブート環境が作成され、ディレクトリを固有のスライスに配置することができます。しかし、この新しいファイルシステムはまだ元のブート環境と共有できません。再度 lucreate -m コマンドを実行して、もう1つ別のブート環境を作成する必要があります。この2つの新しいブート環境では、ディレクトリを共有できます。	たとえば、Solaris 9 リリースから Solaris 10 11/06 リリースにアップグレードし、両方の OS で /home を共有する場合は、lucreate -m コマンドを実行します。/home を独立したファイルシステムとして専用のスライス上に持つ Solaris 9 リリースを作成できます。次に、再度 lucreate -m コマンドを実行し、そのブート環境を複製します。この3番目のブート環境を Solaris 10 11/06 リリースへアップグレードします。/home は Solaris 9 リリースと Solaris 10 11/06 リリース間で共有されます。	共有可能なファイルシステムおよびクリティカルファイルシステムの概要については、 22 ページ の「 ファイルシステムのタイプ 」を参照してください。

新しいブート環境の内容のカスタマイズ

新しいブート環境を作成するときに、ディレクトリやファイルの一部を新しいブート環境へのコピーから除外できます。ディレクトリを除外した場合は、そのディレクトリ内にある特定のファイルやサブディレクトリが含まれるようにすることが可能です。これらのサブディレクトリは、新しいブート環境にコピーされます。たとえば、/etc/mail にあるすべてのファイルとディレクトリを除外するが、/etc/mail/staff にあるすべてのファイルとディレクトリは含まれるように指定できます。次のコマンドでは、staff サブディレクトリが新しいブート環境にコピーされます。

```
# lucreate -n second_disk -x /etc/mail -y /etc/mail/staff
```



注意-ファイル除外オプションは、注意して使用してください。システムに必要なファイルやディレクトリは削除しないでください。

次の表に、lucreate コマンドでディレクトリやファイルを除外または追加するためのオプションを示します。

指定方法	除外用のオプション	追加用のオプション
ディレクトリまたはファイルの名前を指定します	<code>-x exclude_dir</code>	<code>-y include_dir</code>
対象のファイルやディレクトリのリストを含むファイルを使用します	<code>-f list_filename</code> <code>-z list_filename</code>	<code>-Y list_filename</code> <code>-z list_filename</code>

ブート環境の作成時にディレクトリやファイルをカスタマイズする例については、91ページの「ブート環境の作成と内容のカスタマイズ(コマンド行インタフェース)」を参照してください。

ブート環境間でのファイルの同期

新しいブート環境に切り替えてアクティブにする準備ができれば、新しいブート環境をアクティブにしてリブートするだけです。新たに作成したブート環境を初めて起動するとき、ブート環境間でファイルの同期がとられます。ここでいう「同期」とは、前にアクティブであったブート環境にあるシステムファイルやディレクトリを、ブートされているブート環境にコピーすることです。変更されているファイルやディレクトリがコピーされます。

/etc/lu/synclist へのファイルの追加

Solaris Live Upgrade では、変更されているクリティカルなファイルがチェックされます。クリティカルなファイルの内容が2つのブート環境で異なっている場合、そのファイルはアクティブなブート環境から新しいブート環境にコピーされます。ファイルの同期は、新しいブート環境の作成後に `/etc/passwd` や `/etc/group` などのクリティカルなファイルが変更された場合のために用意されています。

`/etc/lu/synclist` ファイルには、同期するディレクトリやファイルのリストが記述されています。アクティブなブート環境から新しいブート環境にほかのファイルをコピーする場合もあるでしょう。必要に応じて、ディレクトリやファイルを `/etc/lu/synclist` に追加できます。

`/etc/lu/synclist` のリストにないファイルを追加すると、システムをブートできなくなる場合があります。同期処理では、ファイルのコピーとディレクトリの作成だけが行われます。ファイルやディレクトリの削除は行われません。

次の `/etc/lu/synclist` ファイルの例は、このシステムで同期される標準のディレクトリとファイルを示しています。

```
/var/mail                OVERWRITE
/var/spool/mqueue        OVERWRITE
/var/spool/cron/crontabs OVERWRITE
```

/var/dhcp	OVERWRITE
/etc/passwd	OVERWRITE
/etc/shadow	OVERWRITE
/etc/opasswd	OVERWRITE
/etc/oshadow	OVERWRITE
/etc/group	OVERWRITE
/etc/pwhist	OVERWRITE
/etc/default/passwd	OVERWRITE
/etc/dfs	OVERWRITE
/var/log/syslog	APPEND
/var/adm/messages	APPEND

次のディレクトリやファイルなどは、`synclist` ファイルに追加してもよいでしょう。

/var/yp	OVERWRITE
/etc/mail	OVERWRITE
/etc/resolv.conf	OVERWRITE
/etc/domainname	OVERWRITE

`synclist` ファイルのエントリは、ファイルまたはディレクトリです。2番目のフィールドは、ブート環境をアクティブ化するときに行われる更新の方法を示します。ファイルの更新には3通りの方法があります。

- **OVERWRITE** - 新しいブート環境のファイルの内容を、アクティブなブート環境のファイルの内容で上書きします。2番目のフィールドに動作を指定しない場合は、**OVERWRITE** がデフォルトの動作となります。エントリがディレクトリである場合は、サブディレクトリもすべてコピーされます。すべてのファイルが上書きされます。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。
- **APPEND** - 新しいブート環境のファイルの末尾に、アクティブなブート環境のファイルの内容を追加します。この追加によってファイル内のエントリが重複することがあります。ディレクトリには**APPEND** 動作を指定することはできません。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。
- **PREPEND** - 新しいブート環境のファイルの先頭に、アクティブなブート環境のファイルの内容を追加します。この追加によってファイル内のエントリが重複することがあります。ディレクトリには**PREPEND** 動作を指定することはできません。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。

ブート環境間での強制的な同期

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。最初にブートと同期を行なったあとは、要求しない限り Solaris Live Upgrade は同期処理を行いません。

- CUI を使って強制的に同期を行うには、プロンプトに対して **yes** と入力します。

- CLI を使って強制的に同期を行うには、`luactivate` コマンドの `-s` オプションを使用します。

複数のバージョンの Solaris OS を使用していると、強制的に同期を行いたい場合があります。email や passwd/group などのファイルに加えた変更を、アクティブにするブート環境に反映させたい場合があります。強制的に同期を実行すると、Solaris Live Upgrade は、同期をとるファイルの間に矛盾がないかチェックします。新しいブート環境がブートされ、矛盾が検出されると、警告が出されます。この場合、ファイルの同期は行われません。このような場合でも、アクティブ化は正常に終了します。新しいブート環境とアクティブなブート環境の両方で同じファイルに変更を加えると、矛盾が発生することがあります。たとえば、元のブート環境で /etc/passwd ファイルに変更を加えます。そして、新しいブート環境で /etc/passwd ファイルに別の変更を加えた場合などです。このような場合、同期処理では、どちらのファイルをコピーするべきか判断できません。



注意-以前のアクティブブート環境で発生した変更によりユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、このオプションを使用するには十分注意してください。たとえば、現在のブート環境で Solaris 10 11/06 ソフトウェアを実行して、強制的な同期処理を行なったあとで、Solaris 9 リリースにブート環境を戻したとします。この場合、Solaris 9 リリースでファイルが変更されることがあります。ファイルは OS のリリースに依存しているため、Solaris 9 リリースのブートは失敗する可能性があります。Solaris 10 11/06 のファイルと Solaris 9 のファイルは互換性があるとは限らないからです。

x86:GRUB メニューを使ったブート環境のアクティブ化

Solaris 10 11/06 以降のリリースでは、GRUB ブートメニューにブート環境間で切り替えを行うためのオプションの方法があります。GRUB メニューは、`luactivate` コマンドまたは「Activate」メニューによるアクティブ化に代わるものです。

作業	インフォメーション
GRUB メニューを使ってブート環境をアクティブ化する	126 ページの「x86:GRUB メニューを使ってブート環境をアクティブ化する(コマンド行インタフェース)」
GRUB メニューを使って元のブート環境にフォールバックする	133 ページの「x86:新しいブート環境のアクティブ化に成功した場合の GRUB メニューを使ったフォールバック」
GRUB の概要と計画に関する情報	『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の第 6 章「Solaris インストールのための GRUB ベースのブート」

作業	インフォメーション
GRUB の全概要とシステム管理作業について	『Solaris のシステム管理 (基本編)』

リモートシステムからの **Solaris Live Upgrade** の使用

キャラクタユーザーインターフェースを tip 回線などを介してリモートで表示する場合は、必要に応じて TERM 環境変数を VT220 に設定してください。また、共通デスクトップ環境 (CDE) を使用する場合は、TERM 変数の値を xterm ではなく dtterm に設定してください。

Solaris Live Upgrade によるブート環境の作成 (作業)

この章では、Solaris Live Upgrade のインストール、メニューの使用、およびブート環境の作成について説明します。この章の内容は次のとおりです。

- 57 ページの「Solaris Live Upgrade インタフェースの概要」
- 58 ページの「Solaris Live Upgrade メニューの使用 (CUI)」
- 59 ページの「作業マップ: Solaris Live Upgrade のインストールとブート環境の作成」
- 59 ページの「Solaris Live Upgrade のインストール」
- 63 ページの「Solaris Live Upgrade の起動と停止 (キャラクタユーザーインタフェース)」
- 64 ページの「新しいブート環境の作成」

Solaris Live Upgrade インタフェースの概要

Solaris Live Upgrade は、キャラクタユーザーインタフェース (CUI) を介して使用することもコマンド行インタフェース (CLI) で使用することも可能です。次の説明では、CUI と CLI の両方の場合について手順を示してあります。

インタフェースのタイプ	説明
キャラクタユーザーインタフェース (CUI)	CUI では、Solaris Live Upgrade の一部の機能にアクセスできません。CUI は、複数バイトロケールおよび 8 ビットロケールでは実行できません。
コマンド行インタフェース (CLI)	ここで説明する CLI の手順は、Solaris Live Upgrade コマンドの基本的な使い方を示すものです。コマンド一覧については、第 10 章を参照してください。また、これらのコマンドで使用するオプションの詳細は、関連する各マニュアルページを参照してください。

Solaris Live Upgrade メニューの使用 (CUI)

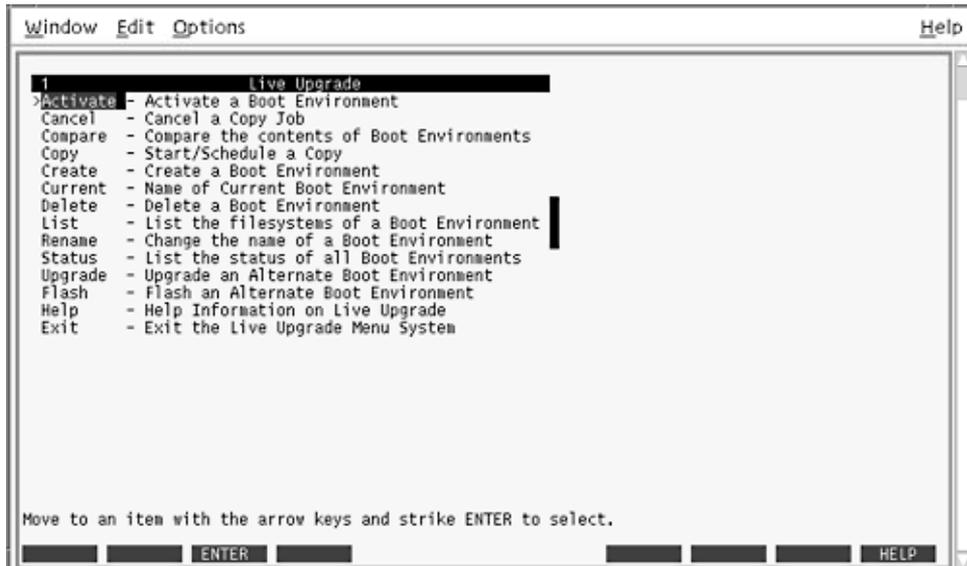


図 4-1 Solaris Live Upgrade のメインメニュー

Solaris Live Upgrade キャラクタユーザーインタフェースのメニュー間の移動には、矢印キーとファンクションキーを使用します。上下に移動する場合やフィールド内にカーソルを置く場合は、矢印キーを使用してください。処理を実行する場合は、ファンクションキーを使用してください。メニューの最下部には、キーボード上のファンクションキーを示す黒い矩形が表示されます。たとえば、最初の黒い矩形は F1、2 番目の黒い矩形は F2 をそれぞれ表します。有効になっている矩形には、「Save」などの処理を示す語句が表示されています。「Configuration」メニューでは、矩形ではなくファンクションキーの番号と対応する動作が表示されます。

- F3 を使用すると、どのメニューの場合も作業の保存 (SAVE) が行われてそのメニューが終了します。
- F6 を使用すると、どのメニューにおいても作業が取り消され (CANCEL)、変更の保存が行われないままメニューが終了します。
- ほかのファンクションキーの動作は、メニューによって異なります。

次に説明する作業において、ファンクションキーを押すように指示されています。キーボード上のキーが Solaris Live Upgrade メニュー上のファンクションキーと適切に対応していない場合は、Control-F および該当する番号を使用してください。

作業マップ: Solaris Live Upgrade のインストールとブート環境の作成

表 4-1 作業マップ: Solaris Live Upgrade の使用

作業	説明	参照先
システムにパッチをインストールします	Solaris Live Upgrade には、特定のパッチリビジョンセットが必要です	60 ページの「Solaris Live Upgrade に必要なパッチのインストール」
Solaris Live Upgrade パッケージをインストールします	OS にパッケージをインストールします	59 ページの「Solaris Live Upgrade のインストール」
Solaris Live Upgrade を起動します	Solaris Live Upgrade のメインメニューを起動します	63 ページの「Solaris Live Upgrade の起動と停止 (キャラクタユーザーインターフェース)」
ブート環境を作成します	非アクティブブート環境にファイルシステムをコピーして再構成します	64 ページの「新しいブート環境の作成」

Solaris Live Upgrade のインストール

現在の OS に Solaris Live Upgrade パッケージをインストールする必要があります。アップグレード後の OS のリリース番号と同じリリース番号の Solaris Live Upgrade パッケージをインストールする必要があります。たとえば、OS を現在使用している Solaris 9 リリースから Solaris 10 11/06 リリースにアップグレードする場合、Solaris 10 11/06 リリースの Solaris Live Upgrade パッケージをインストールする必要があります。

いくつかのパッチが必要になる場合があります。Solaris Live Upgrade パッケージをインストールする前に、これらのパッチをインストールします。詳細については、以下を参照してください。

- 60 ページの「Solaris Live Upgrade に必要なパッチのインストール」
- 61 ページの「pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法」
- 61 ページの「Solaris インストールプログラムを使用して Solaris Live Upgrade をインストールする方法」

Solaris Live Upgrade に必要なパッチのインストール

説明	参照先
<p>注意 - Solaris Live Upgrade を正しく操作するためには、指定の OS バージョン用の特定のパッチリビジョンのセットがインストールされている必要があります。Solaris Live Upgrade をインストールまたは実行する前に、これらのパッチをインストールする必要があります。</p> <p>x86 のみ - このパッチのセットがインストールされていない場合、Solaris Live Upgrade は失敗し、次のエラーメッセージが表示されることがあります。次のエラーメッセージが表示されなくても、必要なパッチがインストールされていない場合があります。Solaris Live Upgrade のインストールを試みる前に、SunSolve の infodoc に記載されたすべてのパッチがすでにインストール済みであることを必ず確認してください。</p> <pre>ERROR: Cannot find or is not executable: </sbin/biosdev>. ERROR: One or more patches required by Live Upgrade has not been installed.</pre> <p>infodoc 72099 に記載されたパッチは、随時変更される可能性があります。これらのパッチにより、Solaris Live Upgrade の欠陥が修正される可能性があると同時に、Solaris Live Upgrade が依存するコンポーネントの欠陥も修正される可能性があります。Solaris Live Upgrade で問題が発生した場合は、最新の Solaris Live Upgrade パッチがインストールされていることを確認してください。</p>	<p>http://sunsolve.sun.com で最新のパッチリストを確認してください。SunSolve の Web サイトで、infodoc 72099 を検索してください。</p>
<p>Solaris 8、または Solaris 9 OS を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。</p>	<p>Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、Java 2 Runtime Environment 推奨パッチクラスタをインストールします。このパッチクラスタは http://sunsolve.sun.com から入手できます。</p>

▼ 必要なパッチをインストールするには

- 1 SunSolveSM の Web サイトから、パッチリストを取得します。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 patchadd コマンドを使用してパッチをインストールします。

```
# patchadd path_to_patches
```
- 4 必要に応じてシステムをリブートします。いくつかのパッチは、有効にするためにリブートする必要があります。
x86 のみ: システムをリブートする必要があります。そうしないと、Solaris Live Upgrade は失敗します。

▼ pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 次の順序でパッケージをインストールします。

```
# pkgadd -d path_to_packages SUNWlur SUNWluu
```

path_to_packages ソフトウェアパッケージのあるディレクトリの絶対パスを指定します。
- 3 指定したパッケージが正常にインストールされていることを確認します。

```
# pkgchk -v SUNWlur SUNWluu
```

▼ Solaris インストールプログラムを使用して Solaris Live Upgrade をインストールする方法

注-この手順では、システムでボリュームマネージャーを実行していると仮定しています。ボリュームマネージャーを使用したリムーバブルメディアの管理の詳細は、『Solaris のシステム管理 (デバイスとファイルシステム)』を参照してください。

- 1 Solaris Operating System DVD または Solaris SOFTWARE - 2 CD を挿入します。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 インストーラを実行します。
 - Solaris Operating System DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。
 - SPARC システムの場合:

```
# cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
# ./liveupgrade20
```
 - x86 システムの場合:

```
# cd /cdrom/cdrom0/Solaris_10/Tools/Installers
# ./liveupgrade20
```Solaris インストールプログラムの GUI が表示されます。
 - Solaris SOFTWARE - 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```

Solaris インストールプログラムの GUI が表示されます。
- 4 「インストール形式の選択 (Select Type of Install)」パネルで「カスタム (Custom)」をクリックします。
- 5 「ロケールの選択 (Locale Selection)」パネルで、インストールする言語をクリックします。
- 6 インストールするソフトウェアを選択します。
 - DVD の場合、「コンポーネントの選択 (Component Selection)」パネルの「次へ (Next)」をクリックしてパッケージをインストールします。
 - CD の場合、「製品の選択 (Product Selection)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (Default Install)」をクリックします。また、ソフトウェアの選択を解除するには、ほかの製品をクリックします。
- 7 Solaris インストールプログラムの指示に従って、ソフトウェアをインストールします。

Solaris Live Upgrade の起動と停止 (キャラクターユーザーインターフェース)

Solaris Live Upgrade のメニュープログラムの起動と停止方法について説明します。

▼ Solaris Live Upgrade メニューを起動する

注-キャラクターユーザーインターフェースを tip 回線などを介してリモートで表示する場合は、必要に応じて TERM 環境変数を VT220 に設定してください。また、共通デスクトップ環境 (CDE) を使用する場合は、TERM 変数の値を xterm ではなく dtterm に設定してください。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/lu
```

Solaris Live Upgrade のメインメニューが表示されます。

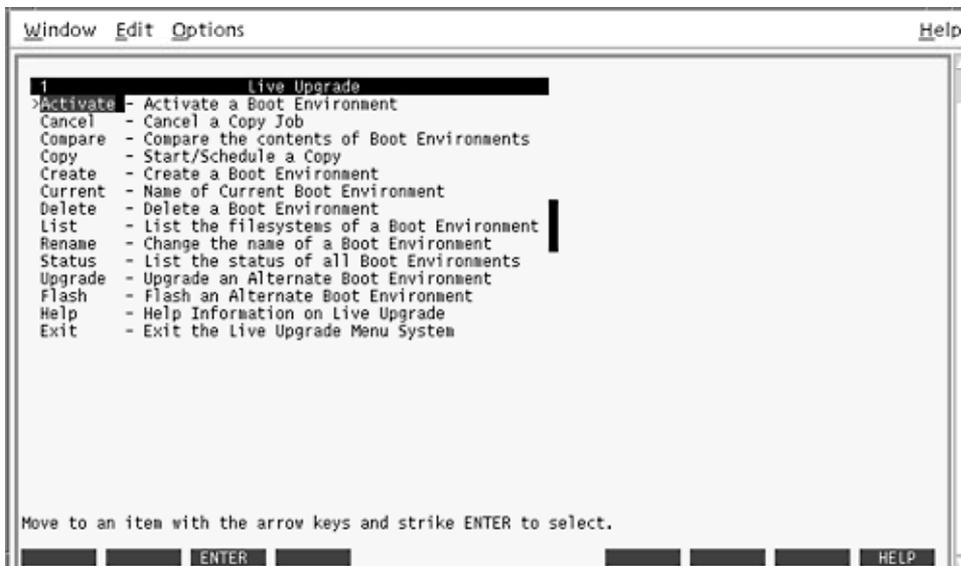


図 4-2 Solaris Live Upgrade のメインメニュー

▼ Solaris Live Upgrade のメニューを終了するには

- ▶ メインメニューの「Exit」を選択します。

新しいブート環境の作成

ブート環境を作成すると、クリティカルファイルシステムをアクティブなブート環境から新しいブート環境にコピーできます。必要に応じてディスクを再編成し、ファイルシステムをカスタマイズして、クリティカルファイルシステムを新しいブート環境にコピーするには、CUIの「Create」メニュー、「Configuration」サブメニュー、`lucreate` コマンドを使用します。

ファイルシステムは、新しいブート環境にコピーする前にカスタマイズできます。このため、クリティカルファイルシステムディレクトリを親のディレクトリにマージすることも、親ディレクトリから分離することも可能になります。ユーザー定義の(共有可能)ファイルシステムは、デフォルトで複数のブート環境で共有されます。ただし、必要に応じて共有可能なファイルシステムをコピーすることもできます。スワップ(共有可能なファイルシステム)の分割やマージも可能です。クリティカルファイルシステムと共有可能ファイルシステムの概要については、22 ページの「ファイルシステムのタイプ」を参照してください。

▼ ブート環境を作成する(キャラクタユーザーインタフェース)

- 1 メインメニューから「Create」を選択します。
「Create a Boot Environment」サブメニューが表示されます。
- 2 アクティブブート環境(必要に応じて)と新しいブート環境の名前を入力し、確定します。アクティブブート環境の名前の入力が必要なのは、最初にブート環境を作成するときだけです。
ブート環境名は英数字で 30 文字以内とします。使用できるのは英数文字だけで、マルチバイト文字を使用することはできません。
Name of Current Boot Environment: **solaris8**
Name of New Boot Environment: **solaris10**
- 3 **F3** を押して変更を保存します。
「Configuration」メニューが表示されます。

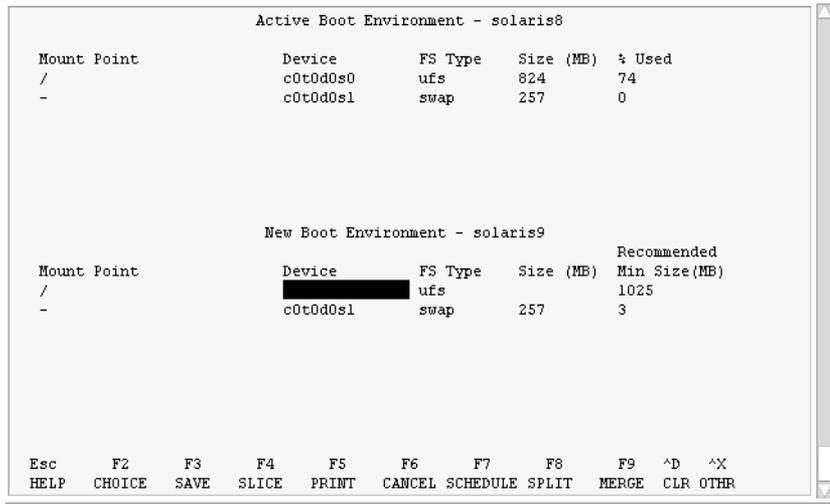


図 4-3 Solaris Live Upgrade の「Configuration」メニュー

「Configuration」メニューには次の項目があります。

- 元のブート環境は、画面の上部に表示されます。画面の下部には作成されるブート環境が表示されます。
- 「Device」フィールドには次の情報が含まれています。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます。
 - Solaris ボリュームマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
 - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます。
 - クリティカルファイルシステムの選択領域は、クリティカルファイルシステムを選択するまで空白のままです。/usr、/var、/opt などのクリティカルファイルシステムは、分割可能です。ルート (/) ファイルシステムとマージすることもできます。
 - /export やスワップなどの共有可能なファイルシステムは「Device」フィールドに表示されます。これらのファイルシステムは、ソースブート環境とターゲットブート環境の両方に同じマウントポイントを持っています。スワップはデフォルトで共有されますが、スワップスライスの分割とマージ(追加と削除)も行うことができます。
 クリティカルファイルシステムと共有可能ファイルシステムの概要については、[22 ページの「ファイルシステムのタイプ」](#)を参照してください。
- 「FS_Type」フィールドでは、ファイルシステムの種類を変更できます。ファイルシステムの種類は、次のいずれかになります。
 - vxfs: Veritas ファイルシステムを示します

- `swap`: スワップファイルシステムを示します
- `ufs`: UFS ファイルシステムを示します

4 (省略可能) 次の作業は、必要に応じて行うことができます。

- 画面上の情報を ASCII ファイルに出力するには F5 を押します。
- ファイルシステムのリストをスクロールするには、Control-X を押します。
この操作で、アクティブブート環境のファイルシステムと新しいブート環境のファイルシステムを切り替えてスクロールできるようになります。
- 「Configuration」メニューを閉じるには、F6 を押します。
 - 「Configuration」メニューが表示されている場合は、変更が保存されず、ファイルシステムは修正されません。
 - 「Configuration」サブメニューが表示されている場合は、「Configuration」メニューに戻ります。

5 F2 を押して、利用できるスライスを選択します。

「Choices」メニューでは、カーソルが置かれているフィールドに対して、そのシステム上で利用できるスライスが表示されます。このメニューが利用できるフィールドは「Device」と「FS_Type」です。

- a. 矢印キーを使用してフィールド内にカーソルを置き、スライスまたはファイルシステムの種類を選択します。
 - 「Device」フィールドにカーソルを置くと、すべての空きスライスが表示されます。ルート (/) ファイルシステムの場合、「Choices」メニューに表示されるのはルート (/) ファイルシステムの制限事項を満たす空きスライスだけです。[48 ページの「ルート \(/\) ファイルシステムのスライスを選択するための指針」](#)を参照してください。
 - 「FS_Type」フィールドにカーソルを置くと、利用できるすべてのファイルシステムタイプが表示されます。
 - 現在のファイルシステムには、ボールド書体のスライスを選択できます。スライスのサイズは、ファイルシステムのサイズにアップグレード用に 30% を加えた値が割り当てられます。
 - ボールド書体ではないスライスは、そのファイルシステムをサポートするにはサイズが小さすぎることを意味します。ディスクスライスを作成し直す方法については、[手順 6](#) を参照してください。
- b. Return キーを押してスライスを選択します。
選択したスライスが「Device」フィールドに表示されるか、あるいは「FS_Type」フィールド内でファイルシステムの種類が変化します。

- 6 (省略可能) 空きスライスが最小要件を満たしていない場合は、**F4**を押して任意の空きスライスを分割し直してください。
- 「Solaris Live Upgrade Slice Configuration」メニューが表示されます。
- 新しいスライスを作成できるように、`format(1M)` コマンドが実行されます。画面の指示に従って新しいスライスを作成してください。
- 矢印キーを使って、「Device」フィールドと「FS_Type」フィールド間を移動できます。デバイスを選択すると、「Size (Mbytes)」フィールドに値が自動的に表示されます。
- a. デバイスを解放するには、**Control-D**を押します。
以上の操作でスライスが利用できるようになり、「Choices」メニューに表示されます。
 - b. **F3**を押して「**Configuration**」メニューに戻ります。
- 7 (省略可能) クリティカルファイルシステムを分割すると、そのファイルシステムは別々のマウントポイントに配置されます。ファイルシステムを分割するには、次の手順に従ってください。
- (ファイルシステムのマージについては、[手順8](#)を参照してください)。
- a. 分割するファイルシステムを選択します。
`/usr`、`/var`、`/opt`などのファイルシステムは、それらの親ディレクトリから分割できます。
-
- 注-ブート環境のファイルシステムを作成する場合のルールは、Solaris OSのファイルシステムを作成する場合と同じです。Solaris Live Upgradeでは、クリティカルファイルシステムに無効な構成を作成してしまうことを回避できません。たとえば、`lucreate` コマンドを入力して、ルート(`/`)と`/kernel`に別々のファイルシステムを作成することができませんが、このようにルート(`/`)ファイルシステムを分割するのは誤りです。
-
- b. **F8**を押します。
 - c. 新しいブート環境のファイルシステム名を入力します。例:
Enter the directory that will be a separate file system
on the new boot environment: **/opt**
新しいファイルシステムが検証されると、画面に新しい行が追加されます。
 - d. **F3**を押して「**Configuration**」メニューに戻ります。
「Configuration」メニューが表示されます。

- 8 (省略可能) マージを行うと、ファイルシステムは同じマウントポイントに配置されます。ファイルシステムをその親ディレクトリにマージするには、次の操作を行います。(ファイルシステムの分割については、[手順7](#)を参照してください)。
 - a. マージするファイルシステムを選択します。
/usr、/var、/optなどのファイルシステムをそれらの親ディレクトリにマージできません。
 - b. **F9**を押します。
次の例のように、結合されるファイルシステムが表示されます。
/opt will be merged into /.
 - c. Returnキーを押します。
 - d. **F3**を押して「**Configuration**」メニューに戻ります。
「**Configuration**」メニューが表示されます。
- 9 (省略可能) スワップスライスを追加するか削除するかを決定します。
 - スワップスライスを分割して新しいスライスに配置する場合は、[手順10](#)に進みます。
 - スワップスライスを削除する場合は、[手順11](#)に進みます。
- 10 (省略可能) スワップスライスを分割するには、次の手順に従います。
 - a. 「**Device**」フィールドで、分割するスワップスライスを選択します。
 - b. **F8**を押します。
 - c. プロンプトに対して、次のコマンドを入力します。
Enter the directory that will be a separate filesystem on
the new BE: **swap**
 - d. **F2(Choice)**を押します。
「**Choice**」メニューに、スワップに利用できるスライスが表示されます。
 - e. スワップを配置するスライスを選択します。
そのスライスが「**Device**」フィールドに表示され、スワップの新しいスライスとなります。
- 11 (省略可能) スワップスライスを削除するには、次の手順に従います。
 - a. 「**Device**」フィールドで、削除するスワップスライスを選択します。

- b. F9を押します。
- c. プロンプトが表示されたら「y」と入力します。

```
Slice /dev/dsk/c0t4d0s0 will not be swap partition.  
Please confirm? [y, n]: y
```

このスワップスライスが削除されました。

- 12 今すぐにブート環境を作成するか、あとで作成するようにスケジュールするかを決定します。

- すぐに新しいブート環境を作成する場合は、F3を押します。
 構成が保存され、構成画面が閉じます。このファイルシステムがコピーされ、ブート環境がブート可能になり、非アクティブブート環境が作成されます。
 ブート環境の作成には、システム構成に応じて、1時間以上かかる場合があります。続いて、Solaris Live Upgrade メインメニューが表示されます。
- 後で作成されるようにスケジュールする場合は、次の例に示すように「y」と入力し、続いて開始時刻と電子メールアドレスを入力します。

```
Do you want to schedule the copy? y  
Enter the time in 'at' format to schedule create: 8:15 PM  
Enter the address to which the copy log should be mailed: someone@anywhere.com
```

処理の完了は、電子メールで通知されます。

時間の書式については、at(1)のマニュアルページを参照してください。

スケジュールできるのは一度に1つのジョブだけです。

作成が完了すると、非アクティブブート環境をアップグレードできるようになります。第5章を参照してください。

▼ ブート環境をはじめて作成する(コマンド行インタフェース)

lucreate コマンドに -m オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。たとえば、-m オプションを1回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のすべてのファイルシステムが、-m で指定されたファイルシステムにマージされます。-m オプションを2回使用すると、ファイルシステムが2つ作成されます。-m オプションを使ってファイルシステムを作成する場合、次の項目に留意してください。

- `-m` オプションを1個指定して、新しいブート環境のルート (*/*) ファイルシステムを作成する必要があります。`-m` オプションを指定しないで `lucreate` を実行すると、「Configuration」メニューが表示されます。「Configuration」メニューでは、新しいマウントポイントにファイルを変更して新しいブート環境をカスタマイズできます。
- 現在のブート環境にクリティカルファイルシステムがある場合、このファイルシステムは、`-m` オプションで指定しなくても新しく作成されたファイルシステムの上位2番目のファイルシステムにマージされます。
- 新しいブート環境には、`-m` オプションで指定されたファイルシステムだけが作成されます。現在のブート環境に複数のファイルシステムがあり、新しいブート環境にも同じ数のファイルシステムを作成する場合は、ファイルシステムごとに1個ずつ `-m` オプションを指定します。たとえば、ルート (*/*) ファイルシステム、`/opt` ファイルシステム、`/var` ファイルシステムがある場合、これらを新しいブート環境に作成するには、それぞれに `-m` オプションを1個ずつ指定します。
- マウントポイントが重複しないようにしてください。たとえば、ルート (*/*) ファイルシステムを2つ作成することはできません。

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のように入力して新しいブート環境を作成します。

```
# lucreate [-A 'BE_description'] -c BE_name \
-m mountpoint:device[,metadevice]:fs_options [-m ...] -n BE_name
```

`-A 'BE_description'`

(省略可能) ブート環境名 (`BE_name`) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

`-c BE_name`

アクティブブート環境に名前 `BE_name` を割り当てます。このオプションは省略可能で、最初のブート環境を作成する場合だけ使用されます。`lucreate` を初めて実行する場合に `-c` オプションを省略すると、デフォルトの名前が作成されます。

デフォルトの名前は、次の基準に従って選択されます。

- 物理ブートデバイスが判別可能な場合は、その物理ブートデバイスのベース名が現在のブート環境の名前になります。

たとえば、物理ブートデバイスが `/dev/dsk/c0t0d0s0` であれば、現在のブート環境には `c0t0d0s0` という名前が与えられます。

- 物理ブートデバイスが判別不可能な場合は、`uname` コマンドの `-s` オプションと `-r` オプションで取得される名前が組み合わせられます。

たとえば、`uname -s` で取得される OS の名前が `SunOS`、`uname -r` で取得されるリリース名が `5.9` であれば、現在のブート環境には `SunOS5.9` という名前が与えられます。

- 上記のどちらの方法でも名前を決定できない場合、現在のブート環境には `current` という名前が与えられます。

注-最初のブート環境を作成した後は、`-c` オプションを指定しても無視されるか、エラーメッセージが表示されます。

- 現在のブート環境と同じ名前を指定すると、このオプションは無視されます。
- 現在のブート環境と異なる名前を指定すると、作成は失敗し、エラーメッセージが表示されます。次の例は、ブート環境の名前によってエラーメッセージが発生するようすを示しています。

```
# lucurr
c0t0d0s0
# lucreate -c /dev/dsk/c1t1d1s1 -n newbe -m /:/dev/dsk/c1t1d1s1:ufs
ERROR: current boot environment name is c0t0d0s0: cannot change
name using <-c c1t1d1s1>
```

`-m mountpoint:device[,metadevice]:fs_options [-m ...]`

新しいブート環境のファイルシステム構成を `fstab` で指定します。 `-m` に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す `-` (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。 `/dev/dsk/cwtxdydz` の形式で表されます
 - Solaris ボリュームマネージャーのボリューム名。 `/dev/md/dsk/dnum` の形式で表されます
 - Veritas Volume Manager のボリューム名。 `/dev/md/vxfs/dsk/dnum` の形式で表されます
 - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- `fs_options` フィールドには、次のいずれかを指定できます。
 - `ufs`: UFS ファイルシステムを示します。
 - `vxfs`: Veritas ファイルシステムを示します。
 - `swap`: スワップファイルシステムを示します。スワップマウントポイントはハイフン `-` で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-n *BE_name*

作成するブート環境の名前。 *BE_name* は、システム上で一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-1 ブート環境を作成する(コマンド行)

この例では、アクティブブート環境の名前は `first_disk` です。 `-m` オプションはファイルシステムのマウントポイントを表します。ルート (`/`) ファイルシステムと `/usr` ファイルシステムが作成されます。新しいブート環境の名前は `second_disk` です。「`mydescription`」という記述は、`second_disk` に対応しています。新しいブート環境 `second_disk` のスワップは、自動的にソースである `first_disk` から共有されます。

```
# lucreate -A 'mydescription' -c first_disk -m /:/dev/dsk/c0t4d0s0:ufs \  
-m /usr:/dev/dsk/c0t4d0s3:ufs -n second_disk
```

▼ ブート環境を作成しファイルシステムをマージする(コマンド行インタフェース)

注 `-lucreate` コマンドに `-m` オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。たとえば、`-m` オプションを1回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のファイルシステムがすべて1つのファイルシステムにマージされます。`-m` オプションを2回使用すると、ファイルシステムが2つ作成されます。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# lucreate -A 'BE_description' \  
-m mountpoint:device[,metadevice]:fs_options \  
-m [...] -m mountpoint:merged:fs_options -n BE_name
```

-A *BE_description*

(省略可能) ブート環境名 (*BE_name*) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-m mountpoint:device[,metadevice]:fs_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-mに引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
 - Solaris ボリュームマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます
 - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます
 - キーワード *merged*。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- *fs_options* フィールドには、次のいずれかを指定できます。
 - *ufs*: UFS ファイルシステムを示します。
 - *vxfs*: Veritas ファイルシステムを示します。
 - *swap*: スワップファイルシステムを示します。スワップマウントポイントはハイフン(-)で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-n BE_name

作成するブート環境の名前。BE_name は、システム上で一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-2 ブート環境を作成しファイルシステムをマージする(コマンド行インタフェース)

この例の場合、現在のブート環境には、ルート(/)ファイルシステム、/usr ファイルシステム、/opt ファイルシステムがあります。/opt ファイルシステムは、親ファイルシステム /usr にマージされます。新しいブート環境の名前は second_disk です。

「mydescription」という記述は、second_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-m /usr/opt:merged:ufs -n second_disk
```

▼ ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)

注-ブート環境のファイルシステムを作成する場合のルールは、Solaris OSのファイルシステムを作成する場合と同じです。Solaris Live Upgradeでは、クリティカルファイルシステムに無効な構成を作成してしまうことを回避できません。たとえば、lucreate コマンドを入力して、ルート(/)と/kernelに別々のファイルシステムを作成することができますが、このようにルート(/)ファイルシステムを分割するのは誤りです。

1つのディレクトリを複数のマウントポイントに分割すると、ファイルシステム間でハードリンクが維持されなくなります。たとえば、/usr/stuff1/fileが/usr/stuff2/fileにハードリンクされている場合に/usr/stuff1と/usr/stuff2を別々のファイルシステムに分割すると、ファイル間のリンクは解除されます。lucreate から警告メッセージが表示され、解除されたハードリンクの代わりとなるシンボリックリンクが作成されます。

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solarisのシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

2 次のコマンドを入力します。

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
-m mountpoint:device[,metadevice]:fs_options -n new_BE
```

-A 'BE_description'

(省略可能)ブート環境名(BE_name)の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-m mountpoint:device[,metadevice]:fs_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-mに引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す-(ハイフン)を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdyszの形式で表されます
 - Solaris ポリウムマネージャのメタデバイス名。/dev/md/dsk/dnumの形式で表されます

- Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます
- キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- fs_options フィールドには、次のいずれかを指定できます。
 - ufs: UFS ファイルシステムを示します。
 - vxfs: Veritas ファイルシステムを示します。
 - swap: スワップファイルシステムを示します。スワップマウントポイントはハイフン(-)で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-n BE_name

作成するブート環境の名前。BE_name は、システム上で一意となるように指定する必要があります。

例 4-3 ブート環境を作成しファイルシステムを分割する(コマンド行インタフェース)

この例では、前述のコマンドによってルート(/)ファイルシステムを新しいブート環境内の複数のディスクスライスに分割しています。ここでは、/usr、/var、および/opt をすべてルート(/)に置いている次のソースブート環境を想定してください。

```
/dev/dsk/c0t0d0s0 /
```

新しいブート環境で、次に示すように別々のスライスにマウントすることによって、ファイルシステム /usr、/var、/opt を分割します。

```
/dev/dsk/c0t1d0s0 /
```

```
/dev/dsk/c0t1d0s1 /var
```

```
/dev/dsk/c0t1d0s7 /usr
```

```
/dev/dsk/c0t1d0s5 /opt
```

「mydescription」という記述は、ブート環境名 second_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s7:ufs \
-m /var:/dev/dsk/c0t1d0s1:ufs -m /opt:/dev/dsk/c0t1d0s5:ufs \
-n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

▼ ブート環境を作成しスワップを再構成する(コマンド行インタフェース)

スワップスライスは、デフォルトでは複数のブート環境で共有されます。-m オプションでスワップを指定「しない」場合、現在のブート環境と新しいブート環境が同じスワップスライスを共有します。新しいブート環境のスワップを構成し直す場合は、-m オプションを使用してそのブート環境に対してスワップスライスの追加または削除を行なってください。

注-スワップスライスを分割したりマージしたりするには、現在のブート環境(-s オプションを使用した場合はソースブート環境)以外のブート環境では、スワップスライスが使用中であってはならないという制限があります。ファイルシステムの種類(スワップ、ufs など)にかかわらず、スワップスライスがほかのブート環境によって使用されている場合、ブート環境の作成は失敗します。

既存のスワップスライスを使用してブート環境を作成した後、`vfstab` ファイルを編集することができます。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# lucreate [-A 'BE_description'] \
  -m mountpoint:device[,metadevice]:fs_options \
  -m -:device:swap -n BE_name
```

```
-A 'BE_description'
```

(省略可能)ブート環境名(BE_name)の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます

- Solaris ボリュームマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます
- Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます
- キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- fs_options フィールドには、次のいずれかを指定できます。
 - ufs: UFS ファイルシステムを示します。
 - vxfs: Veritas ファイルシステムを示します。
 - swap: スワップファイルシステムを示します。スワップマウントポイントはハイフン(-)で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-n BE_name

作成するブート環境の名前。BE_name は、一意となるように指定する必要があります。

スワップが別のスライスまたはデバイスに移動し、新しいブート環境が作成されます。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第 5 章を参照してください。

例 4-4 ブート環境を作成しスワップを再構成する(コマンド行インタフェース)

この例の場合、現在のブート環境には、/dev/dsk/c0t0d0s0 にルート(/)、/dev/dsk/c0t0d0s1 にスワップがあります。新しいブート環境はルート(/)を /dev/dsk/c0t4d0s0 にコピーし、/dev/dsk/c0t0d0s1 と /dev/dsk/c0t4d0s1 の両方をスワップスライスとして使用します。「mydescription」という記述は、ブート環境名 second_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \
-m -:/dev/dsk/c0t4d0s1:swap -n second_disk
```

これらのスワップ割り当ては、second_disk からブートが行われて初めて有効になります。スワップスライスが多数存在する場合は、-M オプションを使用してください。78 ページの「リストを使用してブート環境を作成しスワップを再構成する(コマンド行インタフェース)」を参照してください。

▼ リストを使用してブート環境を作成しスワップを再構成する(コマンド行インタフェース)

スワップスライスが多数存在する場合は、スワップリストを作成してください。
lucreate は、新しいブート環境のスワップスライスにこのリストを使用します。

注-スワップスライスを分割したりマージしたりするには、現在のブート環境(-s オプションを使用した場合はソースブート環境)以外のブート環境では、スワップスライスが使用中であってはならないという制限があります。スワップスライスのファイルシステムの種類(swap、ufs)などにかかわらず、スワップスライスがほかのブート環境によって使用されている場合、ブート環境の作成は失敗します。

- 1 新しいブート環境で使用されるスワップスライスのリストを作成します。このファイルの場所と名前はユーザーが決定できます。この例では、`/etc/lu/swapslices` ファイルにはデバイスとスライスが挙げられています。

```

-:/dev/dsk/c0t3d0s2:swap
-:/dev/dsk/c0t3d0s2:swap
-:/dev/dsk/c0t4d0s2:swap
-:/dev/dsk/c0t5d0s2:swap
-:/dev/dsk/c1t3d0s2:swap
-:/dev/dsk/c1t4d0s2:swap
-:/dev/dsk/c1t5d0s2:swap

```

- 2 次のコマンドを入力します。

```

# lucreate [-A 'BE_description'] \
  -m mountpoint:device[,metadevice]:fs_options \
  -M slice_list -n BE_name

```

-A 'BE_description'

(省略可能)ブート環境名(BE_name)の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-m mountpoint:device[,metadevice]:fs_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-mに引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。`/dev/dsk/cwtxdysz` の形式で表されます
 - Solaris ポリウムマネージャのメタデバイス名。`/dev/md/dsk/dnum` の形式で表されます

- Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます
- キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- fs_options フィールドには、次のいずれかを指定できます。
 - ufs: UFS ファイルシステムを示します。
 - vxfs: Veritas ファイルシステムを示します。
 - swap: スワップファイルシステムを示します。スワップマウントポイントはハイフン(-)で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-M slice_list

ファイル slice_list 中には、-m オプションのリストが記述されています。これらの引数は、-m に指定されている書式で指定してください。ハッシュ記号(#)で始まるコメント行は無視されます。-M オプションは、ブート環境用のファイルシステムが多数存在する場合に便利です。-m オプションと -M オプションは一緒に使えます。たとえば、slice_list にスワップスライスを記録しておき、-m を使用して、ルート(/)スライスと /usr スライスを指定できます。

-m オプションと -M オプションでは、特定のマウントポイントについて複数のスライスを指定できます。これらのスライスを処理する場合、lucreate は利用不可能なスライスをスキップして利用できる最初のスライスを選択します。

-n BE_name

作成するブート環境の名前。BE_name は、一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-5 リストを使用してブート環境を作成しスワップを再構成する(コマンド行インタフェース)

この例では、新しいブート環境のスワップは、/etc/lu/swapslices ファイルに挙げられている一連のスライスです。「mydescription」という記述は、second_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c02t4d0s0:ufs -m /usr:/dev/dsk/c02t4d0s1:ufs \
-M /etc/lu/swapslices -n second_disk
```

▼ ブート環境を作成し共有可能ファイルシステムをコピーする(コマンド行インタフェース)

新しいブート環境に共有可能ファイルシステムをコピーする場合は、`-m` オプションを使用してマウントポイントがコピーされるように指定してください。それ以外の場合、共有可能なファイルシステムはデフォルトで共有され、`vfstab` ファイルに指定された同じマウントポイントを使用します。共有可能ファイルシステムに対する更新は、両方の環境に適用されます。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 ブート環境を作成します。

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

```
-A 'BE_description'
```

(省略可能) ブート環境名 (BE_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
 - Solaris ポリウムマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます
 - Veritas Volume Manager のポリウム名。/dev/vx/dsk/volume_name の形式で表されます
 - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- `fs_options` フィールドには、次のいずれかを指定できます。
 - `ufs`: UFS ファイルシステムを示します。
 - `vxfs`: Veritas ファイルシステムを示します。
 - `swap`: スワップファイルシステムを示します。スワップマウントポイントはハイフン (-) で表します。

- 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

`-n BE_name`
作成するブート環境の名前。`BE_name`は、一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-6 ブート環境を作成し共有可能ファイルシステムをコピーする(コマンド行インタフェース)

この例の場合、現在のブート環境には、ルート(/)ファイルシステムと/home ファイルシステムがあります。新しいブート環境では、ルート(/)ファイルシステムがルート(/)と/usrの2つのファイルシステムに分割されます。/home ファイルシステムは新しいブート環境にコピーされます。「mydescription」という記述は、ブート環境名 second_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \  
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

▼ 別々のソースから単一のブート環境を作成(コマンド行インタフェース)

lucreate コマンドは、アクティブブート環境内のファイルシステムに基づいてブート環境を作成します。アクティブブート環境以外のブート環境に基づいてブート環境を作成する場合は、`-s` オプションを指定して lucreate を実行します。

注-新しいブート環境をアクティブにした後、フォールバックを行う必要がある場合は、ソースブート環境ではなく以前にアクティブだったブート環境に戻ります。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 ブート環境を作成します。

```
# lucreate [-A 'BE_description'] -s source_BE_name  
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A 'BE_description'

(省略可能) ブート環境名 (BE_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-s source_BE_name

新しいブート環境に対するソースブート環境を指定します。このソースはアクティブブート環境ではありません。

-m mountpoint:device[,metadevice]:fs_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- **mountpoint** には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- **device** フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
 - Solaris ポリウムマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます
 - Veritas Volume Manager のポリウム名。/dev/vx/dsk/volume_name の形式で表されます
 - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- **fs_options** フィールドには、次のいずれかを指定できます。
 - **ufs**: UFS ファイルシステムを示します。
 - **vxfs**: Veritas ファイルシステムを示します。
 - **swap**: スワップファイルシステムを示します。スワップマウントポイントはハイフン (-) で表します。
 - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、[85 ページの「RAID-1 ポリウム \(ミラー\) を持つブート環境の作成 \(コマンド行インタフェース\)」](#) を参照してください。

-n BE_name

作成するブート環境の名前。BE_name は、システム上で一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。[第 5 章](#) を参照してください。

例 4-7 別々のソースから単一のブート環境を作成(コマンド行インタフェース)

この例では、ソースブート環境 `third_disk` 内のルート (`/`) ファイルシステムに基づいてブート環境を作成します。`third_disk` はアクティブブート環境ではありません。「`mydescription`」という記述は、`second_disk` という名前の新しいブート環境に対応しています。

```
# lucreate -A 'mydescription' -s third_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -n second_disk
```

▼ Solaris フラッシュアーカイブ用の空のブート環境の作成(コマンド行インタフェース)

`lucreate` コマンドは、アクティブブート環境内のファイルシステムに基づいてブート環境を作成します。`lucreate` コマンドに `-s` オプションを指定して実行すると、空のブート環境を短時間で作成できます。スライスは、指定のファイルシステム用に予約されていますが、ファイルシステムはコピーされません。このブート環境は、名前が付けられてはいますが、実際には、Solaris フラッシュアーカイブがインストールされる時にはじめて作成されることとなります。空のブート環境にアーカイブがインストールされると、ファイルシステムは予約されたスライスにインストールされます。

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 空のブート環境を作成します。

```
# lucreate -A 'BE_name' -s - \
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A '*BE_description*'

(省略可能) ブート環境名 (`BE_name`) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-s -

空のブート環境を作成します。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を指定します。`-m` に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す `-` (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。

- ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
- Solaris ボリュームマネージャーのメタデバイス名。/dev/md/dsk/dnum の形式で表されます
- Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume_name の形式で表されます
- キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- fs_options フィールドには、次のいずれかを指定できます。
 - ufs: UFS ファイルシステムを示します。
 - vxfs: Veritas ファイルシステムを示します。
 - swap: スワップファイルシステムを示します。スワップマウントポイントはハイフン(-)で表します。
 - 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-n *BE_name*

作成するブート環境の名前。*BE_name* は、システム上で一意となるように指定する必要があります。

例 4-8 Solaris フラッシュアーカイブ用の空のブート環境の作成(コマンド行インタフェース)

この例では、ファイルシステムを一切含まないブート環境を作成します。

「mydescription」という記述は、second_disk という名前の新しいブート環境に対応しています。

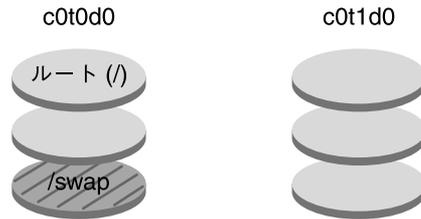
```
# lucreate -A 'mydescription' -s - \
-m /:/dev/dsk/c0t1d0s0:ufs -n second_disk
```

空のブート環境の作成が完了したら、フラッシュアーカイブをインストールし、アクティブ(ブート可能な状態)にすることができます。第 5 章を参照してください。

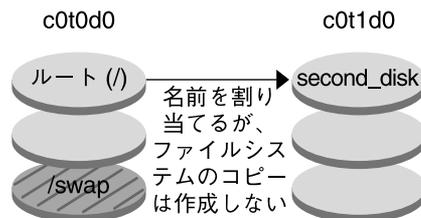
空のブート環境の作成とアーカイブのインストールの例については、172 ページの「空のブート環境を作成して Solaris フラッシュアーカイブをインストールする例(コマンド行インタフェース)」を参照してください。

次の図は、空のブート環境の作成の様子を示しています。

2つの物理ディスク
からなる元のシステム



空のブート環境の作成



```
コマンド: # lucreate
          -m /:/dev/dsk/c0t1d0s0:ufs \
          -n second_disk
```

▼ RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)

ブート環境を作成するとき、Solaris Live Upgrade は Solaris ボリュームマネージャテクノロジーを使って RAID-1 ボリュームを作成します。ブート環境を作成するとき、Solaris Live Upgrade を使って次の作業を行うことができます。

- 単一スライスの連結(サブミラー)を RAID-1 ボリューム(ミラー)から切り離します。必要な場合は、内容を保存して新しいブート環境の内容にすることができます。内容はコピーされないため、新しいブート環境を短時間で作成できます。ミラーから切り離されたサブミラーは、元のミラーの一部ではなくなります。サブミラーに対する読み取りや書き込みがミラーを介して実行されることはなくなります。
- ミラーを含んだブート環境を作成します。
- 新しく作成したミラーに単一スライスの連結を接続します。

Solaris Live Upgrade のミラー化機能を使用するには、状態データベースと状態データベースの複製を作成する必要があります。状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。

- 状態データベースの作成については、『Solaris ボリュームマネージャの管理』の第6章「状態データベース (概要)」を参照してください。
- Solaris ボリュームマネージャの概要と、Solaris Live Upgrade で実行できる作業については、27 ページの「RAID-1 ボリュームファイルシステムを持つブート環境の作成」を参照してください。
- Solaris Live Upgrade を使用するときには使用できない Solaris ボリュームマネージャの複雑な構成の詳細については、『Solaris ボリュームマネージャの管理』の第2章「記憶装置管理の概念」を参照してください。

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のように入力して新しいブート環境を作成します。

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options [-m...] \
-n BE_name
```

```
-A 'BE_description'
```

(省略可能)ブート環境名 (*BE_name*) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を `fstab` で指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
 - Solaris ボリュームマネージャのボリューム名。/dev/md/dsk/dnum の形式で表されます
 - Veritas Volume Manager のボリューム名。/dev/md/vxfs/dsk/dnum の形式で表されます
 - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- *fs_options* フィールドには、次のいずれかを指定できます。
 - `ufs`: UFS ファイルシステムを示します。
 - `vxfs`: Veritas ファイルシステムを示します。
 - `swap`: スワップファイルシステムを示します。スワップマウントポイントはハイフン (-) で表します。

- 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。
 - `mirror` を指定すると、指定したデバイスに RAID-1 ボリューム(ミラー)を作成できます。その後の `-m` オプションで `attach` を指定して、少なくとも1つの連結を新しいミラーに接続する必要があります。指定するデバイスには、正しく名前が付けられている必要があります。たとえば、論理デバイスの名前 `/dev/md/dsk/d10` をミラー名として使用できます。デバイスの命名の詳細は、『Solaris ボリュームマネージャの管理』の「Solaris ボリュームマネージャコンポーネントの概要」を参照してください。
 - `detach` を指定すると、指定したマウントポイントに関連付けられているボリュームから連結を切り離すことができます。ボリュームを指定する必要はありません。
 - `attach` を指定すると、指定したマウントポイントに関連付けられているミラーに連結を接続できます。指定した物理ディスクスライスは、単一デバイスの連結になり、ミラーに接続されます。ディスクに接続する連結を指定するには、デバイス名の後ろにコンマと連結の名前を付加します。コンマと連結の名前を省略して `lucreate` を実行すると、空いているボリュームが連結用に選択されます。

`lucreate` で作成できるのは、単一の物理スライスから成る連結だけです。このコマンドでは、1つのミラーに3つまで連結を接続できます。

- `preserve` を指定すると、既存のファイルシステムとその内容を保存できます。このキーワードを使うと、ソースブート環境の内容をコピーする処理を省略できます。内容を保存することで、新しいブート環境を短時間で作成できます。特定のマウントポイントについて、`preserve` で指定できるのは1つの物理デバイスだけです。`preserve` を指定して `lucreate` コマンドを実行すると、指定したファイルシステムに対してデバイスの内容が適切かどうかを検査されます。この検査は限定的なものなので、適合性を保証することはできません。

`preserve` キーワードは、物理スライスと Solaris ボリュームマネージャのボリュームの両方に使用できます。

- UFS ファイルシステムが物理スライスに置かれている場合に `preserve` キーワードを使用すると、UFS ファイルシステムの内容がそのスライスに保存されます。次の `-m` オプションの例では、`preserve` キーワードを使って、物理デバイス `c0t0d0s0` の内容を、ルート (`/`) ファイルシステムのマウントポイント用のファイルシステムとして保存します。

```
-m /:/dev/dsk/c0t0d0s0:preserve,ufs
```

- UFS ファイルシステムがボリュームに置かれている場合に `preserve` キーワードを使用すると、UFS ファイルシステムの内容がそのボリュームに保存されます。

次の `-m` オプションの例では、`preserve` キーワードを使って、RAID-1 ボリューム(ミラー) `d10` の内容を、ルート (`/`) ファイルシステムのマウントポイント用のファイルシステムとして保存します。

```
-m /:/dev/md/dsk/d10:preserve,ufs
```

次の `-m` オプションの例では、RAID-1 ボリューム(ミラー) `d10` がルート (`/`) ファイルシステムのマウントポイント用のファイルシステムとして構成されます。単一スライスの連結 `d20` が現在のミラーから切り離されます。`d20` がミラー `d10` に接続されます。ルート (`/`) ファイルシステムは、サブミラー `d20` に保持されます。

```
-m /:/dev/md/dsk/d10:mirror,ufs -m /:/dev/md/dsk/d20:detach,attach,preserve
```

```
-n BE_name
```

作成するブート環境の名前。`BE_name` は、システム上で一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-9 ミラーを持つブート環境の作成とデバイスの指定(コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「`mydescription`」という記述は、`another_disk` に対応しています。
- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。`d10` というミラーが作成されます。このミラー `d10` に、現在のブート環境のルート (`/`) ファイルシステムがコピーされます。ミラー `d10` にあるデータはすべて上書きされます。
- 2つのスライス `c0t0d0s0` および `c0t1d0s0` は、サブミラー `d1` および `d2` として指定されています。これら2つのサブミラーは、ミラー `d10` に追加されます。
- 新しいブート環境には `another_disk` という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:attach \  
-m /:/dev/dsk/c0t1c0s0,/dev/md/dsk/d2:attach -n another_disk
```

例 4-10 ミラーを持つブート環境の作成とサブミラー名の省略(コマンド行インタフェース)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「`mydescription`」という記述は、`another_disk` に対応しています。

- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。 `d10` というミラーが作成されます。このミラー `d10` に、現在のブート環境のルート (`/`) ファイルシステムがコピーされます。ミラー `d10` にあるデータはすべて上書きされます。
- 2つのスライス `c0t1d0s0` および `c0t2d0s0` は、サブミラーとして指定されています。サブミラーを指定せずに `lucreate` コマンドを実行すると、利用可能なボリューム名の一覧から名前が選択されます。これら2つのサブミラーは、ミラー `d10` に接続されます。
- 新しいブート環境には `another_disk` という名前が付けられます。

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:attach \
-m /:/dev/dsk/c0t1d0s0:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-11 ブート環境の作成とサブミラーの切り離し(コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「`mydescription`」という記述は、`another_disk` に対応しています。
- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。 `d10` というミラーが作成されます。
- スライス `c0t0d0s0` が現在のミラーから切り離されます。このスライスはサブミラー `d1` として指定され、ミラー `d10` に追加されます。このサブミラーの内容であるルート (`/`) ファイルシステムは保存され、コピー処理は発生しません。スライス `c0t1d0s0` はサブミラー `d2` として指定され、ミラー `d10` に追加されます。
- 新しいブート環境には `another_disk` という名前が付けられます。

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:detach,attach,preserve \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d2:attach -n another_disk
```

この例は、次の例のように簡略化できます。物理デバイスや論理デバイスの短縮名が使用されています。サブミラー `d1` および `d2` の指示子は省略されています。

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve \
-m /:/dev/dsk/c0t1d0s0:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-12 ブート環境の作成、サブミラーの切り離しと内容の保存(コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another_diskに対応しています。
- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。d20 というミラーが作成されます。
- スライス `c0t0d0s0` が現在のミラーから切り離され、ミラー d20 に追加されます。サブミラーの名前は指定されていません。このサブミラーの内容であるルート (`/`) ファイルシステムは保存され、コピー処理は発生しません。
- 新しいブート環境には another_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve \
-n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第 5 章を参照してください。

例 4-13 2つのミラーを持つブート環境の作成(コマンド行インタフェース)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another_diskに対応しています。
- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。このミラー d10 に、現在のブート環境のルート (`/`) ファイルシステムがコピーされます。ミラー d10 にあるデータはすべて上書きされます。
- 2つのスライス `c0t0d0s0` および `c0t1d0s0` は、サブミラー d1 および d2 として指定されています。これら2つのサブミラーは、ミラー d10 に追加されます。
- `lucreate` コマンドにより、`/opt` マウントポイントの UFS ファイルシステムが構成されます。d11 というミラーが作成されます。このミラー d11 に、現在のブート環境の `/opt` ファイルシステムがコピーされます。ミラー d11 にあるデータはすべて上書きされます。
- 2つのスライス `c2t0d0s1` および `c3t1d0s1` は、サブミラー d3 および d4 として指定されています。これら2つのサブミラーは、ミラー d11 に追加されます。
- 新しいブート環境には another_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:attach \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d2:attach \
-m /opt:/dev/md/dsk/d11:ufs,mirror \
-m /opt:/dev/dsk/c2t0d0s1,/dev/md/dsk/d3:attach \
-m /opt:/dev/dsk/c3t1d0s1,/dev/md/dsk/d4:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

▼ ブート環境の作成と内容のカスタマイズ(コマンド行インタフェース)

次のオプションを使って、新しいブート環境のファイルシステムの内容を変更できます。ディレクトリやファイルは新しいブート環境にコピーされません。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のように入力して新しいブート環境を作成します。

```
# lucreate -m mountpoint:device[,metadevice]:fs_options [-m ...] \
[-x exclude_dir] [-y include] \
[-Y include_list_file] \
[-f exclude_list_file]\
[-z filter_list] [-I] -n BE_name
```

-m mountpoint:device[,metadevice]:fs_options [-m ...]

新しいブート環境のファイルシステム構成を `vfstab` で指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成するファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
 - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます
 - Solaris ポリウムマネージャーのポリウム名。/dev/md/dsk/dnum の形式で表されます
 - Veritas Volume Manager のポリウム名。/dev/md/vxfs/dsk/dnum の形式で表されます
 - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します
- `fs_options` フィールドには、次のいずれかを指定できます。
 - `ufs`: UFS ファイルシステムを示します。
 - `vxfs`: Veritas ファイルシステムを示します。
 - `swap`: スワップファイルシステムを示します。スワップマウントポイントはハイフン (-) で表します。

- 論理デバイス(ミラー)であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、85 ページの「RAID-1 ボリューム(ミラー)を持つブート環境の作成(コマンド行インタフェース)」を参照してください。

-x *exclude_dir*

ファイルやディレクトリを除外して、新しいブート環境にコピーされないようにします。このオプションのインスタンスを複数使用して、複数のファイルまたはディレクトリを除外できます。

exclude_dir には、除外するディレクトリやファイルの名前を指定します。

-y *include_dir*

指定されたディレクトリやファイルを新しいブート環境にコピーします。ディレクトリは除外するが、個々のサブディレクトリやファイルを含める場合、このオプションを使用します。

include_dir には、含めるサブディレクトリやファイルの名前を指定します。

-Y *list_filename*

リスト内のディレクトリやファイルを新しいブート環境にコピーします。ディレクトリは除外するが、個々のサブディレクトリやファイルを含める場合、このオプションを使用します。

- *list_filename* は、リストを含むファイルのフルパスです。
- *list_filename* ファイルでは、1 ファイルを 1 行で表す必要があります。
- 行でディレクトリを指定すると、そのディレクトリの下にあるすべてのサブディレクトリとファイルが含まれます。行でファイルを指定すると、そのファイルだけが含まれます。

-f *list_filename*

リスト内のファイルやディレクトリを除外して、新しいブート環境にコピーされないようにします。

- *list_filename* は、リストを含むファイルのフルパスです。
- *list_filename* ファイルでは、1 ファイルを 1 行で表す必要があります。

-z *list_filename*

リスト内のディレクトリやファイルを新しいブート環境にコピーします。リスト内の各ファイルまたはディレクトリには、プラス「+」またはマイナス「-」記号を付けます。プラスはファイルやディレクトリを含めることを、マイナスはファイルやディレクトリを除外することを示します。

- *list_filename* は、リストを含むファイルのフルパスです。
- *list_filename* ファイルでは、1 ファイルを 1 行で表す必要があります。プラスまたはマイナスとファイル名との間には 1 つの半角スペースが必要です。

- 行でディレクトリとプラス (+) を指定すると、そのディレクトリの下にあるすべてのサブディレクトリとファイルがコピーされます。行でファイルとプラス (+) を指定すると、そのファイルだけがコピーされます。

-I

システムファイルの整合性チェックを無効にします。このオプションは慎重に使用してください。

重要なシステムファイルをブート環境から除外してしまうことを防ぐために、`lucreate` は整合性チェックを実行します。このチェックにより、システムパッケージデータベースに登録されたすべてのファイルが検査され、そのいずれかが除外されると、ブート環境の作成が停止します。このオプションを指定すると、この整合性チェックが無効になります。このオプションを指定すると、より短時間でブート環境を作成できますが、問題を検出できなくなる可能性があります。

-n *BE_name*

作成するブート環境の名前。*BE_name* は、システム上で一意となるように指定する必要があります。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第5章を参照してください。

例 4-14 ブート環境の作成とファイルの除外(コマンド行インタフェース)

この例では、新しいブート環境に `second_disk` という名前が付けられます。ソースブート環境には、ファイルシステムとしてルート (/) があります。新しいブート環境では、`/var` ファイルシステムがルート (/) ファイルシステムから分割され、別のスライスに置かれます。`lucreate` コマンドにより、ルート (/) マウントポイントと `/var` マウントポイントの UFS ファイルシステムが構成されます。また、2つの `/var` メールファイル `root` および `staff` は、新しいブート環境にコピーされません。ソースブート環境と新しいブート環境の間で自動的にスワップが共有されます。

```
# lucreate -n second_disk \
-m /:/dev/dsk/c0t1d0s0:ufs -m /var/mail:/dev/dsk/c0t2d0s0:ufs \
-x /var/mail/root -x /var/mail/staff
```

例 4-15 ブート環境を作成し、ファイルを除外または含める(コマンド行インタフェース)

この例では、新しいブート環境に `second_disk` という名前が付けられます。ソースブート環境には、OS用のファイルシステムとしてルート (/) があります。ソースブート環境には `/mystuff` というファイルシステムもあります。`lucreate` コマンドにより、ルート (/) マウントポイントと `/mystuff` マウントポイントの UFS ファイルシステムが構成されます。新しいブート環境には、`/mystuff` 内の2つのディレクトリだけがコピーされます。`/latest` および `/backup` です。ソースブート環境と新しいブート環境の間で自動的にスワップが共有されます。

```
# lucreate -n second_disk \  
-m /:/dev/dsk/c01t0d0s0:ufs -m /mystuff:/dev/dsk/c1t1d0s0:ufs \  
-x /mystuff -y /mystuff/latest -y /mystuff/backup
```

Solaris Live Upgrade によるアップグレード (作業)

この章では、Solaris Live Upgrade を使用して非アクティブブート環境のアップグレードとアクティブ化を行う方法について説明します。この章の内容は次のとおりです。

- 95 ページの「作業マップ: ブート環境のアップグレード」
- 96 ページの「ブート環境のアップグレード」
- 112 ページの「ブート環境への Solaris フラッシュアーカイブのインストール」
- 119 ページの「ブート環境のアクティブ化」

Solaris Live Upgrade は、メニューを介して使用することもコマンド行インタフェースで使用することもできます。次の説明では、両方のインタフェースについて手順を説明しています。これらの手順では、Solaris Live Upgrade の使用に関する詳しい説明は省略しています。コマンドの詳細については、第 10 章を参照してください。また、コマンド行インタフェースのオプションの詳細については、各マニュアルページを参照してください。

作業マップ: ブート環境のアップグレード

表 5-1 作業マップ: Solaris Live Upgrade を使ったアップグレード

| 作業 | 説明 | 参照先 |
|--|---|---|
| ブート環境をアップグレードするか Solaris フラッシュアーカイブをインストールします。 | <ul style="list-style-type: none"> ■ OS イメージを使用して非アクティブブート環境をアップグレードします。 ■ 非アクティブブート環境へ Solaris フラッシュアーカイブをインストールします。 | <ul style="list-style-type: none"> ■ 96 ページの「ブート環境のアップグレード」 ■ 112 ページの「ブート環境への Solaris フラッシュアーカイブのインストール」 |
| 非アクティブブート環境をアクティブにします。 | 変更を有効にし、非アクティブブート環境をアクティブに切り替えます。 | 119 ページの「ブート環境のアクティブ化」 |

表 5-1 作業マップ: Solaris Live Upgrade を使ったアップグレード (続き)

| 作業 | 説明 | 参照先 |
|-------------------------------------|-------------------------------|-------|
| (省略可能) アクティブ化で問題が発生した場合にフォールバックします。 | 問題が発生する場合は元のブート環境をアクティブに戻します。 | 第 6 章 |

ブート環境のアップグレード

「Upgrade」メニューまたは `luupgrade` コマンドを使用してブート環境をアップグレードします。この節では、次の場所に置かれているファイルを使用して非アクティブブート環境をアップグレードする手順について説明します。

- ネットワークファイルシステム (NFS) サーバー
- ローカルファイル
- ローカルテープ
- ローカルデバイス (DVD または CD)

アップグレードのガイドライン

最新の OS へアップグレードする間、アクティブブート環境への影響はありません。新しいファイルは非アクティブブート環境のクリティカルファイルシステムとマージされますが、共有可能ファイルシステムは変更されません。

アップグレードを行う代わりに、Solaris フラッシュアーカイブを作成して、非アクティブブート環境にアーカイブをインストールできます。それらの新しいファイルは非アクティブブート環境のクリティカルファイルシステムを上書きしますが、共有可能ファイルシステムは変更されません。112 ページの「ブート環境への Solaris フラッシュアーカイブのインストール」を参照してください。

物理ディスクスライス、Solaris ボリュームマネージャーのボリューム、および Veritas Volume Manager のボリュームがどのような組み合わせで含まれているブート環境もアップグレードできます。ルート (/) ファイルシステム用のスライスは、RAID-1 ボリューム (ミラー) に含まれている単一スライスの連結でなければいけません。ミラー化されたファイルシステムを持つブート環境の作成手順については、85 ページの「RAID-1 ボリューム (ミラー) を持つブート環境の作成 (コマンド行インタフェース)」を参照してください。

注-現在のシステム上に VxVM ボリュームが構成されている場合は、`lucreate` コマンドを使用して新しいブート環境を作成できます。新しいブート環境にデータをコピーすると、Veritas ファイルシステム構成が失われ、新しいブート環境に UFS ファイルシステムが作成されます。

パッケージまたはパッチによるシステムのアップグレード

Solaris Live Upgrade を使ってパッケージやパッチをシステムに追加できます。Solaris Live Upgrade では、現在動作しているシステムのコピーが作成されます。この新しいブート環境は、アップグレードすることも、パッケージやパッチを追加することもできます。Solaris Live Upgrade を使用すると、システムのダウンタイムはリブートの時間だけですみます。luupgrade コマンドを使って、パッチやパッケージを新しいブート環境に追加できます。



注意 - Solaris Live Upgrade でパッケージやパッチの追加または削除を行うには、パッケージやパッチが SVR4 パッケージガイドラインに準拠している必要があります。Sun のパッケージはこのガイドラインに準拠していますが、サードパーティーベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、パッケージ追加ソフトウェアの障害が発生するか、アップグレード中にアクティブブート環境が改変されてしまう可能性があります。

パッケージの要件については、[付録 B](#) を参照してください。

表 5-2 パッケージやパッチを使ったブート環境のアップグレード

| インストールの種類 | 説明 | 参照先 |
|-----------------|---|--|
| ブート環境へのパッチの追加 | 新しいブート環境を作成してから、-t オプションを指定して luupgrade コマンドを実行します。 | 103 ページの「ブート環境のオペレーティングシステムイメージにパッチを追加する (コマンド行インタフェース)」 |
| ブート環境へのパッケージの追加 | -p オプションを指定して luupgrade コマンドを実行します。 | 102 ページの「ブート環境のオペレーティングシステムイメージにパッケージを追加する (コマンド行インタフェース)」 |

▼ ブート環境のオペレーティングシステムイメージをアップグレードする (キャラクターインタフェース)

この手順でアップグレードを行う場合は、DVD または結合されたインストールイメージを使用する必要があります。CD でインストールする場合は、[100 ページの「複数の CD を使用してオペレーティングシステムイメージをアップグレードする \(コマンド行インタフェース\)」](#) の手順に従う必要があります。

注-この手順では、システムでボリュームマネージャーを実行していると仮定していません。ボリュームマネージャーを使用したリムーバブルメディアの管理の詳細は、『Solarisのシステム管理(デバイスとファイルシステム)』を参照してください。

- 1 **Solaris Live Upgrade** のメインメニューから「Upgrade」を選択します。
「Upgrade」メニューが表示されます。
- 2 新しいブート環境の名前を入力します。
- 3 **Solaris** インストールイメージが置かれている場所のパスを入力します。

| インストールメディアの種類 | 説明 |
|--------------------|---|
| ネットワークファイルシステム | インストールイメージが置かれているネットワークファイルシステムのパスを指定します。 |
| ローカルファイル | インストールイメージが置かれているローカルファイルシステムのパスを指定します。 |
| ローカルテープ | インストールイメージが置かれているローカルテープデバイスとテープ上の位置を指定します。 |
| ローカルデバイス、DVD、またはCD | ローカルデバイスと、インストールイメージのパスを指定します。 |

- SPARC: DVD または CD を使用する場合は、次の例のように、そのディスクへのパスを入力します。

```
/cdrom/cdrom0/s0/Solaris_10/s0
```

- 1つに結合されたイメージがネットワーク上に存在する場合は、次の例のようにそのネットワークファイルシステムのパスを入力します。

```
/net/installmachine/export/Solaris_10/os_image
```

- 4 **F3** を押してアップグレードします。
アップグレードが完了すると、メインメニューが表示されます。

▼ ブート環境のオペレーティングシステムイメージをアップグレードする(コマンド行インタフェース)

この手順でアップグレードを行う場合は、DVD または結合されたインストールイメージを使用する必要があります。複数の CD が必要なインストールでは、100 ページの「複数

の CD を使用してオペレーティングシステムイメージをアップグレードする(コマンド行インタフェース)」の手順に従ってください。

- 1 **Solaris Live Upgrade** の SUNWlur パッケージと SUNWluu パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「[pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法](#)」を参照してください。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 次のように入力して、アップグレードするブート環境とインストールソフトウェアのパスを指定します。

```
# luupgrade -u -n BE_name -s os_image_path
```

-u ブート環境のオペレーティングシステムイメージをアップグレードします。

-n BE_name アップグレード対象のブート環境の名前を指定します。

-s os_image_path オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。

例 5-1 DVD メディアからブート環境の OS イメージをアップグレードする(コマンド行インタフェース)

この例では、second_disk ブート環境は DVD メディアを使用してアップグレードされます。pkgadd コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -u -n second_disk -s /cdrom/cdrom0/s0
```

例 5-2 ネットワークインストールイメージからブート環境の OS イメージをアップグレードする(コマンド行インタフェース)

この例では、second_disk ブート環境をアップグレードします。pkgadd コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -u -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image
```

▼ 複数の CD を使用してオペレーティングシステムイメージをアップグレードする(コマンド行インタフェース)

オペレーティングシステムイメージは複数の CD 上にあるため、このアップグレード手順を使用する必要があります。ほかの CD がある場合は、`luupgrade` コマンドに `-i` オプションを指定してそれらの CD をインストールします。

- 1 **Solaris Live Upgrade** の `SUNWlur` パッケージと `SUNWluu` パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「[pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法](#)」を参照してください。

- 2 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 3 次のように入力して、アップグレードするブート環境とインストールソフトウェアのパスを指定します。

```
# luupgrade -u -n BE_name -s os_image_path
```

`-u` ブート環境のオペレーティングシステムイメージをアップグレードします。

`-n BE_name` アップグレード対象のブート環境の名前を指定します。

`-s os_image_path` オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。

- 4 1 枚目の CD のインストーラ処理が完了したら、2 枚目の CD を挿入します。

- 5 この手順は前述のものと同じですが、`-u` オプションではなく `-i` オプションを使用します。メニューまたはテキストモードで、2 枚目の CD 上のインストーラを実行するように選択してください。

- 次のコマンドは、メニューを使用して 2 枚目の CD 上のインストーラを実行します。

```
# luupgrade -i -n BE_name -s os_image_path
```

- 次のコマンドは、テキストモードで 2 枚目の CD 上のインストーラを実行します。ユーザーに入力を求めるプロンプトは表示されません。

```
# luupgrade -i -n BE_name -s os_image_path -0 '-nodisplay -noconsole'
```

| | |
|----------------------------|--|
| -i | 追加の CD がインストールされます。ソフトウェアは、指定されたメディア上のインストールプログラムを探し、そのプログラムを実行します。インストーラプログラムは、-s で指定します。 |
| -n <i>BE_name</i> | アップグレード対象のブート環境の名前を指定します。 |
| -s <i>os_image_path</i> | オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。 |
| -O '-nodisplay -noconsole' | (省略可能) テキストモードで 2 枚目の CD 上のインストーラを実行します。ユーザーに入力を求めるプロンプトは表示されません。 |

6 インストールする CD ごとに、手順 4 と手順 5 を繰り返します。

このようにして、ブート環境をアクティブにする準備が整います。119 ページの「ブート環境のアクティブ化」を参照してください。

例 5-3 SPARC: 複数の CD によりオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)

この例では、second_disk ブート環境をアップグレードします。インストールイメージは、2 枚の CD に入っています。Solaris SOFTWARE - 1 CD と Solaris SOFTWARE - 2 CD です。-u オプションでは、CD セットのすべてのパッケージを格納するだけの領域があるかどうかを判定します。-o オプションと -nodisplay および -noconsole オプションを指定すると、2 枚目の CD の読み取りの後にキャラクタユーザーインタフェースは表示されません。これらのオプションを使用すると、情報の入力を求められません。このインタフェースを表示する場合は、これらのオプションを省略します。

アップグレード先のリリースの Solaris Live Upgrade パッケージをインストールします。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
```

Solaris SOFTWARE - 1 CD を挿入し、次のように入力します。

- SPARC システムの場合:

```
# luupgrade -u -n second_disk -s /cdrom/cdrom0/s0
```

- x86 システムの場合:

```
# luupgrade -u -n second_disk -s /cdrom/cdrom0/
```

Solaris SOFTWARE - 2 CD を挿入し、次のように入力します。

```
# luupgrade -i -n second_disk -s /cdrom/cdrom0 -O '-nodisplay \
-noconsole'
```

Repeat this step for each CD that you need.

インストールする各 CD について、上記の手順を繰り返します。

▼ ブート環境のオペレーティングシステムイメージにパッケージを追加する(コマンド行インタフェース)

次の手順では、新しいブート環境に対してパッケージの削除と追加を行います。



注意 - Solaris Live Upgrade でアップグレードしたりパッケージやパッチの追加・削除を行なったりするには、パッケージやパッチが SVR4 パッケージガイドラインに準拠していなければなりません。Sun のパッケージはこのガイドラインに準拠していますが、サードパーティーベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、パッケージ追加ソフトウェアの障害が発生するか、アクティブブート環境が改変されてしまう可能性があります。

パッケージの要件については、[付録 B](#) を参照してください。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 新しいブート環境から 1 つのパッケージまたは一連のパッケージを削除するには、次のように入力します。

```
# luupgrade -P -n second_disk package-name
```

-P 指定したパッケージをブート環境から削除することを示します。

-n *BE_name* パッケージを削除するブート環境の名前を指定します。

package-name 削除するパッケージの名前を指定します。複数のパッケージ名を指定する場合は、スペースで区切ります。

- 3 新しいブート環境に 1 つのパッケージまたは一連のパッケージを追加するには、次のように入力します。

```
# luupgrade -p -n second_disk -s /path-to-packages package-name
```

-p ブート環境にパッケージを追加することを示します。

-n *BE_name* パッケージを追加するブート環境の名前を指定します。

-s *path-to-packages* 追加するパッケージが含まれているディレクトリへのパスを指定します。

package-name 追加するパッケージの名前を指定します。複数のパッケージ名を指定する場合は、スペースで区切ります。

例 5-4 ブート環境のオペレーティングシステムイメージにパッケージを追加する(コマンド行インタフェース)

この例では、second_disk ブート環境に対してパッケージの削除と追加を行います。

```
# luupgrade -P -n second_disk SUNWabc SUNWdef SUNWghi
# luupgrade -p -n second_disk -s /net/installmachine/export/packages \
SUNWijk SUNWlmn SUNWpkr
```

▼ ブート環境のオペレーティングシステムイメージにパッチを追加する(コマンド行インタフェース)

次の手順では、新しいブート環境に対してパッチの削除と追加を行います。



注意 - Solaris Live Upgrade でパッケージやパッチの追加または削除を行うには、パッケージやパッチが SVR4 パッケージガイドラインに準拠している必要があります。Sun のパッケージはこのガイドラインに準拠していますが、サードパーティーベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、パッケージ追加ソフトウェアの障害が発生するか、アクティブブート環境が変更されてしまう可能性があります。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 新しいブート環境から 1 つのパッチまたは一連のパッチを削除するには、次のように入力します。

```
# luupgrade -T -n second_disk patch_name
```

-T 指定したパッチをブート環境から削除することを示します。

-n *BE_name* パッチを削除するブート環境の名前を指定します。

patch-name 削除するパッチの名前を指定します。複数のパッチ名を指定する場合は、スペースで区切ります。
- 3 新しいブート環境に 1 つのパッチまたは一連のパッチを追加するには、次のコマンドを入力します。

```
# luupgrade -t -n second_disk -s /path-to-patches patch-name
```

-t ブート環境にパッチを追加することを示します。

-n *BE_name* パッチを追加するブート環境の名前を指定します。

| | |
|---------------------------------|--|
| <code>-s path-to-patches</code> | 追加するパッチが含まれているディレクトリへのパスを指定します。 |
| <code>patch-name</code> | 追加するパッチの名前を指定します。複数のパッチ名を指定する場合は、スペースで区切ります。 |

例 5-5 ブート環境のオペレーティングシステムイメージにパッチを追加する (コマンド行インタフェース)

この例では、`second_disk` ブート環境に対してパッチの削除と追加を行います。

```
# luupgrade -T -n second_disk 222222-01
# luupgrade -t -n second_disk -s /net/installmachine/export/packages \
333333-01 444444-01
```

▼ ブート環境にインストールされているパッケージの情報を取得する (コマンド行インタフェース)

次の手順では、新しいブート環境にインストールされているパッケージの整合性を確認します。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 新しいブート環境に新たにインストールされたパッケージの整合性を確認するには、次のように入力します。

```
# luupgrade -C -n second_disk -O "-v" package-name
```

| | |
|---------------------------|--|
| <code>-C</code> | 指定したパッケージに対して <code>pkgchk</code> コマンドを実行することを示します。 |
| <code>-n BE_name</code> | 検査を実行するブート環境の名前を指定します。 |
| <code>-O</code> | <code>pkgchk</code> コマンドにオプションを直接渡します。 |
| <code>package-name</code> | 検査するパッケージの名前を指定します。複数のパッケージ名を指定する場合は、スペースで区切ります。パッケージ名を省略すると、指定したブート環境にあるすべてのパッケージに対して検査が実行されます。 |
| <code>"-v"</code> | コマンドを冗長モードで実行することを指定します。 |

例 5-6 ブート環境のパッケージの整合性を確認する (コマンド行インタフェース)

この例では、パッケージ SUNWabc、SUNWdef、および SUNWghi を検査して、これらが正しくインストールされ損傷がないことを確認します。

```
# luupgrade -C -n second_disk SUNWabc SUNWdef SUNWghi
```

JumpStart プロファイルを使用したアップグレード

Solaris Live Upgrade で使用する JumpStart プロファイルを作成することができます。ユーザーがカスタム JumpStart プログラムに精通しているのであれば、これはカスタム JumpStart で使用するものと同じプロファイルです。次の手順により、luupgrade コマンドに -j オプションを指定して使用して、プロファイルの作成、プロファイルのテスト、およびインストールを行うことができます。



注意 - Solaris OS を Solaris フラッシュアーカイブとともにインストールする場合は、アーカイブおよびインストール用メディアに同一の OS バージョンが含まれている必要があります。たとえば、アーカイブが Solaris 10 オペレーティングシステムで、DVD メディアを使用している場合は、Solaris 10 DVD メディアを使用してアーカイブをインストールする必要があります。OS バージョンが一致しないと、ターゲットシステムへのインストールは失敗します。次のキーワードまたはコマンドを使用する場合は、同一のオペレーティングシステムが必要です。

- プロファイルの archive_location キーワード
- -s、-a、-j、および -J オプションを指定した luupgrade コマンド

詳細については、以下を参照してください。

- 105 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」
- 110 ページの「Solaris Live Upgrade で使用されるプロファイルをテストする」
- 111 ページの「プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行インタフェース)」
- JumpStart プロファイルの作成については、『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の「プロファイルの作成」を参照してください。

▼ Solaris Live Upgrade で使用されるプロファイルを作成する

ここでは、Solaris Live Upgrade で使用されるプロファイルの作成方法について説明します。luupgrade コマンドに -j オプションを指定してこのプロファイルを使用し、非アクティブブート環境をアップグレードできます。

このプロファイルの使用方法については、次の2つの節を参照してください。

- プロファイルを使用したアップグレードについては、111 ページの「プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行インタフェース)」を参照してください。
- プロファイルを使用した Solaris フラッシュのインストールについては、116 ページの「プロファイルを使用した Solaris フラッシュアーカイブのインストール (コマンド行インタフェース)」を参照してください。

1 テキストエディタを使用してテキストファイルを作成します。

ファイルにわかりやすい名前を付けます。プロファイルの名前は、システムに Solaris ソフトウェアをインストールするためにそのプロファイルをどのように使用するかを示すものにしてください。たとえば、このプロファイルに `upgrade_Solaris_10` という名前を付けます。

2 プロファイルにプロファイルキーワードと値を追加します。

Solaris Live Upgrade プロファイルで使用できるのは、次の表に示されたアップグレードキーワードだけです。

次の表に、`Install_type` キーワードの値 `upgrade` または `flash_install` とともに使用できるキーワードを示します。

| 初期アーカイブ作成のキーワード | 説明 | 参照 |
|---|--|--|
| (必須) <code>Install_type</code> | システムの既存の Solaris 環境をアップグレードするか、システムに Solaris フラッシュアーカイブをインストールするかを定義します。このキーワードには、次の値を使用します。 <ul style="list-style-type: none"> ■ アップグレードの場合は、<code>upgrade</code> ■ Solaris フラッシュインストールの場合は、<code>flash_install</code> ■ Solaris フラッシュ差分インストールの場合は、<code>flash_update</code> | このキーワードで使用可能なすべての値の説明は、『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の「 <code>install_type</code> プロファイルキーワード」を参照してください。 |
| (Solaris フラッシュアーカイブの場合は必須)
<code>archive_location</code> | 指定された位置から Solaris フラッシュアーカイブを取得します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の「 <code>archive_location</code> プロファイルキーワード」を参照してください。 |
| (省略可能) <code>cluster</code> (クラスタの追加または削除) | システムにインストールされるソフトウェアグループに対してクラスタを追加するか、または削除するかを指定します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の「 <code>cluster</code> プロファイルキーワード (ソフトウェアグループの追加)」を参照してください。 |

| 初期アーカイブ作成のキーワード | 説明 | 参照 |
|----------------------------|--|---|
| (省略可能) geo | システムにインストールする地域ロケールか、あるいはシステムのアップグレード時に追加する地域ロケールを指定します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「geo プロファイルキーワード」を参照してください。 |
| (省略可能) local_customization | クローンシステムに Solaris フラッシュアーカイブをインストールする前に、カスタムスクリプトを作成して、クローンシステム上のローカル構成を保存できます。local_customization キーワードは、これらのスクリプトの格納先ディレクトリを示します。この値は、クローンシステム上のスクリプトへのパスです。 | 配置前および配置後スクリプトについては、『Solaris 10 11/06 インストールガイド(Solaris フラッシュアーカイブの作成とインストール)』の「カスタムスクリプトの作成」を参照してください。 |
| (省略可能) locale | インストールまたはアップグレード時に追加するロケールパッケージを指定します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「locale プロファイルキーワード」を参照してください。 |
| (省略可能) package | システムにインストールされるソフトウェアグループに対してパッケージを追加するか、または削除するかを指定します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「package プロファイルキーワード」を参照してください。 |

次の表に、Install_type キーワードの値 flash_update とともに使用できるキーワードを示します。

| 差分アーカイブ作成のキーワード | 説明 | 参照 |
|-----------------------|---|--|
| (必須) Install_type | システムに Solaris フラッシュアーカイブをインストールすることを指定します。差分アーカイブを表す値は flash_update です。 | このキーワードで使用可能なすべての値の説明は、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「install_type プロファイルキーワード」を参照してください。 |
| (必須) archive_location | 指定された位置から Solaris フラッシュアーカイブを取得します。 | このキーワードで使用可能な値のリストについては、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「archive_location プロファイルキーワード」を参照してください。 |

| 差分アーカイブ作成のキーワード | 説明 | 参照 |
|-------------------------------|---|---|
| (省略可能)
forced_deployment | ソフトウェアで想定されているものとは異なるクローンシステムに、Solaris フラッシュ差分アーカイブを強制的にインストールします。
forced_deployment を使用すると、クローンシステムを期待される状態にするために、新規ファイルがすべて削除されます。ファイルを削除して良いかどうか判断できない場合には、デフォルトを使用してください。デフォルトでは、新規ファイルが削除されそうになると、インストールが停止します。 | このキーワードの詳細は、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「forced_deployment プロファイルキーワード (Solaris フラッシュ差分アーカイブのインストール)」を参照してください。 |
| (省略可能)
local_customization | クローンシステムに Solaris フラッシュアーカイブをインストールする前に、カスタムスクリプトを作成して、クローンシステム上のローカル構成を保存できます。local_customization キーワードは、これらのスクリプトの格納先ディレクトリを示します。この値は、クローンシステム上のスクリプトへのパスです。 | 配置前および配置後スクリプトについては、『Solaris 10 11/06 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の「カスタムスクリプトの作成」を参照してください。 |
| (省略可能)
no_content_check | Solaris フラッシュ差分アーカイブを使用してクローンシステムをインストールする場合、no_content_check キーワードを使用してファイルごとの検証を省略できます。ファイルごとの検証により、クローンシステムがマスターシステムの複製であることが保証されます。クローンシステムが元のマスターシステムの複製であることが確実にある場合を除き、このキーワードの使用は避けてください。 | このキーワードの詳細は、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「no_content_check プロファイルキーワード (Solaris フラッシュアーカイブのインストール)」を参照してください。 |
| (省略可能) no_master_check | Solaris フラッシュ差分アーカイブを使用してクローンシステムをインストールする場合、no_master_check キーワードを使用してファイルの検証を省略できます。クローンシステムのファイルの検証は行われません。この検証により、クローンシステムが元のマスターシステムから構築されていることが保証されます。クローンシステムが元のマスターシステムの複製であることが確実にある場合を除き、このキーワードの使用は避けてください。 | このキーワードの詳細は、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「no_master_check プロファイルキーワード (Solaris フラッシュアーカイブのインストール)」を参照してください。 |

- 3 プロファイルをローカルシステムのディレクトリに保存します。
- 4 プロファイルの所有者が root で、そのアクセス権が 644 に設定されていることを確認します。
- 5 (省略可能) プロファイルをテストします。
プロファイルのテスト方法については、110 ページの「Solaris Live Upgrade で使用されるプロファイルをテストする」を参照してください。

例 5-7 Solaris Live Upgrade プロファイルの作成

この例では、プロファイルはアップグレードパラメータを提供します。このプロファイルでは、Solaris Live Upgrade の `luupgrade` コマンドに `-u` および `-j` オプションを指定して、非アクティブなブート環境をアップグレードします。このプロファイルでは、パッケージおよびクラスタを追加します。地域ロケールおよび追加ロケールも、プロファイルに追加されます。プロファイルにロケールを追加する場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。

```
# profile keywords      profile values
# -----
install_type           upgrade
package                SUNWxman add
cluster                SUNWcacc add
geo                    C_Europe
locale                 zh_TW
locale                 zh_TW.BIG5
locale                 zh_TW.UTF-8
locale                 zh_HK.UTF-8
locale                 zh_HK.BIG5HK
locale                 zh
locale                 zh_CN.GB18030
locale                 zh_CN.GBK
locale                 zh_CN.UTF-8
```

例 5-8 差分アーカイブをインストールするための Solaris Live Upgrade プロファイルの作成

次のプロファイルの例は、Solaris Live Upgrade でクローンシステムに差分アーカイブをインストールするためのものです。差分アーカイブで指定されているファイルだけが、追加、削除、または変更されます。Solaris フラッシュアーカイブは、NFS サーバーから取得されます。イメージは元のマスターシステムから構築されたものなので、クローンシステムのイメージの妥当性検査は行われません。このプロファイルは、Solaris Live Upgrade の `luupgrade` コマンドの `-u` オプションと `-j` オプションで使用します。

```
# profile keywords      profile values
# -----
install_type           flash_update
archive_location       nfs installserver:/export/solaris/archive/solarisarchive
no_master_check
```

`luupgrade` コマンドによる差分アーカイブのインストールについては、116 ページの「プロファイルを使用した Solaris フラッシュアーカイブのインストール(コマンド行インタフェース)」を参照してください。

▼ Solaris Live Upgrade で使用されるプロファイル进行测试する

プロファイルを作成したら、`luupgrade` コマンドを使用してプロファイル进行测试します。`luupgrade` が生成するインストール出力を調べることによって、意図したとおりにプロファイルが動作するかをすばやく調べることができます。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 プロファイル进行测试する

```
# luupgrade -u -n BE_name -D -s os_image_path -j profile_path
```

| | |
|-------------------------------|--|
| <code>-u</code> | ブート環境のオペレーティングシステムイメージをアップグレードします。 |
| <code>-n BE_name</code> | アップグレード対象のブート環境の名前を指定します。 |
| <code>-D</code> | <code>luupgrade</code> コマンドは、選択されたブート環境のディスク構成を使用して、 <code>-j</code> オプションと共に渡されたプロファイルオプション进行测试します。 |
| <code>-s os_image_path</code> | オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。 |
| <code>-j profile_path</code> | アップグレード用に構成されたプロファイルのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。 |

例 5-9 Solaris Live Upgrade を使用したプロファイルのテスト

次の例では、プロファイル名は `Flash_profile` です。このプロファイルは、`second_disk` という名前の非アクティブブート環境でのテストに成功します。

```
# luupgrade -u -n ulb08 -D -s /net/installsvr/export/u1/combined.ulwos \
-j /var/tmp/flash_profile
Validating the contents of the media /net/installsvr/export/u1/combined.ulwos.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains Solaris version 10.
Locating upgrade profile template to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE second_disk.
Determining packages to install or upgrade for BE second_disk.
Simulating the operating system upgrade of the BE second_disk.
The operating system upgrade simulation is complete.
```

INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of the upgrade operation.

INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of cleanup operations required.

The Solaris upgrade of the boot environment second_disk is complete.

これで、プロファイルを使用して非アクティブブート環境をアップグレードできます。

▼ プロファイルを使用して **Solaris Live Upgrade** でアップグレードする (コマンド行インタフェース)

ここでは、プロファイルを使用した OS のアップグレード方法を、手順を追って説明します。

プロファイルを使用して Solaris フラッシュアーカイブをインストールする場合は、116 ページの「[プロファイルを使用した Solaris フラッシュアーカイブのインストール \(コマンド行インタフェース\)](#)」を参照してください。

プロファイルにロケールを追加した場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。



注意 - Solaris OS を Solaris フラッシュアーカイブとともにインストールする場合は、アーカイブおよびインストール用メディアに同一の OS バージョンが含まれている必要があります。たとえば、アーカイブが Solaris 10 オペレーティングシステムで、DVD メディアを使用している場合は、Solaris 10 DVD メディアを使用してアーカイブをインストールする必要があります。OS バージョンが一致しないと、ターゲットシステムへのインストールは失敗します。次のキーワードまたはコマンドを使用する場合は、同一のオペレーティングシステムが必要です。

- プロファイルの `archive_location` キーワード
- `-s`、`-a`、`-j`、および `-J` オプションを指定した `luupgrade` コマンド

- 1 **Solaris Live Upgrade** の `SUNWlur` パッケージと `SUNWluu` パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「[pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法](#)」を参照してください。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 プロファイルを作成します。
Solaris Live Upgrade プロファイルで使用可能なアップグレードキーワードのリストについては、105 ページの「[Solaris Live Upgrade で使用されるプロファイルを作成する](#)」を参照してください。

4 次のコマンドを入力します。

```
# luupgrade -u -n BE_name -s os_image_path -j profile_path
```

- u ブート環境のオペレーティングシステムイメージをアップグレードします。
- n *BE_name* アップグレード対象のブート環境の名前を指定します。
- s *os_image_path* オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。
- j *profile_path* プロファイルへのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。プロファイル作成についての詳細は、[105 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」](#)を参照してください。

このようにして、ブート環境をアクティブにする準備が整います。

例 5-10 カスタム JumpStart プロファイルを使用したブート環境のアップグレード (コマンド行インタフェース)

この例では、`second_disk` ブート環境はプロファイルを使用してアップグレードされます。-j オプションを使用して、プロファイルにアクセスします。このようにして、ブート環境をアクティブにする準備が整います。プロファイルを作成する場合は、[105 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」](#)を参照してください。`pkgadd` コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

ブート環境への Solaris フラッシュアーカイブのインストール

この節では、Solaris Live Upgrade を使用して Solaris フラッシュアーカイブをインストールする手順を説明します。Solaris フラッシュアーカイブをインストールすると、新しいブート環境上に存在する共有ファイル以外のすべてのファイルが上書きされます。アーカイブは、次のメディアに格納されています。

- HTTP サーバー
- FTP サーバー - コマンド行インタフェースでのみ使用可能

- ネットワークファイルシステム (NFS) サーバー
- ローカルファイル
- ローカルテープ
- ローカルデバイス (DVD または CD)

Solaris フラッシュアーカイブをインストールおよび作成する場合は、次の点に注意してください。

| 説明 | 例 |
|--|---|
| <p>注意 - Solaris OS を Solaris フラッシュアーカイブとともにインストールする場合は、アーカイブおよびインストール用メディアに同一の OS バージョンが含まれている必要があります。OS バージョンが一致しないと、ターゲットシステムへのインストールは失敗します。次のキーワードまたはコマンドを使用する場合は、同一のオペレーティングシステムが必要です。</p> <ul style="list-style-type: none"> ■ プロファイルの <code>archive_location</code> キーワード ■ <code>-s</code>、<code>-a</code>、<code>-j</code>、および <code>-J</code> オプションを指定した <code>luupgrade</code> コマンド | <p>たとえば、アーカイブが Solaris 10 オペレーティングシステムで、DVD メディアを使用している場合は、Solaris 10 DVD メディアを使用してアーカイブをインストールする必要があります。</p> |
| <p>注意 - 非大域ゾーンがインストールされていると、Solaris フラッシュアーカイブは正常に作成されません。Solaris フラッシュ機能は、Solaris ゾーン機能とは互換性がありません。非大域ゾーンに Solaris フラッシュアーカイブを作成するか、非大域ゾーンがインストールされている大域ゾーンにアーカイブを作成すると、作成されるアーカイブは、アーカイブが配置されるときに適切にインストールされません。</p> | |
| 説明 | 参照先 |
| <p>アーカイブ記憶域と関連付けられたパスの正しい構文の例。</p> | <p>『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の「<code>archive_location</code> プロファイルキーワード」を参照してください。</p> |
| <p>Solaris フラッシュのインストール機能を使用するには、マスターシステムをインストールし、Solaris フラッシュアーカイブを作成します。</p> | <p>アーカイブの作成についての詳細は、『Solaris 10 11/06 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第 3 章「Solaris フラッシュアーカイブの作成 (作業)」を参照してください。</p> |

▼ ブート環境への **Solaris** フラッシュアーカイブのインストール(キャラクターインターフェイス)

- 1 **Solaris Live Upgrade** の SUNWlur パッケージと SUNWluu パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「[pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法](#)」を参照してください。

- 2 **Solaris Live Upgrade** のメインメニューから「Flash」を選択します。

「Flash an Inactive Boot Environment」メニューが表示されます。

- 3 **Solaris** フラッシュアーカイブをインストールするブート環境の名前と、インストールメディアの場所を入力します。

Name of Boot Environment: **Solaris_10**

Package media: **/net/install-svr/export/Solaris_10/latest**

- 4 アーカイブを追加するために **F1** を押します。

「Archive Selection」サブメニューが表示されます。

Location - Retrieval Method
<No Archives added> - Select ADD to add archives

このメニューでは、アーカイブのリストを作成できます。アーカイブの追加または削除を行うには、次の手順を実行します。

- a. アーカイブをリストに追加する場合は **F1** を押します。

「Select Retrieval Method」サブメニューが表示されます。

HTTP
NFS
Local File
Local Tape
Local Device

- b. 「**Select Retrieval Method**」メニューで、**Solaris** フラッシュアーカイブの場所を選択します。

| 選択された媒体 | プロンプト |
|---------|--|
| HTTP | Solaris フラッシュアーカイブのアクセスに必要な URL 情報とプロキシ情報を指定します。 |
| NFS | Solaris フラッシュアーカイブが置かれているネットワークファイルシステムのパスを指定します。アーカイブのファイル名も指定できます。 |

| 選択された媒体 | プロンプト |
|----------|---|
| ローカルファイル | Solaris フラッシュアーカイブが置かれているローカルファイルシステムのパスを指定します。 |
| ローカルテープ | Solaris フラッシュアーカイブが置かれているローカルテープドライブとテープ上の位置を指定します。 |
| ローカルデバイス | Solaris フラッシュアーカイブが置かれているローカルデバイス、パス、ファイルシステムの種類を指定します。 |

次のような「Retrieval」サブメニューが表示されます。表示は選択されたメディアによって異なります。

NFS Location:

- c. 次の例のように、アーカイブのパスを入力します。

NFS Location: **host:/path/to archive.flar**

- d. **F3** を押してリストにアーカイブを追加します。
- e. (省略可能) アーカイブをリストから削除する場合は **F2** を押します。
- f. インストールするアーカイブがリストに含まれた時点で **F6** を押して終了します。

- 5 **F3** を押して1つまたは複数のアーカイブをインストールします。

Solaris フラッシュアーカイブがブート環境にインストールされます。ブート環境上のファイルは、共有可能ファイルを除きすべて上書きされます。

このようにして、ブート環境をアクティブにする準備が整います。121 ページの「ブート環境のアクティブ化(キャラクターインターフェイス)」を参照してください。

▼ ブート環境へ Solaris フラッシュアーカイブをインストールする(コマンド行インターフェイス)

- 1 **Solaris Live Upgrade** の **SUNWlur** パッケージと **SUNWluu** パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「**pkgadd** コマンドを使用して **Solaris Live Upgrade** をインストールする方法」を参照してください。
- 2 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

3 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -a archive
```

| | |
|------------------|--|
| -f | オペレーティングシステムを Solaris フラッシュアーカイブからインストールすることを示します。 |
| -n BE_name | アーカイブを使用してインストールするブート環境の名前を指定します。 |
| -s os_image_path | オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。 |
| -a archive | Solaris フラッシュアーカイブへのパス (ローカルファイルシステムでそのアーカイブが利用できる場合)。-s オプションと -a オプションを使用して指定するオペレーティングシステムイメージのバージョンは、同じでなければなりません。 |

例 5-11 ブート環境へ Solaris フラッシュアーカイブをインストールする (コマンド行インタフェース)

この例では、アーカイブは `second_disk` ブート環境にインストールされます。アーカイブはローカルシステムに存在します。-s および -a オプションで指定するオペレーティングシステムのバージョンは、どちらも Solaris 10 11/06 リリースです。second_disk 上のファイルは、共有可能ファイルを除いてすべて上書きされます。pkgadd コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/10
```

このようにして、ブート環境をアクティブにする準備が整います。

▼ プロファイルを使用した **Solaris** フラッシュアーカイブのインストール (コマンド行インタフェース)

ここでは、プロファイルを使用して Solaris フラッシュアーカイブまたは差分アーカイブをインストールする手順を説明します。

プロファイルにロケールを追加した場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。

- 1 **Solaris Live Upgrade** の `SUNWlur` パッケージと `SUNWluu` パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「[pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法](#)」を参照してください。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 プロファイルを作成します。
Solaris Live Upgrade プロファイルで使用可能なキーワードのリストについては、105 ページの「[Solaris Live Upgrade で使用されるプロファイルを作成する](#)」を参照してください。
- 4 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -j profile_path
```

| | |
|-------------------------------|--|
| <code>-f</code> | オペレーティングシステムを Solaris フラッシュアーカイブからインストールすることを示します。 |
| <code>-n BE_name</code> | アップグレード対象のブート環境の名前を指定します。 |
| <code>-s os_image_path</code> | オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。 |
| <code>-j profile_path</code> | フラッシュインストール用に構成された JumpStart プロファイルへのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。 <code>-s</code> オプションのオペレーティングシステムのバージョンと Solaris フラッシュアーカイブのオペレーティングシステムのバージョンは、同一でなければいけません。 |

このようにして、ブート環境をアクティブにする準備が整います。

例 5-12 プロファイルを使用してブート環境に Solaris フラッシュアーカイブをインストールする (コマンド行インタフェース)

この例では、インストールするアーカイブの場所をプロファイルで指定しています。

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       nfs installserver:/export/solaris/flasharchive/solarisarchive
```

プロファイルを作成した後、`luupgrade` コマンドを実行してアーカイブをインストールできます。`-j` オプションを使用して、プロファイルにアクセスします。`pkgadd` コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

このようにして、ブート環境をアクティブにする準備が整います。プロファイルを作成する場合は、105 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」を参照してください。

▼ プロファイルキーワードを使用した Solaris フラッシュアーカイブのインストール(コマンド行インタフェース)

ここでは、プロファイルファイルではなく、`archive_location` キーワードをコマンド行で使用して、Solaris フラッシュアーカイブをインストールする手順を説明します。プロファイルファイルを使用せずに、アーカイブを取得できます。

- 1 **Solaris Live Upgrade** の `SUNWlur` パッケージと `SUNWluu` パッケージをシステムにインストールします。これらのパッケージは、アップグレード先のリリースのものである必要があります。詳細な手順については、61 ページの「`pkgadd` コマンドを使用して **Solaris Live Upgrade** をインストールする方法」を参照してください。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -J 'archive_location path-to-profile'
```

| | |
|-------------------------------|--|
| <code>-f</code> | オペレーティングシステムを Solaris フラッシュアーカイブからアップグレードすることを指定します。 |
| <code>-n BE_name</code> | アップグレード対象のブート環境の名前を指定します。 |
| <code>-s os_image_path</code> | オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。 |

`-J 'archive_location path-to-profile'` `archive_location` プロファイルキーワードと、JumpStart プロファイルへのパスを指定します。-s オプションのオペレーティングシステムのバージョンと Solaris フラッシュアーカイブのオペレーティングシステムのバージョンは、同一でなければいけません。キーワードの値については、『Solaris 10 11/06 インストールガイド(カスタム JumpStart/ 上級編)』の「`archive_location` プロファイルキーワード」を参照してください。

このようにして、ブート環境をアクティブにする準備が整います。

例 5-13 プロファイルキーワードを使用した Solaris フラッシュアーカイブのインストール(コマンド行インタフェース)

この例では、アーカイブは `second_disk` ブート環境にインストールされます。-J オプションと `archive_location` キーワードを使用して、アーカイブを取得します。`second_disk` 上のファイルは、共有可能ファイルを除いてすべて上書きされます。`pkgadd` コマンドは、アップグレード先のリリースの Solaris Live Upgrade パッケージを追加します。

```
# pkgadd -d /server/packages SUNWlur SUNWluu
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-J 'archive_location http://example.com/myflash.flar'
```

ブート環境のアクティブ化

ブート環境をアクティブにすると、次のシステムリブートでブート可能になります。新しいアクティブブート環境で何か問題が発生する場合は、元のブート環境にすぐに戻すことができます。第 6 章を参照してください。

| 説明 | 参照先 |
|---|---|
| キャラクタユーザーインタフェース (CUI) を使用してブート環境をアクティブにするには、この手順を使用します。 | 121 ページの「ブート環境のアクティブ化 (キャラクタユーザーインタフェース)」 |
| 注-ブート環境をはじめてアクティブ化する場合は、「Activate」メニューまたは <code>luactivate</code> コマンドを使用する必要があります。 | |

| 説明 | 参照先 |
|---|--|
| <p>luactivate コマンドを使用してブート環境をアクティブにするには、この手順を使用します。</p> <p>注-ブート環境をはじめてアクティブ化する場合は、「Activate」メニューまたは luactivate コマンドを使用する必要があります。</p> | <p>123 ページの「ブート環境をアクティブにする (コマンド行インタフェース)」</p> |
| <p>ブート環境をアクティブにして強制的にファイルの同期を行うには、この手順を使用します。</p> <p>注-最初のアクティブ化の際にファイルが同期されます。最初のアクティブ化のあとでブート環境を切り替える場合、ファイルは同期されません。</p> | <p>124 ページの「ブート環境をアクティブにしてファイルを同期させる (コマンド行インタフェース)」</p> |
| <p>x86: GRUB メニューを使用してブート環境をアクティブにするには、この手順を使用します。</p> <p>注-GRUB メニューを使用すると、ブート環境を簡単に切り替えることができます。最初のアクティブ化のあと、ブート環境が GRUB メニューに表示されます。</p> | <p>126 ページの「x86: GRUB メニューを使ってブート環境をアクティブ化する (コマンド行インタフェース)」</p> |

ブート環境をアクティブ化するための要件と制限

ブート環境を正常にアクティブにするためには、そのブート環境が次の条件を満たしている必要があります。

| 説明 | 参照先 |
|---|--|
| <p>ブート環境のステータスは「complete」でなければなりません。</p> | <p>ステータスを確認するには、140 ページの「すべてのブート環境のステータスの表示」を参照してください。</p> |
| <p>現在のブート環境とは別のブート環境をアクティブにする場合は、lumount または mount を使用してそのブート環境のパーティションをマウントすることはできません。</p> | <p>lumount (1M) または mount (1M) のマニュアルページを参照してください。</p> |
| <p>比較処理で使用中のブート環境はアクティブにできません。</p> | <p>手順については、145 ページの「ブート環境の比較」を参照してください。</p> |

| 説明 | 参照先 |
|---|--|
| スワップを再構成する場合は、非アクティブブート環境をブートする前に実行してください。デフォルトでは、すべてのブート環境が同じスワップデバイスを共有します。 | スワップを再構成する場合は、次のいずれかの手順を参照してください。
<ul style="list-style-type: none"> ■ 76ページの「ブート環境を作成しスワップを再構成する(コマンド行インタフェース)」 ■ 「ブート環境を作成する(キャラクターインターフェイス)」の手順9 |

x86のみ-x86 システムの場合は、GRUB メニューを使用してアクティブ化することもできます。ただし、次のような例外があります。

- ブート環境が **Solaris 8、9**、または **10 3/05** リリースで作成されている場合、このブート環境は、必ず `luactivate` コマンドまたは「Activate」メニューを使ってアクティブ化する必要があります。これらの古いブート環境は、GRUB メニューには表示されません。
- ブート環境をはじめてアクティブ化する場合は、`luactivate` コマンドまたは「Activate」メニューを使用する必要があります。次のブート時には、そのブート環境の名前が GRUB メインメニューに表示されます。それにより、GRUB メニューで適切なエントリを選択して、このブート環境に切り替えることができます。

125 ページの「x86: GRUB メニューを使ったブート環境のアクティブ化」を参照してください。

▼ ブート環境のアクティブ化(キャラクターインタフェース)

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。ここでいう「同期」とは、いくつかのクリティカルなシステムファイルやディレクトリを、以前にアクティブだったブート環境から、ブート中のブート環境へコピーすることです。Solaris Live Upgrade では、強制的に同期を行うようにプロンプトで指定しない限り、2回目以降のブート時には同期は行われません。

同期の詳細については、53 ページの「ブート環境間でのファイルの同期」を参照してください。

x86のみ-x86 システムの場合は、GRUB メニューを使用してアクティブ化することもできます。ただし、次のような例外があります。

- ブート環境が **Solaris 8、9**、または **10 3/05** リリースで作成されている場合、このブート環境は、必ず `luactivate` コマンドまたは「Activate」メニューを使ってアクティブ化する必要があります。これらの古いブート環境は、GRUB メニューには表示されません。
- ブート環境をはじめてアクティブ化する場合は、`luactivate` コマンドまたは「Activate」メニューを使用する必要があります。次回のブート時には、そのブート環境の名前が GRUB メインメニューに表示されます。それにより、GRUB メニューで適切なエントリを選択して、このブート環境に切り替えることができます。

125 ページの「[x86:GRUB メニューを使ったブート環境のアクティブ化](#)」を参照してください。

1 **Solaris Live Upgrade** のメインメニューで「Activate」を選択します。

2 アクティブにするブート環境の名前を入力します。

```
Name of Boot Environment: Solaris_10  
Do you want to force a Live Upgrade sync operations: no
```

3 ファイルの同期を実行することも、実行せずに処理を続けることもできます。

- Return キーを押して継続します。
ブート環境の最初のブートでは、ファイルの同期が自動的に行われます。
- ファイルの同期を実行できますが、この機能は慎重に使用してください。同期するファイルに、それぞれのブート環境のオペレーティングシステムが対応していなければなりません。ファイルの同期を実行する場合は、次のように入力します。

```
Do you want to force a Live Upgrade sync operations: yes
```



注意-以前のアクティブブート環境で発生した変更によりユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、同期を使用する際には十分注意してください。たとえば、現在のブート環境で Solaris 10 11/06 ソフトウェアを実行していて、強制的な同期処理を行なったあとで、Solaris 9 リリースにブート環境を戻したとします。この場合、Solaris 9 リリースでファイルが変更されることがあります。ファイルは OS のリリースに依存しているため、Solaris 9 リリースのブートは失敗する可能性があります。Solaris 10 11/06 のファイルと Solaris 9 のファイルは互換性があるとは限らないからです。

4 **F3** を押して、アクティブ化の処理を開始します。

5 Return キーを押して継続します。

新しいブート環境は、次のリブート時にアクティブになります。

- 6 非アクティブブート環境をリブートしてアクティブにします。

```
# init 6
```

▼ ブート環境をアクティブにする(コマンド行インタフェース)

次の手順は、新しいブート環境を切り替えて現在稼働中のブート環境にします。

x86のみ-x86 システムの場合は、GRUB メニューを使用してアクティブ化することもできます。ただし、次のような例外があります。

- ブート環境が **Solaris 8、9**、または **10 3/05** リリースで作成されている場合、このブート環境は、必ず `luactivate` コマンドまたは「Activate」メニューを使ってアクティブ化する必要があります。これらの古いブート環境は、GRUB メニューには表示されません。
- ブート環境をはじめてアクティブ化する場合は、`luactivate` コマンドまたは「Activate」メニューを使用する必要があります。次のブート時には、そのブート環境の名前が GRUB メインメニューに表示されます。それにより、GRUB メニューで適切なエントリを選択して、このブート環境に切り替えることができます。

125 ページの「x86: GRUB メニューを使ったブート環境のアクティブ化」を参照してください。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のように入力して、ブート環境をアクティブにします。

```
# /sbin/luactivate BE_name
```

`BE_name` アクティブにするブート環境の名前を指定します。

- 3 リブートします。

```
# init 6
```



注意 - リブートには `init` か `shutdown` コマンドを使用してください。 `reboot` や `halt`、`uadmin` コマンドを使用すると、ブート環境の切り替えは行われません。以前にアクティブであったブート環境が再びブートされます。

例 5-14 ブート環境をアクティブにする(コマンド行インタフェース)

この例では、次のリブート時に `second_disk` ブート環境がアクティブになります。

```
# /sbin/luactivate second_disk
# init 6
```

▼ ブート環境をアクティブにしてファイルを同期させる(コマンド行インタフェース)

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。ここでいう「同期」とは、いくつかのクリティカルなシステムファイルやディレクトリを、以前にアクティブだったブート環境から、ブート中のブート環境へコピーすることです。Solaris Live Upgrade では、強制的に同期を行うように `luactivate` コマンドの `-s` オプションで指定しない限り、2回目以降のブート時には同期は行われません。

x86のみ-GRUBメニューを使ってブート環境間で切り替えを行なった場合も、ファイルは同期されません。ファイルを同期させるには、次の手順を使用する必要があります。

同期の詳細については、53ページの「ブート環境間でのファイルの同期」を参照してください。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solarisのシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

- 2 次のように入力して、ブート環境をアクティブにします。

```
# /sbin/luactivate -s BE_name
```

`-s` 以前にアクティブであったブート環境と新しいブート環境のファイルを同期します。ブート環境の最初のアクティブ化ではファイル間の同期が行われませんが、それ以降のアクティブ化では `-s` オプションを指定しない限りファイルの同期は行われません。



注意-以前のアクティブブート環境で発生した変更がユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、このオプションを使用するには十分注意してください。たとえば、現在のブート環境で Solaris 10 11/06 ソフトウェアを実行していて、強制的な同期処理を行なったあとで、Solaris 9 リリースにブート環境を戻したとします。この場合、Solaris 9 リリースでファイルが変更されることがあります。ファイルは OS のリリースに依存しているため、Solaris 9 リリースのブートは失敗する可能性があります。Solaris 10 11/06 のファイルと Solaris 9 のファイルは互換性があるとは限らないからです。

`BE_name` アクティブにするブート環境の名前を指定します。

3 リブートします。

```
# init 6
```

例 5-15 ブート環境をアクティブにする(コマンド行インタフェース)

この例では、次のリブート時に `second_disk` ブート環境がアクティブになり、ファイルの同期がとられます。

```
# /sbin/luactivate -s second_disk
# init 6
```

x86: GRUB メニューを使ったブート環境のアクティブ化

GRUB メニューでもブート環境間での切り替えを行うことができます。GRUB メニューは、`luactivate` コマンドまたは「Activate」メニューによるアクティブ化(ブート)に代わるものです。次の表に、GRUB メニューを使用するときの注意と制限を示します。

表 5-3 x86: GRUB メニューを使ったアクティブ化の概要

| 作業 | 説明 | 参照先 |
|-------------------|--|--|
| 注意 | ブート環境をアクティブ化したあとは、BIOS ディスク順序を変更しないでください。この順序を変更すると、GRUB メニューが無効になる可能性があります。この問題が発生した場合は、ディスク順序を元の順序に変更すれば、GRUB メニューが修正されます。 | |
| ブート環境のはじめでのアクティブ化 | ブート環境をはじめてアクティブ化する場合は、 <code>luactivate</code> コマンドまたは「Activate」メニューを使用する必要があります。次のブート時には、そのブート環境の名前が GRUB メインメニューに表示されます。それにより、GRUB メニューで適切なエントリを選択して、このブート環境に切り替えることができます。 | 123 ページの「ブート環境をアクティブにする(コマンド行インタフェース)」 |
| ファイルの同期 | はじめてブート環境をアクティブ化すると、現在のブート環境と新しいブート環境の間でファイルが同期されます。以降のアクティブ化では、ファイルは同期されません。GRUB メニューを使ってブート環境間で切り替えを行なった場合も、ファイルは同期されません。 <code>-s</code> オプションを指定して <code>luactivate</code> コマンドを使用すると、強制的に同期処理を実行できます。 | 124 ページの「ブート環境をアクティブにしてファイルを同期させる(コマンド行インタフェース)」 |

表 5-3 x86: GRUB メニューを使ったアクティブ化の概要 (続き)

| 作業 | 説明 | 参照先 |
|------------------------------------|--|---|
| Solaris 10 1/06 リリースより前に作成されたブート環境 | ブート環境が Solaris 8、9、または 10 3/05 リリースで作成されている場合、このブート環境は、必ず <code>luactivate</code> コマンドまたは「Activate」メニューを使ってアクティブ化する必要があります。これらの古いブート環境は、GRUB メニューには表示されません。 | 123 ページの「ブート環境をアクティブにする (コマンド行インタフェース)」 |
| GRUB メニューエントリの編集またはカスタマイズ | <p><code>menu.lst</code> ファイルには、GRUB メニューに表示される情報が含まれています。このファイルは、次の理由のために修正することができます。</p> <ul style="list-style-type: none"> ■ Solaris OS 以外のオペレーティングシステムのために GRUB メニューエントリを追加する。 ■ ブート動作をカスタマイズする。たとえば、ブートを詳細モードに変更したり、自動的に OS をブートするデフォルトの時間を変更したりできます。 | <ul style="list-style-type: none"> ■ 『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の「GRUB ベースのブート (概要)」 ■ 『Solaris のシステム管理 (基本編)』の第 11 章「GRUB ベースのブート (手順)」 |
| | <p>注 - GRUB メニューを変更する場合は、<code>menu.lst</code> ファイルを見つける必要があります。詳細な手順については、155 ページの「x86: GRUB メニューの <code>menu.lst</code> ファイルの検出 (作業)」を参照してください。</p> | |
| | <p>注意 - Solaris Live Upgrade エントリを変更する場合は、GRUB <code>menu.lst</code> ファイルを使用しないでください。変更すると、Solaris Live Upgrade が失敗する可能性があります。<code>menu.lst</code> ファイルを使用してブート動作をカスタマイズすることはできますが、推奨されるカスタマイズの方法は <code>eeeprom</code> コマンドを使用することです。<code>menu.lst</code> ファイルを使用してカスタマイズすると、ソフトウェアのアップグレード中に Solaris OS エントリが変更される場合があります。ファイルへの変更は失われる可能性があります。</p> | |

▼ x86: GRUB メニューを使ってブート環境をアクティブ化する (コマンド行インタフェース)

GRUB メニューを使って 2 つのブート環境間での切り替えを行うことができます。次の制限に注意してください。

- ブート環境の最初のアクティブ化は、`luactivate` コマンドまたは「Activate」メニューを使って行う必要があります。最初のアクティブ化のあと、ブート環境が GRUB メニューに表示されます。そのあと、ブート環境を GRUB メニューからブートすることができます。

- 注意 - GRUB メニューを使ってブート環境を切り替えると、同期処理が省略されます。ファイルの同期の詳細は、54 ページの「ブート環境間での強制的な同期」のリンクを参照してください。
- ブート環境が **Solaris 8、9**、または **10 3/05** リリースで作成されている場合、このブート環境は、必ず `luactivate` コマンドまたは「Activate」メニューを使ってアクティブ化する必要があります。これらの古いブート環境は、GRUB メニューには表示されません。

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 システムをリブートします。

```
# init 6
```

GRUB メインメニューが表示されます。Solaris と、Solaris Live Upgrade ブート環境である `second_disk` という、2つのオペレーティングシステムがリストされます。`failsafe` エントリは、なんらかの理由で主 OS がブートしない場合の回復用です。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
```

```
+-----+
|Solaris                               |
|Solaris failsafe                      |
|second_disk                           |
|second_disk failsafe                  |
+-----+
```

```
Use the ^ and v 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.
```

3 ブート環境をアクティブ化するには、矢印キーを使用して目的のブート環境を選択し、**Return** キーを押します。

選択したブート環境がブートされ、アクティブなブート環境になります。

障害回復:元のブート環境へのフォールバック(作業)

この章では、アクティブ化によって発生した問題を解決する方法について説明します。

アップグレードのあとに障害が検出されたり、アップグレードされたコンポーネントにアプリケーションが対応できない場合は、次のいずれかの手順(プラットフォームによって異なる)を使って元のブート環境にフォールバックしてください。

- **SPARC** システムの場合:
 - 130 ページの「SPARC: 新しいブート環境のアクティブ化に成功した場合のフォールバック」
 - 130 ページの「SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック」
 - 131 ページの「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」
- **x86** システムの場合:
 - 133 ページの「x86: 新しいブート環境のアクティブ化に成功した場合の GRUB メニューを使ったフォールバック」
 - 134 ページの「x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューを使ったフォールバック」
 - 136 ページの「x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューと DVD または CD を使ったフォールバック」

SPARC:元のブート環境へのフォールバック(コマンド行インタフェース)

元のブート環境には、次の3つの方法でフォールバックできます。

- 130 ページの「SPARC: 新しいブート環境のアクティブ化に成功した場合のフォールバック」
- 130 ページの「SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック」
- 131 ページの「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」

▼ SPARC: 新しいブート環境のアクティブ化に成功した場合のフォールバック

新しいブート環境のアクティブ化に成功したが、その結果に満足できない場合は、この手順を使用します。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /sbin/luactivate BE_name
```

BE_name アクティブにするブート環境の名前を指定します。

- 3 リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

▼ SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック

- 新しいブート環境のブートに失敗した場合で、元のブート環境をシングルユーザーモードでブートできる場合は、この手順を使って元のブート環境にフォールバックします。
- メディアやネットインストールイメージからブートする場合は、[131 ページ](#)の「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」を参照してください。

- 1 OK プロンプトで、**Solaris Operating System DVD**、**Solaris SOFTWARE - 1 CD**、ネットワーク、またはローカルディスクからマシンをシングルユーザーモードで起動します。

```
OK boot device_name -s
```

device_name システムをブートするデバイスの名前を指定します (たとえば、`/dev/dsk/c0t0d0s0`)。

- 2 次のコマンドを入力します。

```
# /sbin/luactivate BE_name
```

BE_name アクティブにするブート環境の名前を指定します。

- このコマンドを実行してもプロンプトが表示されない場合は、131 ページの「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」へ進みます。
- プロンプトが表示される場合は、次の手順に進んでください。

3 プロンプトに対して、次のコマンドを入力します。

```
Do you want to fallback to activate boot environment <disk name>
(yes or no)? yes
```

フォールバックによるアクティブ化が正常に行われたことを示すメッセージが表示されます。

4 リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

▼ SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする

DVD、CD、ネットワークインストールイメージ、またはブート可能な別のディスクからブートするには、この手順を使用します。この場合、以前のアクティブブート環境からルート (/) スライスのマウントする必要があります。そのあとで、ブート環境の切り替えを行う `luactivate` コマンドを実行してください。リブートすると、以前のアクティブブート環境が再び起動されます。

1 OK プロンプトで、Solaris Operating System DVD、Solaris SOFTWARE - 1 CD、ネットワーク、またはローカルディスクからマシンをシングルユーザーモードで起動します。

```
OK boot cdrom -s
```

または、

```
OK boot net -s
```

または、

```
OK boot device_name -s
```

`device_name` オペレーティングシステムのコピーが格納されているディスクおよびスライスの名前を指定します (たとえば、`/dev/dsk/c0t0d0s0`)。

- 必要に応じて、フォールバックブート環境のルート (/) ファイルシステムの整合性を確認します。

```
# fsck device_name
```

device_name フォールバックするブート環境のディスクデバイスにあるルート (/) ファイルシステムの名前を指定します。デバイス名は、`/dev/dsk/cwtxdysz` の形式で入力します。

- アクティブブート環境のルート (/) スライスをディレクトリ (/mnt など) にマウントします。

```
# mount device_name /mnt
```

device_name フォールバックするブート環境のディスクデバイスにあるルート (/) ファイルシステムの名前を指定します。デバイス名は、`/dev/dsk/cwtxdysz` の形式で入力します。

- アクティブブート環境のルート (/) スライスから、次のように入力します。

```
# /mnt/sbin/luactivate
```

前の稼働ブート環境がアクティブになり、結果が示されます。

- /mnt をマウント解除します。

```
# umount /mnt
```

- リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

x86: 元のブート環境へのフォールバック

元のブート環境にフォールバックするには、ご使用の環境にもっとも適した手順を選択してください。

- 133 ページの「x86: 新しいブート環境のアクティブ化に成功した場合の GRUB メニューを使ったフォールバック」
- 134 ページの「x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューを使ったフォールバック」
- 136 ページの「x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューと DVD または CD を使ったフォールバック」


```
|Solaris failsafe |
|second_disk |
|second_disk failsafe |
+-----+
Use the ^ and v 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.
```

元のブート環境である Solaris を選択します。

▼ x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューを使ったフォールバック

ブート中に障害が発生した場合、次の手順を使用して元のブート環境にフォールバックします。この例では、GRUB メニューは適切に表示されますが、新しいブート環境のブートに失敗します。デバイスは /dev/dsk/c0t4d0s0 です。元のブート環境 c0t4d0s0 がアクティブブート環境になります。



注意 - Solaris 10 3/05 リリースでは、以前のブート環境と新しいブート環境が異なるディスク上にある場合に推奨されるフォールバック方法には、BIOS でのハードディスクのブート順序の変更が含まれていました。**Solaris 10 1/06** 以降のリリースでは、BIOS ディスク順序を変更する必要はなく、変更しないことを強くお勧めします。BIOS ディスク順序を変更すると、GRUB メニューが無効になることがあり、ブート環境がブートできなくなることがあります。BIOS ディスク順序が変更されている場合は、この順序を元の設定に戻せば、システムの機能が復元されます。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 GRUB メニューを表示するにはシステムをリブートします。

```
# init 6
```

GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris |
|Solaris failsafe |
|second_disk |
|second_disk failsafe |
+-----+
Use the ^ and v 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.

- 3 GRUB メニューから、元のブート環境を選択します。ブート環境は、GRUB ソフトウェアで作成されている必要があります。Solaris 10 1/06 リリースより前に作成されたブート環境は、GRUB ブート環境ではありません。ブート可能な GRUB ブート環境がない場合は、この手順と、136 ページの「x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューと DVD または CD を使ったフォールバック」の手順をスキップしてください。

- 4 GRUB メニューを編集して、シングルユーザーモードでブートします。

- a. GRUB メインメニューを編集するには、**e** と入力します。

GRUB 編集メニューが表示されます。

```
root (hd0,2,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive
```

- b. 矢印キーを使用して、元のブート環境のカーネルエントリを選択します。

- c. ブートエントリを編集するには、**e** と入力します。

GRUB 編集メニューにカーネルエントリが表示されます。

```
grub edit>kernel /boot/multiboot
```

- d. **-s** と入力し、**Enter** キーを押します。

次の例では、**-s** オプションの配置に注意してください。

```
grub edit>kernel /boot/multiboot -s
```

- e. シングルユーザーモードでブート処理を開始するには、**b** と入力します。

- 5 必要に応じて、フォールバックブート環境のルート (/) ファイルシステムの整合性を確認します。

```
# fsck mount_point
```

mount_point 信頼性のあるルート (/) ファイルシステム

- 6 元のブート環境のルートスライスをいずれかのディレクトリ (/mnt など) にマウントします。

```
# mount device_name /mnt
```

device_name フォールバックするブート環境のディスクデバイスにあるルート (/) ファイルシステムの名前を指定します。デバイス名は、/dev/dsk/cwtxdysz の形式で入力します。

- 7 アクティブブート環境のルートスライスから、次のように入力します。

```
# /mnt/sbin/luactivate
```

前の稼働ブート環境がアクティブになり、結果が示されます。

- 8 /mnt をマウント解除します。

```
# umount /mnt
```

- 9 リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

▼ x86: ブート環境のアクティブ化に失敗した場合の GRUB メニューと DVD または CD を使ったフォールバック

ブート中に障害が発生した場合、次の手順を使用して元のブート環境にフォールバックします。この例では、新しいブート環境のブートに失敗しています。また、GRUB メニューは表示されません。デバイスは /dev/dsk/c0t4d0s0 です。元のブート環境 c0t4d0s0 がアクティブブート環境になります。



注意 - Solaris 10 3/05 リリースでは、以前のブート環境と新しいブート環境が異なるディスク上にある場合に推奨されるフォールバック方法には、BIOS でのハードディスクのブート順序の変更が含まれていました。Solaris 10 1/06 以降のリリースでは、BIOS ディスク順序を変更する必要はなく、変更しないことを強くお勧めします。BIOS ディスク順序を変更すると、GRUB メニューが無効になることがあり、ブート環境がブートできなくなることがあります。BIOS ディスク順序が変更されている場合は、この順序を元の設定に戻せば、システムの機能が復元されます。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) を挿入します。
- 3 DVD または CD からブートします。

```
# init 6
```

GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                                     |
|Solaris failsafe                           |
+-----+
Use the ^ and v 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.
```

4 GRUB メニューを編集して、シングルユーザーモードでブートします。

a. GRUB メインメニューを編集するには、e**と入力します。**

GRUB 編集メニューが表示されます。

```
root (hd0,2,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive
```

b. 矢印キーを使用して、元のブート環境のカーネルエントリを選択します。

c. ブートエントリを編集するには、e**と入力します。**

エディタにカーネルエントリが表示されます。

```
grub edit>kernel /boot/multiboot
```

d. **-sと入力し、**Enter**キーを押します。**

次の例では、**-s** オプションの配置に注意してください。

```
grub edit>kernel /boot/multiboot -s
```

e. シングルユーザーモードでブート処理を開始するには、b**と入力します。**

5 必要に応じて、フォールバックブート環境のルート (/) ファイルシステムの整合性を確認します。

```
# fsck mount_point
```

mount_point 信頼性のあるルート (/) ファイルシステム

6 元のブート環境のルートスライスをいずれかのディレクトリ (/mnt など) にマウントします。

```
# mount device_name /mnt
```

device_name フォールバックするブート環境のディスクデバイスにあるルート (/) ファイルシステムの名前を指定します。デバイス名は、*/dev/dsk/cwtxdysz* の形式で入力します。

- 7 アクティブブート環境のルートスライスから、次のように入力します。

```
# /mnt/sbin/luactivate
```

```
Do you want to fallback to activate boot environment c0t4d0s0  
(yes or no)? yes
```

前の稼働ブート環境がアクティブになり、結果が示されます。

- 8 /mnt をマウント解除します。

```
# umount device_name
```

device_name フォールバックするブート環境のディスクデバイスにあるルート (/) ファイルシステムの名前を指定します。デバイス名は、
/dev/dsk/cwtxdysz の形式で入力します。

- 9 リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

Solaris Live Upgrade ブート環境の管理 (作業)

この章では、ブート環境のファイルシステムを最新の状態に維持したり、ブート環境を削除するなど、さまざまな管理作業について説明します。この章の内容は次のとおりです。

- 139 ページの「Solaris Live Upgrade 管理作業の概要」
- 140 ページの「すべてのブート環境のステータスの表示」
- 142 ページの「以前に構成されたブート環境の更新」
- 144 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」
- 145 ページの「ブート環境の比較」
- 146 ページの「非アクティブブート環境の削除」
- 148 ページの「アクティブブート環境の名前の表示」
- 149 ページの「ブート環境の名前の変更」
- 150 ページの「ブート環境名に関連付ける説明の作成または変更」
- 153 ページの「ブート環境の構成の表示」

Solaris Live Upgrade 管理作業の概要

表 7-1 Solaris Live Upgrade 管理作業の概要

| 作業 | 説明 | 参照先 |
|---------------------|---|--|
| (省略可能) ステータスを表示します。 | <ul style="list-style-type: none"> ■ ブート環境の状態 (アクティブ、アクティブ化の処理中、アクティブになるようにスケジュールされている、比較処理中) を表示します。 ■ アクティブブート環境と非アクティブブート環境を比較します。 | <ul style="list-style-type: none"> ■ 140 ページの「すべてのブート環境のステータスの表示」 ■ 145 ページの「ブート環境の比較」 |

表 7-1 Solaris Live Upgrade 管理作業の概要 (続き)

| 作業 | 説明 | 参照先 |
|---------------------------|---|---|
| | <ul style="list-style-type: none"> ■ アクティブブート環境の名前を表示します。 ■ ブート環境の構成を表示します。 | <ul style="list-style-type: none"> ■ 148 ページの「アクティブブート環境の名前の表示」 ■ 153 ページの「ブート環境の構成の表示」 |
| (省略可能) 非アクティブブート環境を更新します。 | ファイルシステムの構成を変更することなく、アクティブブート環境からファイルシステムを再度コピーします。 | 142 ページの「以前に構成されたブート環境の更新」 |
| (省略可能) その他の作業。 | <ul style="list-style-type: none"> ■ ブート環境を削除します。 ■ ブート環境の名前を変更します。 ■ ブート環境の名前に関連付ける説明を作成または変更します。 ■ スケジュールされているジョブを取り消します。 | <ul style="list-style-type: none"> ■ 146 ページの「非アクティブブート環境の削除」 ■ 149 ページの「ブート環境の名前の変更」 ■ 150 ページの「ブート環境名に関連付ける説明の作成または変更」 ■ 144 ページの「スケジュールされた処理(作成/アップグレード/コピー)の取り消し」 |

すべてのブート環境のステータスの表示

ブート環境についての情報を表示するには、「Status」メニューまたは `lustatus` コマンドを使用してください。ブート環境を指定しない場合は、システム上のすべてのブート環境のステータス情報が表示されます。

各ブート環境について、次の詳細情報が表示されます。

- **Name** - 各ブート環境の名前。
- **Complete** - コピー処理、作成処理とも進行中ではなく、ブート環境をブートできる状態であることを示します。作成処理またはアップグレード処理が進行中であったり失敗した場合などは、ステータスは未完了として示されます。たとえば、あるブート環境のコピー処理が進行中であるか、コピー処理がスケジュールされている場合は、そのブート環境は未完了とみなされます。
- **Active** - アクティブブート環境であるかどうかを示します。
- **ActiveOnReboot** - システムの次のリブート時にそのブート環境がアクティブになるかどうかを示します。
- **CopyStatus** - ブート環境の作成またはコピーの状態(作成またはコピーがスケジュールされている、アクティブ、またはアップグレード中)を示します。ステータスが `SCHEDULED` の場合、Solaris Live Upgrade のコピー、名前変更、アップグレードの各処理を行うことはできません。

▼ すべてのブート環境のステータスを表示する (キャラクタユーザーインタフェース)

- ▶ メインメニューから「Status」を選択します。

次のような表が表示されます。

| boot environment
Name | Is
Complete | Active
Now | Active
OnReboot | Can
Delete | Copy
Status |
|--------------------------|----------------|---------------|--------------------|---------------|----------------|
| disk_a_S9 | yes | yes | yes | no | - |
| disk_b_S10database | yes | no | no | yes | COPYING |
| disk_b_S9a | no | no | no | yes | - |

注 - この例では、disk_b_S9a と disk_b_S10database に対してコピー、名前変更、アップグレードの各処理を行うことができません。これは、disk_b_S9a は未完了の状態で、disk_b_S10database は Solaris Live Upgrade による処理中だからです。

▼ すべてのブート環境のステータスを表示する (コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# lustatus BE_name
```

BE_name ステータスを表示する非アクティブブート環境の名前を指定します。
BE_name を省略すると、lustatus によりシステム内のすべてのブート環境のステータスが表示されます。

この例では、すべてのブート環境のステータスが表示されます。

```
# lustatus
boot environment Is Active Active Can Copy
Name Complete Now OnReboot Delete Status
-----
disk_a_S9 yes yes yes no -
disk_b_S10database yes no no yes COPYING
disk_b_S9a no no no yes -
```

注-disk_b_S9a と disk_b_S10database に対してコピー、名前変更、アップグレードの各処理を行うことができません。これは、disk_b_S9a は未完了の状態で、disk_b_S10database は Solaris Live Upgrade による処理中だからです。

以前に構成されたブート環境の更新

「Copy」メニューまたは lumake コマンドを使用して、以前に構成されたブート環境の内容を更新できます。アクティブ(ソース)ブート環境のファイルシステムがターゲットブート環境にコピーされます。ターゲット上にあったデータは破棄されます。コピー元のブート環境のステータスは、「complete」である必要があります。ブート環境のステータスを確認する方法については、140 ページの「すべてのブート環境のステータスの表示」を参照してください。

コピー作業はあとで行われるようにスケジュールできます。スケジュールできるのは一度に1つのジョブだけです。スケジュールされたコピー処理を取り消す方法については、144 ページの「スケジュールされた処理(作成/アップグレード/コピー)の取り消し」を参照してください。

▼ 以前に構成されたブート環境を更新する(キャラクターインタフェース)

- 1 メインメニューから「Copy」を選択します。
- 2 更新する非アクティブブート環境の名前を入力します。
Name of Target Boot Environment: **solaris8**
- 3 コピー処理を継続するか、またはあとでコピーが実行されるようにスケジュールします。
 - コピーを継続するには、Return キーを押します。
以上の手順で、非アクティブブート環境が更新されます。
 - あとでコピーが実行されるようにスケジュールするには、「y」と入力し、時刻(at コマンドの書式を使用)と、結果の送信先電子メールアドレスを指定します。

```
Do you want to schedule the copy? y
Enter the time in 'at' format to schedule copy: 8:15 PM
Enter the address to which the copy log should be mailed:
someone@anywhere.com
```

時間の書式については、at(1)のマニュアルページを参照してください。

以上の手順で、非アクティブブート環境が更新されます。

スケジュールされたコピー処理を取り消す方法については、144 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」を参照してください。

▼ 以前に構成されたブート環境を更新する (コマンド行インタフェース)

この手順では、以前に作成されたブート環境上の古いファイルを上書きしてソースファイルをコピーします。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# lumake -n BE_name [-s source_BE] [-t time] [-m email_address]
```

| | |
|-------------------------|--|
| -n <i>BE_name</i> | ファイルシステムを書き換えるブート環境の名前を指定します。 |
| -s <i>source_BE</i> | (省略可能) ターゲットブート環境にコピーするファイルシステムがあるソースブート環境の名前を指定します。このオプションを省略すると、lumake は現在のブート環境をソースとして使用します。 |
| -t <i>time</i> | (省略可能) 指定されたブート環境上のファイルを指定された時刻に上書きするバッチジョブを設定します。時刻は、at(1) のマニュアルページに指定されている書式で入力します。 |
| -m <i>email_address</i> | (省略可能) コマンドが完了した時点で、ここで指定する電子メールアドレスに lumake の出力を送ります。email_address はチェックされません。このオプションは、-t と併用する必要があります。 |

例 7-1 以前に構成されたブート環境を更新する (コマンド行インタフェース)

この例では、first_disk のファイルシステムが second_disk にコピーされます。処理が完了した時点で、電子メールが Joe@anywhere.com 宛に送信されます。

```
# lumake -n second_disk -s first_disk -m joe@anywhere.com
```

first_disk 上のファイルが second_disk にコピーされ、通知の電子メールが送信されません。スケジュールされたコピー処理を取り消す方法については、144 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」を参照してください。

スケジュールされた処理 (作成/アップグレード/コピー) の取り消し

ブート環境のスケジュールされた処理 (作成、アップグレード、コピー) は、その処理の開始前に取り消すことができます。GUI では、「Create a Boot Environment」、「Upgrade a Boot Environment」、または「Copy a Boot Environment」メニューを使用して、特定の時間に処理が実行されるようにスケジュールすることができます。CLI では、`lumake` コマンドを使用して、処理をスケジュールできます。システムでスケジュールできるジョブは一度に1つだけです。

▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (キャラクタインタフェース)

- 1 メインメニューから「Cancel」を選択します。
- 2 取り消しが可能なブート環境の一覧を表示するには、**F2**を押します。
- 3 取り消すブート環境を選択します。
これで、指定されている時刻に処理は実行されなくなります。

▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 次のコマンドを入力します。

```
# lumancel
```


これで、指定されている時刻に処理は実行されなくなります。

ブート環境の比較

アクティブブート環境とその他のブート環境の相違を確認するには、「Compare」メニューまたは `lucompare` コマンドを使用します。比較をするためには、非アクティブブート環境は完了状態で、コピー処理がスケジュールされていない必要があります。140 ページの「すべてのブート環境のステータスの表示」を参照してください。

`lumount` または `mount` を使用してマウントされたパーティションのあるブート環境は、指定できません。

▼ ブート環境を比較する (キャラクタユーザーインタフェース)

- 1 メインメニューから「Compare」を選択します。
- 2 「Compare to Original」または「Compare to an Active Boot Environment」を選択します。
- 3 F3 を押します。
- 4 次に示すように、元の (アクティブ) ブート環境の名前、非アクティブブート環境の名前、およびファイルのパスを入力します。

```
Name of Parent: solaris8
Name of Child: solaris8-1
Full Pathname of the file to Store Output: /tmp/compare
```

- 5 F3 を押してファイルに保存します。
「Compare」メニューに次の属性が表示されます。
 - 「モード」。
 - 「リンクの数」。
 - 「所有者」。
 - 「グループ」。
 - チェックサム - 指定されたブート環境内のファイルとこれに対応するアクティブブート環境内のファイルが、前述したすべてのフィールドにおいて一致する場合だけ、チェックサムを計算します。すべて一致するがチェックサムは異なるという場合には、異なるチェックサムが比較対象ファイルのエントリに付加されます。
 - 「サイズ」。
 - いずれか一方のブート環境だけに存在するファイル。
- 6 F3 を押して「Compare」メニューに戻ります。

▼ ブート環境を比較する(コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/lucompare -i infile
```

または

```
# /usr/sbin/lucompare -t -o outfile BE_name
```

-i infile *infile* 中に指定されたファイルを比較します。比較するファイルは、絶対パスで指定する必要があります。ファイルのエントリがディレクトリである場合、比較はディレクトリに対して再帰的に行われます。このオプションまたは **-t** のいずれか一方を使用できます (両方は使用できません)。

-t バイナリ以外のファイルだけを比較します。この比較では、ファイルごとに `file(1)` コマンドを使用してそのファイルがテキストファイルであるかを確認します。ユーザーは、このオプションまたは **-i** のいずれか一方を使用できます (両方は使用できません)。

-o outfile 相違についての出力を *outfile* にリダイレクトします。

BE_name アクティブブート環境と比較するブート環境の名前を指定します。

例 7-2 ブート環境の比較(コマンド行インタフェース)

この例では、`first_disk` ブート環境 (ソース) と `second_disk` ブート環境が比較され、結果がファイルに出力されます。

```
# /usr/sbin/lucompare -i /etc/lu/compare/ \  
-o /var/tmp/compare.out second_disk
```

非アクティブブート環境の削除

「Delete」メニューまたは `ludelete` コマンドを使用してブート環境を削除してください。次の制限に注意してください。

- アクティブブート環境および次のリブートでアクティブになるブート環境は、削除できません。

- 削除するブート環境は完了状態でなければなりません。完了状態のブート環境とは、ステータスを変更する処理が終了している環境を指します。ブート環境のステータスを確認する方法については、140 ページの「すべてのブート環境のステータスの表示」を参照してください。
- `lumount` を使用してファイルシステムをマウントしているブート環境を削除することはできません。
- x86 のみ: **Solaris 10 1/06** 以降のリリースでは、アクティブな GRUB メニューがあるブート環境を削除することはできません。ブート環境を再使用するには、`lumake` コマンドまたは `luupgrade` コマンドを使用します。アクティブな GRUB メニューがあるブート環境の確認については、155 ページの「x86: GRUB メニューの `menu.lst` ファイルの検出(作業)」を参照してください。

▼ 非アクティブブート環境を削除する(キャラクターインタフェース)

- 1 メインメニューから「Delete」を選択します。
- 2 削除する非アクティブブート環境の名前を入力します。

```
Name of boot environment: solaris8
```

指定された非アクティブブート環境が削除されます。

▼ 非アクティブブート環境を削除する(コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。
- 2 次のコマンドを入力します。

```
# ludelete BE_name
```

`BE_name` 削除する非アクティブブート環境の名前を指定します。

例 7-3 非アクティブブート環境の削除(コマンド行インタフェース)

この例では、ブート環境 `second_disk` が削除されます。

```
# ludelete second_disk
```

アクティブブート環境の名前の表示

現在のブート環境の名前を表示するには、「Current」メニューまたは `lucurr` コマンドを使用してください。システム上に構成されたブート環境がない場合は、「No Boot Environments are defined」というメッセージが表示されます。`lucurr` で表示されるのは現在のブート環境の名前だけです。次のブート時にアクティブになるブート環境の名前は表示されません。ブート環境のステータスを確認する方法については、[140 ページの「すべてのブート環境のステータスの表示」](#)を参照してください。

▼ アクティブブート環境の名前を表示する (キャラクターインタフェース)

- ▶ メインメニューから「Current」を選択します。
アクティブブート環境の名前または「No Boot Environments are defined」というメッセージが表示されます。

▼ アクティブブート環境の名前を表示する (コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 次のコマンドを入力します。

```
# /usr/sbin/lucurr
```

例 7-4 アクティブブート環境の名前の表示 (コマンド行インタフェース)

この例では、現在のブート環境の名前が表示されます。

```
# /usr/sbin/lucurr  
solaris8
```

ブート環境の名前の変更

ブート環境の名前の変更は、ブート環境の Solaris リリースを別のリリースにアップグレードする場合などに便利です。たとえば、オペレーティングシステムのアップグレード処理後に、ブート環境の名前を `solaris8` から `solaris10` に変更できます。

非アクティブブート環境の名前を変更する場合は、「Rename」メニューまたは `lurename` コマンドを使用してください。

x86のみ - Solaris 10 1/06以降のリリースでは、GRUBメニューは、「Rename」メニューまたは `lurename` コマンドを使用すると自動的に更新されます。更新された GRUBメニューでは、ブートエントリのリストにブート環境の名前が表示されます。GRUBメニューの詳細については、55 ページの「x86: GRUB メニューを使ったブート環境のアクティブ化」を参照してください。

GRUBメニューの `menu.lst` ファイルの検出については、155 ページの「x86: GRUB メニューの `menu.lst` ファイルの検出(作業)」を参照してください。

表 7-2 ブート環境の命名の制約

| 制約 | 参照先 |
|--|--|
| 名前のは、30 文字以内にする必要があります。 | |
| 名前は、英数字またはほかの ASCII 文字 (UNIX シェルで特別な意味を持つ文字を除く) だけで構成できます。 | sh(1) の「クォート」の節を参照してください。 |
| 名前に使用できるのは、8 ビットで表現できるシングルバイトの文字だけです。 | |
| 名前は、システム上で一意となるように指定する必要があります。 | |
| ブート環境の名前を変更するためには、そのステータスが「complete」である必要があります。 | ブート環境のステータスを確認する方法については、140 ページの「すべてのブート環境のステータスの表示」を参照してください。 |
| <code>lumount</code> または <code>mount</code> を使用してファイルをマウントしているブート環境の名前は、変更できません。 | |

▼ 非アクティブブート環境の名前を変更する (キャラクターインターフェース)

- 1 メインメニューから「Rename」を選択します。
- 2 名前を変更するブート環境を指定し、続いて新しい名前を入力します。

- 3 F3 を押して変更を保存します。

▼ 非アクティブブート環境の名前を変更する(コマンド行インタフェース)

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# lucrename -e BE_name -n new_name
```

-e *BE_name* 変更する非アクティブブート環境の名前を指定します。

-n *new_name* 非アクティブブート環境の新しい名前を指定します。

この例では、`second_disk` が `third_disk` に変更されます。

```
# lucrename -e second_disk -n third_disk
```

ブート環境名に関連付ける説明の作成または変更

ブート環境名に説明を付けることができます。この説明によって名前が置き換わることはありません。ブート環境名は長さや文字に制限がありますが、この説明は長さ、内容に制限がありません。シンプルなテキストでも、gif ファイルのような複雑なものでもかまいません。この説明は、次の時点で作成できます。

- ブート環境を作成する時点 (-A オプション指定で `lucreate` コマンドを使用する)
- ブート環境の作成後 (`ludesc` コマンドを使用する)

`lucreate` コマンドで -A オプションを使用する方法の詳細

69 ページの「ブート環境をはじめて作成する (コマンド行インタフェース)」

ブート環境の作成後に説明を作成する方法の詳細

`ludesc(1M)`

▼ テキストを使用してブート環境名の説明を作成または変更する方法

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name 'BE_description'
```

`-n BE_name 'BE_description'` ブート環境名と、ブート環境名に関連付ける新しい説明を指定します。

例 7-5 テキストを使用してブート環境名に説明を加える

この例では、`second_disk` というブート環境に説明が加えられています。この説明は、単一引用符で囲まれたテキストで記述されます。

```
# /usr/sbin/ludesc -n second_disk 'Solaris 10 11/06 test build'
```

▼ ファイルを使用してブート環境名の説明を作成または変更する方法

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name -f file_name
```

`-n BE_name` ブート環境名を指定します。

`file_name` ブート環境名に関連付ける説明が書かれているファイルを指定します。

例 7-6 ファイルを使用してブート環境名に説明を加える

この例では、`second_disk` というブート環境に説明が加えられています。説明は、`gif` ファイル内に入っています。

```
# /usr/sbin/ludesc -n second_disk -f rose.gif
```

▼ テキストで記述された説明からブート環境名を確認する方法

次のコマンドにより、指定した説明に関連付けられたブート環境の名前が戻されます。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/ludesc -A 'BE_description'
```

-A 'BE_description' ブート環境名に関連付ける説明を指定します。

例 7-7 説明からブート環境名を確認する

この例では、説明を指定して -A オプションを使用することでブート環境名 `second_disk` を確認しています。

```
# /usr/sbin/ludesc -A 'Solaris 10 11/06 test build'  
second_disk
```

▼ ファイル内の説明からブート環境名を確認する方法

次のコマンドは、ファイルに関連付けられているブート環境名を表示します。ファイルにはブート環境の説明が含まれます。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 次のコマンドを入力します。

```
# /usr/sbin/ludesc -f file_name
```

-f file_name ブート環境の説明を含むファイルの名前を指定します。

例 7-8 ファイル内の説明からブート環境名を確認する

この例では、-f オプションと、説明を含むファイルの名前を使用することでブート環境の名前 `second_disk` を確認しています。

```
# /usr/sbin/ludesc -f rose.gif
second_disk
```

▼ 名前からブート環境説明を確認する方法

この手順では、コマンドで名前を指定したブート環境の説明が表示されます。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name
```

`-n BE_name` ブート環境名を指定します。

例 7-9 ブート環境名から説明を確認する

この例では、ブート環境名を指定して `-n` オプションを使用することで説明を確認しています。

```
# /usr/sbin/ludesc -n second_disk
Solaris 10 11/06 test build
```

ブート環境の構成の表示

ブート環境の構成を表示するには、「List」メニューまたは `lufslist` コマンドを使用してください。出力される情報は、ブート環境マウントポイントごとの、ディスクスライス (ファイルシステム)、ファイルシステムの種類、およびファイルシステムのサイズです。

▼ 非アクティブブート環境の構成を表示する (キャラクタユーザーインタフェース)

- 1 メインメニューから「List」を選択します。
- 2 構成を表示するブート環境の名前を入力します。
Name of Boot Environment: **solaris8**

3 F3 を押します。

次の例ではリストを表示しています。

| Filesystem | fstype | size(Mb) | Mounted on |
|-------------------|--------|----------|------------|
| /dev/dsk/c0t0d0s1 | swap | 512.11 | - |
| /dev/dsk/c0t4d0s3 | ufs | 3738.29 | / |
| /dev/dsk/c0t4d0s4 | ufs | 510.24 | /opt |

4 F6 を押して「List」メニューに戻ります。

▼ ブート環境の構成を表示する (コマンド行インタフェース)

1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のコマンドを入力します。

```
# lufslist -n BE_name
```

BE_name ファイルシステムの詳細を表示するブート環境の名前を指定します。

次の例ではリストを表示しています。

| Filesystem | fstype | size(Mb) | Mounted on |
|-------------------|--------|----------|------------|
| /dev/dsk/c0t0d0s1 | swap | 512.11 | - |
| /dev/dsk/c0t4d0s3 | ufs | 3738.29 | / |
| /dev/dsk/c0t4d0s4 | ufs | 510.24 | /opt |

x86: GRUB メニューの menu.lst ファイルの検出(作業)

この章では、GRUB menu.lst ファイルを手動で更新する場合の、このファイルの更新方法について説明します。たとえば、デフォルトの OS がブートするまでのデフォルトの時間を変更できます。また、別の OS を GRUB メニューに追加することができます。この章では、menu.lst ファイルの検出例をいくつか示します。

GRUB ベースのブートに関する背景情報については、『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の第 6 章「Solaris インストールのための GRUB ベースのブート」を参照してください。

x86: GRUB メニューの menu.lst ファイルの検出(作業)

GRUB メニューの menu.lst ファイルを検出するには、常に bootadm コマンドを使用する必要があります。アクティブな GRUB メニューを見つけるには、list-menu サブコマンドを使用します。menu.lst ファイルには、システムにインストールされているすべてのオペレーティングシステムが一覧表示されています。このファイルの内容は、GRUB メニューに表示されるオペレーティングシステムの一覧を記述したものです。

通常、アクティブな GRUB メニューの menu.lst ファイルは /boot/grub/menu.lst に置かれています。場合によっては、GRUB menu.lst ファイルが別の場所に置かれていることもあります。たとえば、Solaris Live Upgrade を使用するシステムの場合は、GRUB menu.lst ファイルが現在稼働中のブート環境ではないブート環境に存在することもあります。また、x86 ブートパーティションがあるシステムをアップグレードした場合、menu.lst ファイルは /stubboot ディレクトリに置かれている場合があります。システムのブートには、アクティブな GRUB menu.lst ファイルのみが使用されます。システムのブート時に表示される GRUB メニューを変更するには、アクティブな GRUB menu.lst ファイルに変更を加える必要があります。それ以外の GRUB menu.lst ファイルに変更を加えても、システムのブート時に表示されるメニューに影響はありません。アクティブな GRUB menu.lst ファイルの場所を特定するには、bootadm コマンドを使用します。list-menu サブコマンドを実行すると、アクティブな GRUB メニューの場所が表示されます。次の手順で、GRUB メニューの menu.lst ファイルの場所を特定します。

bootadm コマンドの詳細については、bootadm(1M) のマニュアルページを参照してください。

▼ GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の2つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。Solaris OS はブート済みで、GRUB メニューが含まれています。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is: /boot/grub/menu.lst
default 0
timeout 10
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

▼ アクティブな menu.lst ファイルが別のブート環境にある場合の GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の2つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。この例では、現在稼働中のブート環境には menu.lst ファイルは存在しません。second_disk ブート環境がブートされています。Solaris ブート環境には GRUB メニューが含まれています。Solaris ブート環境はマウントされていません。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is: /dev/dsk/device_name(not mounted)
The filesystem type of the menu device is <ufs>
default 0
timeout 10
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

- 3 menu.lst ファイルを含むファイルシステムがマウントされていないため、ファイルシステムをマウントします。UFS ファイルシステムとデバイス名を指定します。

```
# /usr/sbin/mount -F ufs /dev/dsk/device_name /mnt
```

device_name は、マウントするブート環境のディスクデバイスにあるルート (/) ファイルシステムの場所を指定します。デバイス名は、/dev/dsk/cwtxdysz の形式で入力します。次に例を示します。

```
# /usr/sbin/mount -F ufs /dev/dsk/c0t1d0s0 /mnt
```

GRUB メニューには /mnt/boot/grub/menu.lst でアクセスできます。

- 4 ファイルシステムをマウント解除します。

```
# /usr/sbin/umount /mnt
```

注-ブート環境またはブート環境のファイルシステムをマウントする場合は、使用後に必ずファイルシステムをマウント解除してください。これらのファイルシステムをマウント解除しないと、その後 Solaris Live Upgrade を同じブート環境で実行したときに失敗する可能性があります。

▼ Solais Live Upgrade ブート環境がマウントされている場合の GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の2つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。second_disk ブート環境がブートされています。Solaris ブート環境には GRUB メニューが含まれています。Solaris ブート環境は /.alt.Solaris にマウントされています。

- 1 スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is:  
/.alt.Solaris/boot/grub/menu.lst  
default 0  
timeout 10  
0 Solaris  
1 Solaris failsafe  
2 second_disk  
3 second_disk failsafe
```

GRUB メニューを含むブート環境がすでにマウントされているため、
/.alt.Solaris/boot/grub/menu.lst で menu.lst ファイルにアクセスできます。

▼ 使用しているシステムに **x86** ブートパーティションがある場合の **GRUB** メニューの menu.lst ファイルの検出

次の手順では、システムに次の2つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。second_disk ブート環境がブートされています。システムはアップグレードされており、x86 ブートパーティションが残されています。ブートパーティションは /stubboot にマウントされ、GRUB メニューが含まれています。x86 ブートパーティションについては、『Solaris 10 11/06 インストールガイド (インストールとアップグレードの計画)』の「パーティション分割に関する推奨事項」を参照してください。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is:  
/stubboot/boot/grub/menu.lst  
default 0  
timeout 10  
0 Solaris  
1 Solaris failsafe
```

```
2 second_disk  
3 second_disk failsafe
```

menu.lst ファイルには、 /stubboot/boot/grub/menu.lst でアクセスできます。

Solaris Live Upgrade (例)

この章では、新しいブート環境を作成、アップグレード、およびアクティブ化して、これを新たな稼動環境にする例を示します。この章の内容は次のとおりです。

- 161 ページの「Solaris Live Upgrade によるアップグレードの例 (コマンド行インタフェース)」
- 168 ページの「RAID-1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース)」
- 172 ページの「既存のボリュームから Solaris ボリュームマネージャー RAID-1 ボリュームへの移行例 (コマンド行インタフェース)」
- 172 ページの「空のブート環境を作成して Solaris フラッシュアーカイブをインストールする例 (コマンド行インタフェース)」
- 174 ページの「Solaris Live Upgrade によるアップグレードの例 (キャラクターインターフェイス)」

Solaris Live Upgrade によるアップグレードの例 (コマンド行インタフェース)

この例では、Solaris 9 リリースを実行しているシステムで、`lucreate` コマンドを使用して新しいブート環境を作成します。この新しいブート環境を、`luupgrade` コマンドを使用して Solaris 10 11/06 にアップグレードします。アップグレードしたブート環境を、`luactivate` コマンドによってアクティブ化します。この節では、以前のブート環境にフォールバックする例も示します。

必要なパッチをインストールするには

| 説明 | 参照先 |
|--|--|
| <p>注意 - Solaris Live Upgrade を正しく操作するためには、指定の OS バージョン用の特定のパッチリビジョンのセットがインストールされている必要があります。Solaris Live Upgrade をインストールまたは実行する前に、これらのパッチをインストールする必要があります。</p> <p>x86 のみ - Solaris 10 1/06 以降のリリースでは、このパッチのセットがインストールされていないと、Solaris Live Upgrade が失敗し、次のエラーメッセージが表示されることがあります。次のエラーメッセージが表示されなくても、必要なパッチがインストールされていない場合があります。Solaris Live Upgrade のインストールを試みる前に、SunSolve の infodoc に記載されたすべてのパッチがすでにインストール済みであることを必ず確認してください。</p> <pre>ERROR: Cannot find or is not executable: </sbin/biosdev>. ERROR: One or more patches required by Live Upgrade has not been installed.</pre> <p>infodoc 72099 に記載されたパッチは、随時変更される可能性があります。これらのパッチにより、Solaris Live Upgrade の欠陥が修正される可能性があると同時に、Solaris Live Upgrade が依存するコンポーネントの欠陥も修正される可能性があります。Solaris Live Upgrade で問題が発生した場合は、最新の Solaris Live Upgrade パッチがインストールされていることを確認してください。</p> | <p>http://sunsolve.sun.com で最新のパッチリストを確認してください。SunSolve の Web サイトで、infodoc 72099 を検索してください。</p> |
| <p>Solaris 8、または Solaris 9 OS を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。</p> | <p>Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、Java 2 Runtime Environment 推奨パッチクラスタをインストールします。このパッチクラスタは http://sunsolve.sun.com から入手できます。</p> |

次の手順に従って必要なパッチをインストールします。

SunSolve の Web サイトから、パッチ一覧を取得します。

```
# patchadd /net/server/export/patches
# init 6
```

アクティブブート環境で Solaris Live Upgrade をインストールする方法

注-この手順では、システムでボリュームマネージャーを実行していると仮定しています。ボリュームマネージャーを使用したリムーバブルメディアの管理の詳細は、『Solaris のシステム管理 (デバイスとファイルシステム)』を参照してください。

1. Solaris Operating System DVD または Solaris SOFTWARE - 2 CD を挿入します。
2. 使用しているメディアに合わせて操作を行います。
 - Solaris Operating System DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。

- SPARC システムの場合:

```
# cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
# ./liveupgrade20
```

- x86 システムの場合:

```
# cd /cdrom/cdrom0/Solaris_10/Tools/Installers
# ./liveupgrade20
```

Solaris インストールプログラムの GUI が表示されます。

- Solaris SOFTWARE - 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```

Solaris インストールプログラムの GUI が表示されます。

3. 「インストール形式の選択 (Select Type of Install)」パネルで「カスタム (Custom)」をクリックします。
4. 「ロケールの選択 (Locale Selection)」パネルで、インストールする言語をクリックします。
5. インストールするソフトウェアを選択します。
 - DVD の場合、「コンポーネントの選択 (Component Selection)」パネルの「次へ (Next)」をクリックしてパッケージをインストールします。
 - CD の場合、「製品の選択 (Product Selection)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (Default Install)」をクリックします。また、このソフトウェアの選択を解除するには、ほかの製品をクリックします。
6. Solaris インストールプログラムの指示に従って、ソフトウェアをインストールします。

ブート環境を作成する方法

-c オプションを使用して、ソースブート環境に `c0t4d0s0` という名前を付けます。ソースブート環境の名前設定は最初のブート環境を作成するときだけ必要です。-c オプションによる命名の詳細は、「ブート環境をはじめて作成する」の[手順 2](#)を参照してください。

新しいブート環境の名前は `c0t15d0s0` です。-A オプションを使用して、このブート環境名に関連付ける説明を作成します。

ルート (/) ファイルシステムを新しいブート環境にコピーします。また、ソースブート環境のスワップスライスは共有せずに、新しいスワップスライスを作成します。

```
# lucreate -A 'BE_description' -c /dev/dsk/c0t4d0s0 -m /:/dev/dsk/c0t15d0s0:ufs\
-m -:/dev/dsk/c0t15d0s1:swap -n /dev/dsk/c0t15d0s0
```

非アクティブブート環境をアップグレードする方法

非アクティブブート環境の名前は `c0t15d0s0` です。アップグレードに使用するオペレーティングシステムイメージはネットワークから取得します。

```
# luupgrade -n c0t15d0s0 -u -s /net/ins-svr/export/Solaris_10 \
combined.solaris_wos
```

ブート環境がブート可能か確認する方法

lustatus コマンドは、ブート環境の作成が完了したかどうかを報告します。lustatus コマンドは、ブート環境がブート可能であるかどうかも報告します。

```
# lustatus
```

| boot environment
Name | Is
Complete | Active
Now | Active
OnReboot | Can
Delete | Copy
Status |
|--------------------------|----------------|---------------|--------------------|---------------|----------------|
| c0t4d0s0 | yes | yes | yes | no | - |
| c0t15d0s0 | yes | no | no | yes | - |

非アクティブブート環境をアクティブにする方法

luactivate コマンドを使用して `c0t15d0s0` ブート環境をブート可能にします。続いてシステムをリブートします。これで `c0t15d0s0` がアクティブブート環境になります。 `c0t4d0s0` ブート環境は非アクティブになります。

```
# luactivate c0t15d0s0
# init 6
```

(省略可能) ソースブート環境へフォールバックする方法

新しいブート環境のアクティブ化の状況に応じて、フォールバックのための手順を次の中から選択します。

- SPARC システムの場合:
 - アクティブ化が正常に行われたが元のブート環境に戻すという場合は、[例 9-1](#) を参照してください。
 - アクティブ化に失敗したが、元のブート環境からブート可能な場合は、[例 9-2](#) を参照してください。
 - アクティブ化に失敗し、メディアまたはネットインストールイメージを使用して元のブート環境に戻る必要がある場合は、[例 9-3](#) を参照してください。
- x86 システムで、**Solaris 10 1/06** 以降のリリースの場合および GRUB メニューを使用する場合:
 - アクティブ化に失敗し、GRUB メニューは適切に表示されるが新しいブート環境のブートに失敗する場合は、[例 9-4](#) を参照してください。
 - アクティブ化に失敗し、GRUB メニューが表示されない場合は、[例 9-5](#) を参照してください。

例 9-1 SPARC: ブート環境作成は正常に完了したが元のブート環境にフォールバックさせる場合
この例では、新しいブート環境のアクティブ化が正常に完了したにもかかわらず、元の c0t4d0s0 ブート環境をアクティブブート環境として復元しています。デバイス名は first_disk です。

```
# /sbin/luactivate first_disk
# init 6
```

例 9-2 SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック

この例では、新しいブート環境のブートに失敗しています。シングルユーザーモードで元のブート環境 c0t4d0s0 からブートさせるために、OK プロンプトを表示させる必要があります。

```
OK boot net -s
# /sbin/luactivate first_disk
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# init 6
```

例 9-2 SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック (続き)

元のブート環境 `c0t4d0s0` がアクティブブート環境になります。

例 9-3 SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする

この例では、新しいブート環境のブートに失敗しています。元のブート環境からはブートできないためメディアまたはネットインストールイメージを使用する必要があります。デバイスは `/dev/dsk/c0t4d0s0` です。元のブート環境 `c0t4d0s0` がアクティブブート環境になります。

```
OK boot net -s
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

例 9-4 x86: GRUB メニューを使用して元のブート環境にフォールバックする

次の例では、**Solaris 10 1/06** 以降のリリースで、GRUB メニューを使用してフォールバックする手順を示します。

この例では、GRUB メニューは適切に表示されますが、新しいブート環境のブートに失敗します。フォールバックを可能にするため、元のブート環境はシングルユーザーモードでブートされます。

1. スーパーユーザーになるか、同等の役割になります。
2. GRUB メニューを表示するにはシステムをリブートします。

```
# init 6
```

GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                               |
|Solaris failsafe                       |
|second_disk                            |
|second_disk failsafe                   |
+-----+
Use the ^ and v 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.
```

例 9-4 x86: GRUB メニューを使用して元のブート環境にフォールバックする (続き)

3. GRUB メニューから、元のブート環境を選択します。ブート環境は、GRUB ソフトウェアで作成されている必要があります。**Solaris 10 1/06** リリースより前に作成されたブート環境は、GRUB ブート環境ではありません。ブート可能な GRUB ブート環境がない場合は、例 9-5 に進んでください。
4. 次のように入力して、GRUB メニューを編集します。e。
5. 矢印キーを使用して kernel /boot/multiboot を選択し、e と入力します。GRUB 編集メニューが表示されます。

```
grub edit>kernel /boot/multiboot
```

6. -s と入力して、シングルユーザーモードでブートします。

```
grub edit>kernel /boot/multiboot -s
```

7. ブートして、ブート環境をマウントします。次に、それをアクティブ化します。

```
# b
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

例 9-5 x86: DVD または CD を使用して GRUB メニューで元のブート環境にフォールバックする

次の例では、**Solaris 10 1/06** 以降のリリースにおいて DVD または CD を使用してフォールバックする手順を示します。

この例では、新しいブート環境のブートに失敗しています。また、GRUB メニューは表示されません。フォールバックを可能にするため、元のブート環境はシングルユーザーモードでブートされます。

1. Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) を挿入します。
2. スーパーユーザーになるか、同等の役割になります。
3. DVD または CD からブートします。

```
# init 6
```

GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
```

```
+-----+
```

例 9-5 x86: DVD または CD を使用して GRUB メニューで元のブート環境にフォールバックする (続き)

```
|Solaris |
|Solaris failsafe |
+-----+
Use the ^ and v 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.
```

4. 次のように入力して、GRUB メニューを編集します。e。
5. 矢印キーを使用して kernel /boot/multiboot を選択し、e と入力します。GRUB 編集メニューが表示されます。

```
grub edit>kernel /boot/multiboot
```

6. -s と入力して、シングルユーザーモードでブートします。

```
grub edit>kernel /boot/multiboot -s
```

7. ブートして、ブート環境をマウントします。次に、アクティブ化してリブートします。

```
Edit the GRUB menu by typing: e
Select the original boot environment by using the arrow keys.
grub edit>kernel /boot/multiboot -s
# b
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

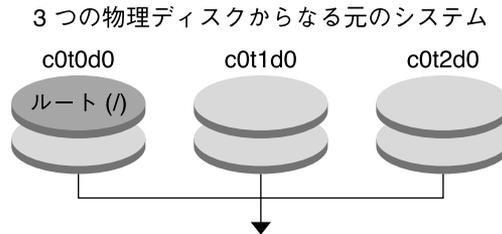
RAID-1 ボリューム(ミラー)の一方を切り離してアップグレードする例(コマンド行インタフェース)

この例では、次の作業の手順を示します。

- 新しいブート環境に RAID-1 ボリューム(ミラー)を作成します
- ミラーの一方を切り離し、アップグレードします
- ミラー(連結)の他方を新しいミラーに接続します

図 9-1 は、3つの物理ディスクから成る現在のブート環境を示します。

RAID-1 ボリューム(ミラー)の一方を切り離し、アップグレードする



ミラーを使ったブート環境の作成

図 9-1 RAID-1 ボリューム(ミラー)の一方を切り離し、アップグレードする

1. ミラーを持つ新しいブート環境 `second_disk` を作成します。

次のコマンドは、次のような処理を実行します。

- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。`d10` というミラーが作成されます。このミラー `d10` に、現在のブート環境のルート (`/`) ファイルシステムがコピーされます。ミラー `d10` にあるデータはすべて上書きされます。
- 2つのスライス `c0t1d0s0` および `c0t2d0s0` は、サブミラーとして指定されています。これら2つのサブミラーは、ミラー `d10` に接続されます。

```
# lucreate -c first_disk -n second_disk \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t1d0s0:attach \  
-m /:/dev/dsk/c0t2d0s0:attach
```

2. ブート環境 `second_disk` をアクティブ化します。

```
# /sbin/luactivate second_disk  
# init 6
```

3. 別のブート環境 `third_disk` を作成します。

次のコマンドは、次のような処理を実行します。

- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。`d20` というミラーが作成されます。
- スライス `c0t1d0s0` が現在のミラーから切り離され、ミラー `d20` に追加されます。このサブミラーの内容であるルート (`/`) ファイルシステムは保持され、コピー処理は発生しません。

```
# lucreate -n third_disk \  
-m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

4. 新しいブート環境 `third_disk` をアップグレードします。

- ```
luupgrade -u -n third_disk \
-s /net/installmachine/export/Solaris_10/OS_image
```
5. アップグレードされたブート環境にパッチを追加します。
- ```
# luupgrade -t n third_disk -s /net/patches 222222-01
```
6. ブート環境 `third_disk` をアクティブ化して、このブート環境からシステムを実行します。
- ```
/sbin/luactivate third_disk
init 6
```
7. ブート環境 `second_disk` を削除します。
- ```
# ludelete second_disk
```
8. 次のコマンドは、次のような処理を実行します。
- ミラー `d10` を消去します。
 - `c0t2d0s0` の連結の数を調べます。
 - `metastat` コマンドで見つけた連結を、ミラー `d20` に接続します。`metattach` コマンドは、新しく接続した連結と、ミラー `d20` の連結とを同期します。連結にあるデータはすべて上書きされます。
- ```
metaclear d10
metastat -p | grep c0t2d0s0
dnum 1 1 c0t2d0s0
metattach d20 dnum
```
- num*      `metastat` コマンドで見つかった連結の数。

これで、新しいブート環境 `third_disk` がアップグレードされ、この環境からシステムが実行されます。`third_disk` には、ミラー化されたルート (`/`) ファイルシステムが含まれています。

図 9-2 は、上記の例のコマンドでミラーを切り離してアップグレードする手順の全体を示しています。

## RAID-1 ボリューム(ミラー)の一方を切り離し、アップグレードする(続き)

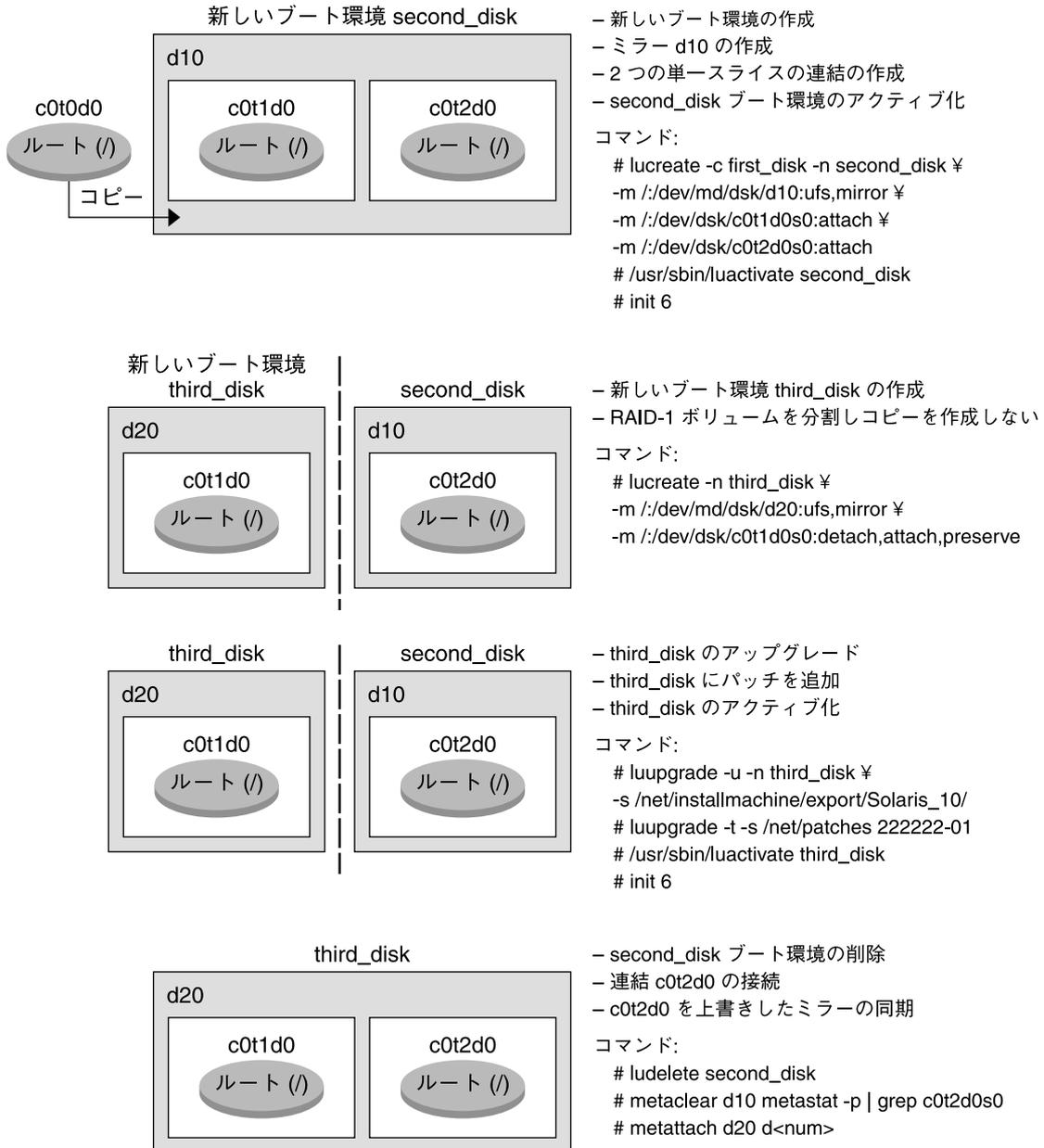


図 9-2 RAID-1 ボリューム(ミラー)の一方を切り離し、アップグレードする(続き)

## 既存のボリュームから **Solaris** ボリュームマネージャー RAID-1 ボリュームへの移行例(コマンド行インタフェース)

Solaris Live Upgrade では、RAID-1 ボリューム (ミラー) 上に新しいブート環境を作成できます。現在のブート環境のファイルシステムは、次のいずれかです。

- 物理ストレージデバイス
- Solaris ボリュームマネージャーの制御下の RAID-1 ボリューム
- Veritas VXFS 制御下のボリューム

ただし、新しいブート環境のターゲットは、Solaris ボリュームマネージャー RAID-1 ボリュームでなければなりません。たとえば、ルート(/) ファイルシステムのコピー用に指定するスライスは、/dev/vx/dsk/rootvol となります。rootvol はルート(/) ファイルシステムを含むボリュームです。

この例では、現在のブート環境のルート(/) ファイルシステムは Solaris ボリュームマネージャーボリューム以外のボリューム上にあります。新しいブート環境では、Solaris ボリュームマネージャー RAID-1 ボリュームである `c0t2d0s0` 上にルート(/) ファイルシステムが作成されます。lucreate コマンドは、現在のボリュームを Solaris ボリュームマネージャーボリュームに移行させるコマンドです。新しいブート環境の名前は `svm_be` です。lustatus コマンドを使用すると、新しいブート環境のアクティブ化とリブートの準備ができていかどうかわかります。ブート環境がアクティブ化され、現在のブート環境になります。

```
lucreate -n svm_be -m /:/dev/md/dsk/d1:mirror,ufs \
-m /:/dev/dsk/c0t2d0s0:attach
lustatus
luactivate svm_be
lustatus
init 6
```

## 空のブート環境を作成して **Solaris** フラッシュアーカイブをインストールする例(コマンド行インタフェース)

次の手順を3段階に分けて説明します。

- 空のブート環境の作成
- アーカイブのインストール
- ブート環境をアクティブにし、現在実行中のブート環境にする

lucreate コマンドは、アクティブブート環境内のファイルシステムに基づいてブート環境を作成します。lucreate コマンドに `-s` オプションを指定して実行すると、空のブー

ト環境を短時間で作成できます。スライスは、指定のファイルシステム用に予約されていますが、ファイルシステムはコピーされません。このブート環境は、名前が付けられてはいますが、実際には、Solaris フラッシュアーカイブがインストールされる時にはじめて作成されることとなります。空のブート環境にアーカイブがインストールされると、ファイルシステムは予約されたスライスにインストールされます。その後、ブート環境をアクティブ化します。

## 空のブート環境を作成する方法

最初の手順で、空のブート環境を作成します。指定されたファイルシステム用にスライスが予約されますが、現在のブート環境からファイルシステムがコピーされることはありません。新しいブート環境の名前は `second_disk` です。

```
lucreate -s - -m /:/dev/dsk/c0t1d0s0:ufs \
-n second_disk
```

これで、ブート環境に Solaris フラッシュアーカイブを格納する準備ができました。

図 9-3 は、空のブート環境の作成の様子を示しています。

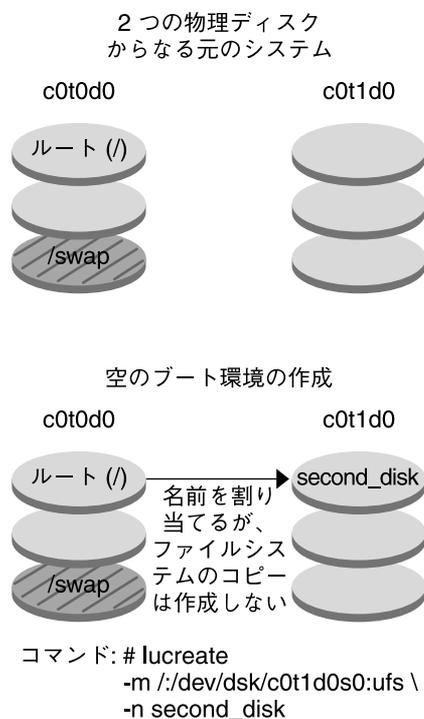


図 9-3 空のブート環境の作成

## 新しいブート環境へ Solaris フラッシュアーカイブをインストールする方法

2 番目の手順では、前の例で作成した `second_disk` ブート環境に、アーカイブをインストールします。アーカイブはローカルシステムに存在します。-s および -a オプションで指定するオペレーティングシステムのバージョンは、どちらも Solaris 10 11/06 リリースです。アーカイブの名前は `Solaris_10.flar` です。

```
luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/10.flar
```

このようにして、ブート環境をアクティブにする準備が整います。

## 新しいブート環境をアクティブ化する方法

最後の手順では、`luactivate` コマンドを使用して、`second_disk` ブート環境をブート可能な状態にします。続いてシステムをリブートします。これで `second_disk` がアクティブブート環境になります。

```
luactivate second_disk
init 6
```

- 空のブート環境の詳しい作成手順については、83 ページの「Solaris フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インターフェース)」を参照してください。
- Solaris フラッシュアーカイブの詳しい作成手順については、『Solaris 10 11/06 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第 3 章「Solaris フラッシュアーカイブの作成 (作業)」を参照してください。
- ブート環境をアクティブ化したり、元のブート環境にフォールバックしたりする詳細な手順については、第 6 章を参照してください。

## Solaris Live Upgrade によるアップグレードの例 (キャラクターインターフェース)

この例では、Solaris 9 リリースを実行しているシステム上に新しいブート環境を作成します。この新しいブート環境を Solaris 10 6/06 リリースにアップグレードします。続いて、アップグレードしたブート環境をアクティブにします。

## アクティブブート環境で Solaris Live Upgrade をインストールする方法

1. Solaris Operating System DVD または Solaris SOFTWARE - 2 CD を挿入します。
2. インストーラを実行します。
  - Solaris Operating System DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。
    - SPARC システムの場合:

```
cd /cdrom/cdrom0/S0/Solaris_10/Tools/Installers
./liveupgrade20
```

Solaris インストールプログラムの GUI が表示されます。
    - x86 システムの場合:

```
cd /cdrom/cdrom0/Solaris_10/Tools/Installers
./liveupgrade20
```

Solaris インストールプログラムの GUI が表示されます。
  - Solaris SOFTWARE - 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```

Solaris インストールプログラムの GUI が表示されます。
3. 「インストール形式の選択 (Select Type of Install)」パネルで「カスタム (Custom)」をクリックします。
4. 「ロケールの選択 (Locale Selection)」パネルで、インストールする言語をクリックします。
5. インストールするソフトウェアを選択します。
  - DVD の場合、「コンポーネントの選択 (Component Selection)」パネルの「次へ (Next)」をクリックしてパッケージをインストールします。
  - CD の場合、「製品の選択 (Product Selection)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (Default Install)」をクリックします。また、ソフトウェアの選択を解除するには、ほかの製品をクリックします。
6. Solaris インストールプログラムの指示に従って、ソフトウェアをインストールします。

## 必要なパッチをインストールするには

説明	参照先
<p>注意 - Solaris Live Upgrade を正しく操作するためには、指定の OS バージョン用の特定のパッチリビジョンのセットがインストールされている必要があります。Solaris Live Upgrade をインストールまたは実行する前に、これらのパッチをインストールする必要があります。</p> <p><b>x86 のみ - Solaris 10 1/06</b> 以降のリリースでは、このパッチのセットがインストールされていないと、Solaris Live Upgrade が失敗し、次のエラーメッセージが表示されることがあります。次のエラーメッセージが表示されなくても、必要なパッチがインストールされていない場合があります。Solaris Live Upgrade のインストールを試みる前に、SunSolve の infodoc に記載されたすべてのパッチがすでにインストール済みであることを必ず確認してください。</p> <pre>ERROR: Cannot find or is not executable: &lt;/sbin/biosdev&gt;. ERROR: One or more patches required by Live Upgrade has not been installed.</pre> <p>infodoc 72099 に記載されたパッチは、随時変更される可能性があります。これらのパッチにより、Solaris Live Upgrade の欠陥が修正される可能性があると同時に、Solaris Live Upgrade が依存するコンポーネントの欠陥も修正される可能性があります。Solaris Live Upgrade で問題が発生した場合は、最新の Solaris Live Upgrade パッチがインストールされていることを確認してください。</p>	<p><a href="http://sunsolve.sun.com">http://sunsolve.sun.com</a> で最新のパッチリストを確認してください。SunSolve の Web サイトで、infodoc 72099 を検索してください。</p>
<p>Solaris 8、または Solaris 9 OS を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。</p>	<p>Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、Java 2 Runtime Environment 推奨パッチクラスタをインストールします。このパッチクラスタは <a href="http://sunsolve.sun.com">http://sunsolve.sun.com</a> から入手できます。</p>

次の手順に従って必要なパッチをインストールします。

SunSolve の Web サイトから、パッチ一覧を取得します。

```
patchadd /net/server/export/patches
init 6
```

## ブート環境を作成する方法

この例では、ソースブート環境の名前は `c0t4d0s0` です。ルート (`/`) ファイルシステムを新しいブート環境にコピーします。また、ソースブート環境のスワップスライスは共有せず、新しいスワップスライスを作成します。

1. スーパーユーザーになるか、同等の役割になります。
2. キャラクターユーザーインタフェースを表示します。

```
/usr/sbin/lu
```

Solaris Live Upgrade のメインメニューが表示されます。

3. メインメニューから「Create」を選択します。

```
Name of Current Boot Environment: c0t4d0s0
Name of New Boot Environment: c0t15d0s0
```

4. F3 を押します。  
「Configuration」メニューが表示されます。
5. スライスを選択するために、「Configuration」メニューで F2 を押します。  
「Choices」メニューが表示されます。
6. ディスク `c0t15d0` からルート (`/`) ファイルシステム用としてスライス 0 を選択します。
7. 構成メニューで、分割するスワップスライスを選択して `c0t15d0` 上にスワップ用の新しいスライスを作成します。
8. スワップ用のスライスを選択するために、F2 を押します。「Choices」メニューが表示されます。
9. 新しいスワップスライスとして、ディスク `c0t15d0` からスライス 1 を選択します。
10. F3 を押して新しいブート環境を作成します。

## 非アクティブブート環境をアップグレードする方法

次に、この新しいブート環境をアップグレードします。アップグレードに使用する新しいバージョンのオペレーティングシステムは、ネットワークイメージから取得します。

1. メインメニューから「Upgrade」を選択します。

```
Name of New Boot Environment: c0t15d0s0
Package Media: /net/ins3-svr/export/Solaris_10/combined.solaris_wos
```

2. F3 を押します。

## 非アクティブブート環境をアクティブにする方法

`c0t15d0s0` ブート環境をブート可能にします。続いてシステムをリブートします。これで `c0t15d0s0` がアクティブブート環境になります。 `c0t4d0s0` ブート環境は非アクティブになります。

1. メインメニューから「Activate」を選択します。

```
Name of Boot Environment: c0t15d0s0
Do you want to force a Live Upgrade sync operations: no
```

2. F3 を押します。
3. Return キーを押します。
4. 次のコマンドを入力します。

```
init 6
```

フォールバックが必要な場合は、前述の例のコマンド行による作業を行います。次を参照してください。165 ページの「(省略可能) ソースブート環境へフォールバックする方法」。

# ◆◆◆ 第 10 章

## Solaris Live Upgrade (コマンドリファレンス)

---

次の表に、コマンド行で入力できるコマンドを示します。Solaris Live Upgrade には、次の表に示すすべてのコマンド行ユーティリティーのマニュアルページが含まれています。

### Solaris Live Upgrade のコマンド

作業	コマンド名
非アクティブブート環境をアクティブにします。	luactivate(1M)
スケジュールされた処理(コピーまたは作成)を取り消します。	lucancel(1M)
アクティブブート環境を非アクティブブート環境と比較します。	lucompare(1M)
非アクティブブート環境を更新するためにファイルシステムをコピーし直します。	lumake(1M)
ブート環境を作成します。	lucreate(1M)
アクティブブート環境に名前を付けます。	lucurr(1M)
ブート環境を削除します。	ludelete(1M)
ブート環境の名前に記述を追加します。	ludesc(1M)
各ブート環境のクリティカルファイルシステムを表示します。	lufslst(1M)

作業	コマンド名
ブート環境内のすべてのファイルシステムをマウントできるようにします。このコマンドを使用すると、ブート環境がアクティブでない時にそのブート環境内のファイルを変更できます。	lumount(1M)
ブート環境の名前を変更します。	lurename(1M)
すべてのブート環境のステータスを表示します。	lustatus(1M)
ブート環境に存在するすべてのファイルシステムのマウントを解除します。このコマンドを使用すると、ブート環境がアクティブでない時にそのブート環境内のファイルを変更できます。	luumount(1M)
非アクティブブート環境上の OS をアップグレードするか、あるいは非アクティブブート環境上にフラッシュアーカイブをインストールします。	luupgrade(1M)

パート II

## 付録

このパートでは、リファレンス情報について説明します。



## 問題発生時の解決方法 (作業)

---

この章では、Solaris 10 11/06 ソフトウェアのインストール時に発生する可能性のあるエラーメッセージと一般的な問題の一覧を示し、それぞれの問題の解決方法を示します。まず、次のリストを使用して、インストールプロセスのどこで問題が発生したか確認してください。

- 183 ページの「ネットワークインストールの設定に関する問題」
- 184 ページの「システムのブートに関する問題」
- 190 ページの「Solaris OS の初期インストール」
- 192 ページの「Solaris OS のアップグレード」

---

注-この付録で「ブート可能なメディア」と記載されている場合、これはインストールプログラムおよび JumpStart インストールを意味します。

---

## ネットワークインストールの設定に関する問題

Unknown client "*host\_name*"

原因: `add_install_client` コマンドの *host\_name* 引数は、ネームサービス内のホストではありません。

対処方法: ホスト *host\_name* をネームサービスに追加し、`add_install_client` コマンドを実行し直してください。

Error: <system name> does not exist in the NIS ethers map

Add it, and rerun the `add_install_client` command

説明: `add_install_client` コマンドを実行すると、上記のエラーが出てコマンドが失敗します。

原因: インストールサーバーに追加しようとしているクライアントが、サーバーの `/etc/ethers` ファイルに存在しません。

対処方法: 必要な情報をインストールサーバー上の `/etc/ethers` ファイルに追加し、`add_install_client` コマンドを実行し直してください。

1. スーパーユーザーになるか、同等の役割になります。
2. このクライアント上で `ethers` アドレスを見つけます。

```
ifconfig -a grep ethers
ether 8:0:20:b3:39:1d
```

3. インストールサーバー上で、エディタを使って `/etc/ethers` ファイルを開きます。そのアドレスをリストに追加します。
4. この例のように、クライアント上で `add_install_client` をもう一度実行します。

```
./add_install_client bluegill sun4u
```

## システムのブートに関する問題

### メディアからのブート時のエラーメッセージ

`le0: No carrier - transceiver cable problem`

原因: システムがネットワークに接続されていません。

対処方法: ネットワークに接続せずに使用しているシステムの場合は、このメッセージは無視してください。ネットワークに接続されているシステムの場合は、Ethernet が正しく接続されているかどうか確認してください。

`The file just loaded does not appear to be executable`

原因: ブート用の適切な媒体が見つかりません。

対処方法: インストールサーバーからネットワークを介して Solaris 10 11/06 ソフトウェアをインストールするように正しく設定されているか確認します。たとえば、次のような確認を行います。

- Solaris Operating System DVD または Solaris SOFTWARE CD のイメージをインストールサーバーにコピーした場合は、設定時にシステムのプラットフォームグループを正しく指定したかどうかを確認します。
- DVD または CD メディアを使用する場合は、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD がインストールサーバー上にマウントされていてアクセスできることを確認します。

`boot: cannot open <filename> (SPARC システムのみ)`

原因: PROM の `boot - file` の値を明示的に指定したときに発生するエラーです。

---

注 `-filename` は、対象となるファイルの名前です。

---

対処方法: 次の手順を実行します。

- PROM の `boot-file` の値を (無指定) に設定変更します。
- `diag-switch` が `off` と `true` に設定されているか確認します。

#### Can't boot from file/device

原因: インストールメディアがブート可能なメディアを見つけることができません。

対処方法: 次の条件が満たされているか確認します。

- DVD-ROM または CD-ROM ドライブがシステムに適切に取り付けられ、電源が入っている。
- Solaris Operating System DVD または Solaris SOFTWARE - 1 CD がドライブに挿入されている。
- ディスクに傷や埃が付いていない。

#### WARNING: clock gained xxx days -- CHECK AND RESET DATE! (SPARC システムのみ)

説明: これは参考情報です。

対処方法: メッセージは無視して、インストールを継続してください。

#### Not a UFS file system (x86 システムのみ)

原因: Solaris 10 11/06 ソフトウェアをインストールしたとき (Solaris インストールプログラムまたはカスタム JumpStart を使って)、ブートドライブを選択しませんでした。BIOS を編集してシステムをブートする必要があります。

対処方法: BIOS を選択してブートします。詳細は、BIOS のマニュアルを参照してください。

## メディアからのブート時の一般的な問題

システムがブートしない。

説明: 初めてカスタム JumpStart サーバーを設定する場合、エラーメッセージを返さないブート問題が発生することがあります。システムについての情報およびシステムがどのようにブートするかを調べるには、`-v` オプションを指定してブートコマンドを実行してください。`-v` オプションを使用すると、ブートコマンドは画面に詳しいデバッグ情報を表示します。

---

注 - このフラグを指定しなくてもメッセージは出力されますが、システムログファイルが出力先となります。詳細については、`syslogd(1M)` を参照してください。

---

対処方法: SPARC システムの場合、`ok` プロンプトで次のコマンドを入力します。

**ok boot net -v - install**

Toshiba SD-M 1401 DVD-ROM が搭載されたシステムで DVD メディアからのブートが失敗する

説明: 使用しているシステムにファームウェアバージョン 1007 の Toshiba SD-M1401 DVD-ROM が搭載されている場合、システムは Solaris Operating System DVD からブートできません。

対処方法: 111649-03 以降のパッチを適用して Toshiba SD-M1401 DVD-ROM ドライブのファームウェアを更新します。このパッチ 111649-03 は [sunsolve.sun.com](http://sunsolve.sun.com) から入手できます。

メモリー増設用以外の PC カードを挿入すると、システムがハングまたはパニックを起こす。(x86 システムのみ)

原因: メモリー増設用以外の PC カードは、ほかのデバイスが使用するのと同じメモリーリソースを使用できません。

対処方法: この問題を解決するには、PC カードのマニュアルを参照してアドレス範囲を確認してください。

システムがプロンプトを出す前にハングする。(x86 システムのみ)

対処方法: サポートされていないハードウェアです。ハードウェアのマニュアルを参照してください。

## ネットワークからのブート時のエラーメッセージ

WARNING: getfile: RPC failed: error 5 (RPC Timed out).

説明: インストールクライアントのブート要求に対して、ネットワーク上の複数のサーバーが応答したときに発生するエラーです。インストールクライアントの接続先のブートサーバーが間違っているため、インストールは停止します。次の原因が考えられます。

原因: 1 このインストールクライアントが登録された `/etc/bootparams` ファイルが複数のサーバーに存在する可能性があります。

対処方法: 1 ネットワーク上の複数のサーバーの `/etc/bootparams` エントリにインストールクライアントが登録されていないか調べます。複数のサーバーに登録がされている場合は、インストールに使用するインストールサーバー（またはブートサーバー）以外のサーバーの `/etc/bootparams` ファイルから登録を削除します。

原因: 2 複数の `/tftpboot` または `/rplboot` ディレクトリにこのインストールクライアントが登録されている可能性があります。

対処方法: 2 ネットワーク上の複数のサーバーの `/tftpboot` または `/rplboot` ディレクトリにインストールクライアントが登録されていないか調べます。複数のサーバーに登

録されている場合は、インストールに使用するインストールサーバー（またはブートサーバー）以外のサーバーの `/tftpboot` または `/rplboot` ディレクトリから、クライアントの登録を削除します。

原因: 3 あるサーバーの `/etc/bootparams` ファイルにこのインストールクライアントが登録されており、別のサーバーの `/etc/bootparams` ファイルで、すべてのシステムがプロファイルサーバーにアクセスできるように記述されている可能性があります。次に示すようにすべてのシステムがプロファイルサーバーにアクセスできるように記述しています。

```
* install_config=profile_server:path
```

このエラーは、NIS または NIS+ の `bootparams` テーブルにこのような行が存在していても発生します。

対処方法: 3 ワイルドカードエントリがネームサービスの `bootparams` マップまたはテーブル (`* install_config=` など) にある場合は、そのエントリを削除し、ブートサーバーの `/etc/bootparams` ファイルに追加します。

**No network boot server. Unable to install the system. See installation instructions.**  
(SPARC システムのみ)

原因: このエラーは、ネットワークからインストールしようとしているシステムで発生します。このシステムは、適切に設定されていません。

対処方法: ネットワークを介してインストールするようにシステムが適切に設定されているか確認します。『Solaris 10 11/06 インストールガイド(ネットワークインストール)』の「CD イメージを使用してネットワークからインストールするシステムの追加」を参照してください。

**prom\_panic: Could not mount file system** (SPARC システムのみ)

原因: このエラーはネットワークから Solaris をインストールしようとしてブートソフトウェアが次のものを見つけられない場合に発生します。

- Solaris Operating System DVD またはインストールサーバー上の Solaris Operating System DVD イメージコピー
- Solaris SOFTWARE - 1 CD またはインストールサーバー上の Solaris SOFTWARE - 1 CD イメージコピー

対処方法: インストール用のソフトウェアがマウントされ共有されるように設定してあることを確認します。

- インストールサーバーの DVD-ROM または CD-ROM ドライブから Solaris をインストールする場合は、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD が CD-ROM ドライブに挿入されてマウントされていること、および `/etc/dfs/dfstab` ファイルで共有されるように設定してあることを確認します。

- インストールサーバーのディスク上にある Solaris Operating System DVD イメージまたは Solaris SOFTWARE - 1 CD イメージのコピーからインストールする場合は、そのコピーのディレクトリパスが /etc/dfs/dfstab ファイル内で共有されていることを確認します。

Timeout waiting for ARP/RARP packet...(SPARC システムのみ)

原因:1 クライアントはネットワークを介してブートしようとしています、認識してくれるシステムを見つけることができません。

対処方法:1 システムのホスト名が NIS または NIS+ のネームサービスに登録されていることを確認します。また、ブートサーバーの /etc/nsswitch.conf ファイル内の bootparams の検索順序を確認します。

たとえば、/etc/nsswitch.conf ファイル内にある次の行は、JumpStart または Solaris インストールプログラムが最初に NIS マップから bootparams 情報を探すことを示しています。ここで情報が見つからない場合、インストーラはブートサーバーの /etc/bootparams ファイルを調べます。

```
bootparams: nis files
```

原因:2 クライアントの Ethernet アドレスが不正です。

対処方法:2 インストールサーバーの /etc/ethers ファイルにあるクライアントの Ethernet アドレスが正しいことを確認します。

原因:3 カスタム JumpStart インストールでは、特定のサーバーをインストールサーバーとして使用するようプラットフォームグループを add\_install\_client コマンドによって指定します。add\_install\_client を使用する際に不正な構成値を使用すると、この問題が発生します。たとえば、インストールするマシンが sun4u であるのに誤って i86pc と指定した場合などが考えられます。

対処方法:3 正しいアーキテクチャー値を使用して add\_install\_client を実行し直します。

ip: joining multicasts failed on tr0 - will use link layer broadcasts for multicast (x86 システムのみ)

原因:このエラーメッセージは、トークンリングカードを使ってシステムをブートしたときに表示されます。Ethernet のマルチキャストとトークンリングのマルチキャストの動作は異なります。ドライバはこのエラーメッセージを返して、マルチキャストアドレスが無効なことを知らせます。

対処方法:このエラーメッセージは無視してください。マルチキャストがうまく動作しななければ、IP は代わりにレイヤーブロードキャストを使用し、インストールは失敗しません。

Requesting Internet address for Ethernet\_Address (x86 システムのみ)

原因:クライアントはネットワークを介してブートしようとしています、認識してくれるシステムを見つけることができません。

対処方法: システムのホスト名がネームサービスに登録されていることを確認します。システムのホスト名がNISまたはNIS+のネームサービスに登録されているのに、システムがこのエラーメッセージを表示し続ける場合は、リブートしてください。

RPC: Timed out No bootparams (whoami) server responding; still trying... (x86 システムのみ)

原因: クライアントはネットワークからブートしようとしていますが、インストールサーバー上の `/etc/bootparams` ファイルにエントリを持つシステムを見つけることができません。

対処方法: インストールサーバー上で `add_install_client` を実行します。これにより `/etc/bootparams` ファイルに適切なエントリが追加され、クライアントがネットワークからブートできるようになります。

Still trying to find a RPL server... (x86 システムのみ)

原因: システムはネットワークからブートしようとしていますが、サーバーではこのシステムをブートするように設定されていません。

対処方法: インストールサーバー上で、インストールするシステム用に `add_install_client` を実行します。 `add_install_client` コマンドは、必要なネットワークブートプログラムを含む `/rplboot` ディレクトリを設定します。

CLIENT MAC ADDR: FF FF FF FF FF FF (DHCP によるネットワークインストールのみ)

原因: DHCP サーバーが正しく構成されていません。このエラーは、DHCP マネージャ内でオプションやマクロが正しく定義されていない場合に発生する可能性があります。

対処方法: DHCP マネージャで、オプションおよびマクロが正しく定義されていることを確認します。ルーターオプションが定義されており、その値がネットワークインストールで使用するサブネットを正しく表していることを確認します。

## ネットワークからのブート時の一般的な問題

システムはネットワークを介してブートされるが、指定したインストールサーバー以外のシステムからブートされる。

原因: このクライアントが登録された `/etc/bootparams` エントリと `/etc/ethers` エントリが別のシステム上に存在します。

対処方法: ネームサーバー上で、インストールするシステムの `/etc/bootparams` エントリを更新します。このエントリは、次の構文に従う必要があります。

```
install_system root=boot_server:path install=install_server:path
```

また、サブネット内で複数のサーバーの `bootparams` ファイルにインストールクライアントが登録されていないか確認します。

システムがネットワークからブートしない (DHCP によるネットワークインストールのみ)。

原因: DHCP サーバーが正しく構成されていません。このエラーは、システムが DHCP サーバーのインストールクライアントとして構成されていない場合に発生することがあります。

対処方法: DHCP マネージャーソフトウェアで、クライアントシステムのインストールオプションとマクロが定義されていることを確認します。詳細については、『Solaris 10 11/06 インストールガイド (ネットワークインストール)』の「DHCP サービスによるシステム構成情報の事前設定 (作業)」を参照してください。

## Solaris OS の初期インストール

初期インストールが失敗する

対処方法: Solaris のインストールが失敗する場合、インストールを再実行する必要があります。インストールを再実行するには、Solaris Operating System DVD、Solaris SOFTWARE - 1 CD、またはネットワークを利用してシステムをブートする必要があります。

Solaris ソフトウェアが部分的にインストールされたあとでは、このソフトウェアのインストールを解除することはできません。バックアップからシステムを復元するか、Solaris インストールの処理をもう一度行う必要があります。

`/cdrom/cdrom0/SUNWxxx/reloc.cpio: Broken pipe`

説明: このエラーメッセージは参考情報であり、インストールには影響しません。パイプへ書き込みをしたときに読み取りプロセスが存在しないと、この状況が発生します。

対処方法: メッセージは無視して、インストールを継続してください。

**WARNING: CHANGE DEFAULT BOOT DEVICE (x86 システムのみ)**

原因: これは参考情報です。システムの BIOS に設定されているデフォルトブートデバイスが、ブート時に Solaris Device Configuration Assistant (デバイス構成用補助) を必要とするように設定されている可能性があります。

対処方法: インストールを続行します。Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクを必要としないデバイスに Solaris ソフトウェアをインストールし終わったら、必要に応じて、BIOS に指定されたシステムのデフォルトのブートデバイスを変更します。

---

x86のみ-locale キーワードを使用して初期インストール用のカスタム JumpStart プロファイル进行测试する場合は、`pfinstall -D` コマンドでプロファイル进行测试すると失敗します。回避方法については、192 ページの「Solaris OS のアップグレード」セクションのエラーメッセージ「could not select locale」を参照してください。

---

## ▼ x86: IDE ディスクの不良ブロックの検査

IDE ディスクドライブは、Solaris ソフトウェアがサポートするほかのドライブのように、不良ブロックを自動的に無効にしません。IDE ディスク上に Solaris をインストールする前に、ディスクを検査することをお勧めします。IDE ディスクの検査を行うには、次の手順に従います。

- 1 スーパーユーザーになるか、同等の役割になります。  
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 インストールメディアに応じた方法でブートします。
- 3 インストールタイプの選択を求めるプロンプトが表示されたら、オプション 6 「Single user shell」を選択します。
- 4 `format (1M)` プログラムを起動します。  

```
format
```
- 5 ディスク面の検査をする IDE ディスクドライブを指定します。  

```
cxdy
```

`cx`    コントローラ番号  
`dy`    デバイス番号
- 6 `fdisk` パーティションが存在するかどうかを確認します。
  - Solaris `fdisk` パーティションが存在する場合、手順 7 へ進みます。
  - Solaris `fdisk` パーティションが存在しない場合、`fdisk` コマンドを使用してディスク上に Solaris パーティションを作成します。

```
format> fdisk
```
- 7 次のように入力して、表面解析を開始します。  

```
format> analyze
```

- 8 次のように入力して、現在の設定を確認します。

```
analyze> config
```

- 9 (省略可能) 次のように入力して、設定を変更します。

```
analyze> setup
```

- 10 次のように入力して、不正ブロックを探します。

```
analyze> type_of_surface_analysis
```

`type_of_surface_analysis` read、write、または compare を指定します。

`format` が不良ブロックを発見すると、それらの再マッピングを実行します。

- 11 次のように入力して、解析を終了します。

```
analyze> quit
```

- 12 再マッピングするブロックを指定するかどうか決定します。

- しない場合は、手順 13 へ進みます。
- する場合は、次のコマンドを入力します。

```
format> repair
```

- 13 次のように入力して、`format` プログラムを終了します。

```
quit
```

- 14 次のコマンドを入力して、マルチユーザーモードでメディアを再起動します。

```
exit
```

## Solaris OS のアップグレード

### アップグレード時のエラーメッセージ

No upgradable disks

原因: /etc/vfstab ファイルのスワップエントリが原因でアップグレードに失敗しました。

対処方法: /etc/vfstab ファイルの次の行をコメントにします。

- アップグレードしないディスク上のスワップファイルとスライスを指定している行
- 存在しないスワップファイルを指定している行

- 使用していないスワップスライスを指定している行

usr/bin/bzcat not found

原因: パッチクラスタが必要なために Solaris Live Upgrade が失敗しています。

対処方法: Solaris Live Upgrade のインストールでパッチを使用する必要が生じました。  
<http://sunsolve.sun.com> で最新のパッチリストを確認してください。SunSolve の Web サイトで、infodoc 72099 を検索してください。

Upgradeable Solaris root devices were found, however, no suitable partitions to hold the Solaris install software were found. Upgrading using the Solaris Installer is not possible. It might be possible to upgrade using the Solaris Software 1 CDRom. (x86 システムのみ)

原因: 十分な容量がないため、Solaris SOFTWARE - 1 CD でアップグレードできません。

対処方法: アップグレードするには、512M バイト以上のスワップスライスを作成するか、別のアップグレード方法 (Solaris Operating System DVD の Solaris インストール、ネットインストールイメージ、JumpStart など) を選択します。

ERROR: Could not select locale (x86 システムのみ)

原因: pfinstall -D コマンドを使用して JumpStart プロファイルをドライランテストするとき、次のような条件下ではテストが失敗します。

- プロファイルに locale キーワードが含まれている。
- GRUB ソフトウェアが含まれているリリースをテストしている。**Solaris 10 1/06** 以降のリリースでは、GRUB ブートローダーにより、システムにインストールされているさまざまなオペレーティングシステムを GRUB メニューで簡単にブートできます。

GRUB ソフトウェアの導入に伴い、ミニルートは圧縮されています。ソフトウェアでは、圧縮されたミニルートからロケールのリストを見つけることができなくなりました。ミニルートは最小の Solaris ルート (/) ファイルシステムであり、Solaris インストールメディアに収録されています。

対処方法: 次の手順を実行します。次の値を使用してください。

- MEDIA\_DIR は /cdrom/cdrom0/
- MINIRoot\_DIR は \$MEDIA\_DIR /Solaris\_10/Tools/Boot
- MINIRoot\_ARCHIVE は \$MEDIA\_DIR/boot/x86.miniroot
- TEMP\_FILE\_NAME は /tmp/test

1. スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ミニルートアーカイブを圧縮解除します。

```
/usr/bin/gzcat $MINIRoot_ARCHIVE > $TEMP_FILE_NAME
```

3. lofiadm コマンドを使用して、ミニルートデバイスを作成します。

```
LOFI_DEVICE=/usr/sbin/lofiadm -a $TEMP_FILE_NAME
echo $LOFI_DEVICE
/dev/lofi/1
```

4. lofi コマンドを使用して、ミニルートディレクトリの下にミニルートをマウントします。

```
/usr/sbin/mount -F ufs $LOFI_DEVICE $MINIRoot_DIR
```

5. プロファイルをテストする

```
/usr/sbin/install.d/pfinstall -D -c $MEDIA_DIR $path-to-jumpstart_profile
```

6. テストが完了したら、lofi デバイスのマウントを解除します。

```
umount $LOFI_DEVICE
```

7. lofi デバイスを削除します。

```
lofiadm -d $TEMP_FILE_NAME
```

## アップグレード時の一般的な問題

システム上にアップグレード可能なバージョンの Solaris ソフトウェアが存在するにもかかわらず、アップグレードオプションが提供されない。

原因: 1 /var/sadm ディレクトリがシンボリックリンクであるか、別のファイルシステムからマウントされたディレクトリです。

対処方法: 1 /var/sadm ディレクトリをルート (/) または /var ファイルシステムに移動します。

原因: 2 /var/sadm/softinfo/INST\_RELEASE ファイルが存在しません。

対処方法: 2 次の形式で新しく INST\_RELEASE ファイルを作成します。

```
OS=Solaris
VERSION=x
REV=0
```

x  
システム上の Solaris ソフトウェアのバージョン

原因: 3 /var/sadm/softinfo に SUNWusr が存在しません。

対処方法: 3 初期インストールを行う必要があります。この Solaris ソフトウェアはアップグレードできません。

md ドライバの停止または初期化に失敗する

対処方法: 次の手順を実行します。

- ファイルシステムが RAID-1 ボリュームでない場合は、`vsftab` ファイル内でコメントにします。
- ファイルシステムが RAID-1 ボリュームであれば、ミラーを解除し、インストールし直します。ミラー化の解除については、『Solaris ボリュームマネージャの管理』の「RAID-1 ボリュームの削除(ミラー化の解除)」を参照してください。

Solaris インストールプログラムがファイルシステムをマウントできないため、アップグレードに失敗する。

原因: アップグレード時に、スクリプトは、アップグレード対象のルート (*/*) ファイルシステム上に、システムの `/etc/vfstab` ファイルに記載されているすべてのファイルシステムをマウントしようとします。インストールプログラムがファイルシステムをマウントできない場合、失敗して終了します。

対処方法: システムの `/etc/vfstab` ファイル内のすべてのファイルシステムがマウントできることを確認します。`/etc/vfstab` ファイル内のマウントできない、あるいは問題の原因になっている可能性があるファイルシステムは、すべてコメントにします。Solaris インストールプログラムはアップグレード中、コメントにしたファイルシステムをマウントしません。アップグレードされるソフトウェアを含む、システムベースのファイルシステム(たとえば `/usr`) はコメントにできません。

アップグレードが失敗する

説明: システムにアップグレードに対応できるだけの十分なディスク容量がありません。

原因: 『Solaris 10 11/06 インストールガイド(インストールとアップグレードの計画)』の「ディスク容量の再配置を伴うアップグレード」を参照してディスク容量に問題がないかを確認し、自動配置機能による領域の再配置を行わずに解決できるかどうかを調べます。

RAID-1 ボリュームのルート (*/*) ファイルシステムのアップグレードに関連する問題

対処方法: ルート (*/*) ファイルシステムが Solaris ボリュームマネージャの RAID-1 ボリュームである場合に、アップグレードの問題が発生するときは、『Solaris ボリュームマネージャの管理』の第 25 章「Solaris ボリュームマネージャのトラブルシューティング(作業)」を参照してください。

## ▼ 問題発生後にアップグレードを継続する方法

アップグレードに失敗し、システムをブートできない場合があります。このような状況は、電源の故障やネットワーク接続の障害などが発生した場合に起こる可能性があります。制御できない場合に発生します。

- 1 **Solaris Operating System DVD**、**Solaris SOFTWARE-1 CD**、またはネットワークを利用してシステムをリブートします。
- 2 インストール用のアップグレードオプションを選択します。  
Solaris インストールプログラムは、システムが部分的にアップグレードされているか判断し、アップグレードを継続します。

## x86: GRUB を使用する場合の Solaris Live Upgrade に関する問題

x86 システムで Solaris Live Upgrade と GRUB ブートローダーを使用すると次のようなエラーが発生する可能性があります。

```
ERROR: The media product tools installation directory path-to-installation-directory does not exist.
```

```
ERROR: The media dirctory does not contain an operating system upgrade image.
```

説明: これらのエラーメッセージは、新しいブート環境をアップグレードするために `luupgrade` コマンドを使用するときに発生します。

原因: 古いバージョンの Solaris Live Upgrade が使用されています。システムにインストールした Solaris Live Upgrade パッケージは、メディアおよびメディアに記録されているリリースと互換性がありません。

対処方法: Solaris Live Upgrade パッケージは、常にアップブレード先のリリースのものを使用してください。

例: 次の例のエラーメッセージは、システムの Solaris Live Upgrade パッケージのバージョンがメディアのパッケージのバージョンと異なることを示しています。

```
luupgrade -u -n s10u1 -s /mnt
 Validating the contents of the media </mnt>.
 The media is a standard Solaris media.
 ERROR: The media product tools installation directory
</mnt/Solaris_10/Tools/Boot/usr/sbin/install.d/install_config> does
not exist.
 ERROR: The media </mnt> does not contain an operating system upgrade
image.
```

```
ERROR: Cannot find or is not executable: </sbin/biosdev>.
```

```
ERROR: One or more patches required by Solaris Live Upgrade has not been installed.
```

原因: Solaris Live Upgrade で必要とされる 1 つ以上のパッチが、システムにインストールされていません。このエラーメッセージでは、欠落しているすべてのパッチを認識しているわけではありません。

対処方法: Solaris Live Upgrade を使用する前に、必要なパッチすべてを必ずインストールしてください。 <http://sunsolve.sun.com> を参照して、最新のパッチリストを使用しているかどうか確認してください。 SunSolve の Web サイトで、infodoc 72099 を検索してください。

ERROR: Device mapping command </sbin/biosdev> failed. Please reboot and try again.  
原因: 1 Solaris Live Upgrade が、以前の管理作業が原因でデバイスをマップできません。

対処方法: 1 システムをリブートして、もう一度 Solaris Live Upgrade を実行します

原因: 2 システムをリブートしても同じエラーメッセージが表示される場合は、2 つ以上の同一ディスクがあります。デバイスのマッピングコマンドがそれらのディスクを区別できません。

対処方法: 2 ディスクの一方に、新しいダミーの fdisk パーティションを作成します。fdisk(1M) のマニュアルページを参照してください。そのあとで、システムをリブートします。

Cannot delete the boot environment that contains the GRUB menu

原因: Solaris Live Upgrade には、ブート環境に GRUB メニューが含まれる場合はブート環境を削除できないという制限があります。

対処方法: lumake(1M) コマンドまたは luupgrade(1M) コマンドを使用してブート環境を再使用します。

The file system containing the GRUB menu was accidentally remade. However, the disk has the same slices as before. For example, the disk was not re-sliced.

原因: GRUB メニューを含むファイルシステムは、システムをブート可能な状態に維持するために不可欠です。Solaris Live Upgrade コマンドは、GRUB メニューを破棄しません。ただし、Solaris Live Upgrade コマンド以外のコマンドを使用して GRUB メニューのあるファイルシステムを誤って再作成または破棄すると、回復ソフトウェアは GRUB メニューの再インストールを試みます。回復ソフトウェアは、次のリブート時に GRUB メニューを同じファイルシステムに戻します。たとえば、ファイルシステムで newfs または mkfs コマンドを使用し、誤って GRUB メニューを破棄してしまったとします。GRUB メニューを正しく復元するには、スライスが次の条件を満たす必要があります。

- マウント可能なファイルシステムが含まれている
- スライスが以前に存在していた Solaris Live Upgrade ブート環境の一部である

システムをリブートする前に、必要であればスライスを修正します。

対処方法: システムをリブートします。GRUB メニューのバックアップコピーが自動的にインストールされます。

The GRUB menu's menu.lst file was accidentally deleted.

対処方法: システムをリブートします。GRUB メニューのバックアップコピーが自動的にインストールされます。

## ▼ Veritas VxVm の実行中に Solaris Live Upgrade を使用してアップグレードするとシステムパニックが発生する

Veritas VxVM の実行中に Solaris Live Upgrade を使用してアップグレードを行う場合、次の手順でアップグレードを行わないと、リブート時にシステムパニックが発生します。この問題は、パッケージが Solaris の最新のパッケージガイドラインに従っていない場合に発生します。

- 1 スーパーユーザーになるか、同等の役割になります。  
役割には、認証と特権コマンドが含まれます。役割の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 非アクティブブート環境を作成します。64 ページの「新しいブート環境の作成」を参照してください。
- 3 非アクティブブート環境をアップグレードする前に、非アクティブブート環境上の既存の Veritas ソフトウェアを無効にする必要があります。

- a. 非アクティブブート環境をマウントします。

```
lumount inactive_boot_environment_name mount_point
```

次に例を示します。

```
lumount solaris8 /mnt
```

- b. 次の例のように、vfstab が含まれているディレクトリに移動します。

```
cd /mnt/etc
```

- c. 次の例のように、非アクティブブート環境の vfstab ファイルをコピーします。

```
cp vfstab vfstab.501
```

- d. 次の例のように、コピーされた vfstab 内のすべての Veritas ファイルシステムエントリをコメントにします。

```
sed '/vx\/dsk\/s\/^\/#\/g' < vfstab > vfstab.novxfs
```

各行の最初の文字が # に変わり、その行がコメント行になります。このコメント行は、system ファイルのコメント行とは異なります。

- e. 次の例のように、変更した vfstab ファイルをコピーします。

```
cp vfstab.novxfs vfstab
```

- f. 次の例のように、非アクティブブート環境の **system** ファイルがあるディレクトリに移動します。

```
cd /mnt/etc
```

- g. 次の例のように、非アクティブブート環境の **system** ファイルをコピーします。

```
cp system system.501
```

- h. `drv/vx` を含むすべての **forceload:** エントリをコメントにします。

```
sed '/forceload: drv\/vx\/s\/^\/*/' <system> system.novxfs
```

各行の最初の文字が \* に変わり、その行がコメント行になります。このコメント行は、`vfstab` ファイルのコメント行とは異なります。

- i. 次の例のように、**Veritas install-db** ファイルを作成します。

```
touch vx/reconfig.d/state.d/install-db
```

- j. 非アクティブブート環境のマウントを解除します。

```
luumount inactive_boot_environment_name
```

- 4 非アクティブブート環境をアップグレードします。第 5 章を参照してください。

- 5 非アクティブブート環境をアクティブにします。119 ページの「ブート環境のアクティブ化」を参照してください。

- 6 システムをシャットダウンします。

```
init 0
```

- 7 非アクティブブート環境をシングルユーザーモードでブートします。

```
OK boot -s
```

`vxvm` または `VXVM` を含むメッセージとエラーメッセージがいくつか表示されますが、これらは無視してかまいません。非アクティブブート環境がアクティブになります。

- 8 **Veritas** をアップグレードします。

- a. 次の例のように、システムから **Veritas VRTSvmsa** パッケージを削除します。

```
pkgrm VRTSvmsa
```

- b. **Veritas** パッケージがあるディレクトリに移動します。

```
cd /location_of_Veritas_software
```

- c. システムに最新の **Veritas** パッケージを追加します。

```
pkgadd -d 'pwd' VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvman VRTSvmdv
```

- 9 元の `vfstab` と `system` ファイルを復元します。

```
cp /etc/vfstab.original /etc/vfstab
cp /etc/system.original /etc/system
```

- 10 システムをリブートします。

```
init 6
```

## **x86:** 既存のサービスパーティションが存在しないシステムでは、デフォルトでサービスパーティションが作成されない

診断・サービスパーティションの存在しないシステム上に Solaris 10 11/06 OS をインストールすると、インストールプログラムがデフォルトでサービスパーティションを作成しない場合があります。Solaris パーティションと同じディスクにサービスパーティションを作成する場合、Solaris 10 11/06 OS をインストールする前にサービスパーティションを作り直す必要があります。

サービスパーティションが存在しているシステムに Solaris 8 2/02 OS をインストールした場合、インストールプログラムがサービスパーティションを保持しなかった可能性があります。サービスパーティションを保持するように `fdisk` ブートパーティションレイアウトを手動で編集しなかった場合、インストールプログラムはインストール時にサービスパーティションを削除しています。

---

注 - Solaris 8 2/02 OS のインストール時にサービスパーティションの保持を明示的に指定しなかった場合、サービスパーティションを作り直して Solaris 10 11/06 OS にアップグレードすることができなくなる可能性があります。

---

Solaris パーティションを含むディスクにサービスパーティションを含めたい場合、次のいずれかの方法を選択してください。

## ▼ ネットワークインストールイメージまたは **Solaris Operating System DVD** からのソフトウェアのインストール

ソフトウェアを、ネットインストールイメージからインストールするか、ネットワーク経由で Solaris Operating System DVD からインストールする場合、次の手順を実行します。

- 1 ディスクの内容を削除します。

- 2 インストールする前に、システムの診断用 CD を使用してサービスパーティションを作成します。  
サービスパーティションの作成方法の詳細は、ハードウェアのマニュアルを参照してください。
- 3 ネットワークからシステムをブートします。  
「fdisk パーティションのカスタマイズ」画面が表示されます。
- 4 「デフォルト」をクリックし、デフォルトのブートディスクパーティションレイアウトを読み込みます。  
インストールプログラムにより、サービスパーティションが保持され、Solaris パーティションが作成されます。

## ▼ Solaris SOFTWARE - 1 CD またはネットワークインストールイメージからのインストール

Solaris インストールプログラムを使用して、Solaris SOFTWARE - 1 CD またはブートサーバー上のネットワークインストールイメージからインストールを実行するには、次の手順を実行します。

- 1 ディスクの内容を削除します。
- 2 インストールする前に、システムの診断用 CD を使用してサービスパーティションを作成します。  
サービスパーティションの作成方法の詳細は、ハードウェアのマニュアルを参照してください。
- 3 インストールプログラムにより、Solaris パーティションの作成方法を選択するよう求められます。
- 4 システムをブートします。
- 5 「残りのディスクを使用して Solaris パーティションを配置します」を選択します。  
インストールプログラムにより、サービスパーティションが保持され、Solaris パーティションが作成されます。
- 6 インストールが完了します。



## その他の SVR4 パッケージ要件 (リファレンス)

---

この付録は、パッケージ (特に Sun 以外のパッケージ) のインストールや削除を行うシステム管理者を対象としています。ここで説明するパッケージ要件に従うことにより、次のことが可能になります。

- 現在稼働中のシステムへの変更を防いで、Solaris Live Upgrade によるアップグレードと、非大域ゾーンやディスクレスクライアントの作成・管理とを可能にします。
- カスタム JumpStart などのインストールプログラムを使用する場合に、パッケージを対話式にしないで自動インストールを可能にします。

この章の内容は次のとおりです。

- 203 ページの「稼働中の OS に対する変更の防止」
- 207 ページの「インストール中およびアップグレード中のユーザー操作の回避」
- 208 ページの「ゾーンのパッケージパラメータの設定」

### 稼働中の OS に対する変更の防止

この節で説明する要件に従えば、現在稼働中の OS は変更されません。

#### 絶対パスの使用

オペレーティングシステムのインストールが成功するには、Solaris Live Upgrade の非アクティブブート環境などの代替ルート (/) ファイルシステムをパッケージが認識して、それに従う必要があります。

パッケージの `pkgmap` ファイル (パッケージマップ) には、絶対パスを指定できます。これらのパスが存在する場合、そのファイルは、`pkgadd` コマンドの `-R` オプションとの相対パスに書き込まれます。絶対パスと相対 (再配置可能) パスの両方を含むパッケージは、代替ルート (/) ファイルシステムにもインストールできます。絶対ファイルであれ再配置可能ファイルであれ、その前には `$PKG_INSTALL_ROOT` が付加されるため、`pkgadd` によるインストールでは、すべてのパスが正しく解釈されます。

## pkgadd -R コマンドの使用

pkgadd -R オプションを使ってパッケージをインストールしたり、pkgrm -R オプションを使ってパッケージを削除する場合には、パッケージが、動作中のシステムを変更してはいけません。この機能は、カスタム JumpStart、Solaris Live Upgrade、非大域ゾーン、およびディスクレスクライアントで使用されます。

さらに、pkgadd -R オプションを使ってパッケージをインストールしたり、pkgrm -R オプションを使ってパッケージを削除する場合には、パッケージに同梱のスクリプトで、動作中のシステムを変更してはいけません。インストールスクリプトを作成する場合には、参照するディレクトリやファイルの前に \$PKG\_INSTALL\_ROOT 変数を付加する必要があります。パッケージでは、書き込むすべてのディレクトリやファイルの前に \$PKG\_INSTALL\_ROOT を付加する必要があります。さらに、パッケージでは、\$PKG\_INSTALL\_ROOT 接頭辞を付加せずにディレクトリを削除すべきではありません。

表 B-1 に、スクリプト構文の例を示します。

表 B-1 インストールスクリプト構文の例

スクリプトタイプ	正しい構文	正しくない構文
Bourne シェル「if」ステートメントの一部	<pre>if [ -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf ] ; then</pre>	<pre>if [ -f /etc/myproduct.conf ] ; \ then</pre>
ファイルの削除	<pre>/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>/bin/rm -f /etc/myproduct.conf</pre>
ファイルの変更	<pre>echo "test=no" &gt; \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>echo "test=no" &gt; \ /etc/myproduct.conf</pre>

## \$PKG\_INSTALL\_ROOT と \$BASEDIR の相違点の概要

\$PKG\_INSTALL\_ROOT は、パッケージを追加しようとするマシンのルート (/) ファイルシステムの場所です。この値は、pkgadd コマンドの -R 引数の後にセットされます。たとえば、次のコマンドを実行すると、パッケージのインストール時に \$PKG\_INSTALL\_ROOT の値は /a になります。

```
pkgadd -R /a SUNWvsvm
```

\$BASEDIR は、再配置可能なパッケージオブジェクトがインストールされる「再配置可能」なベースディレクトリを指しています。ここにインストールされるのは、再配置可能オブジェクトだけです。再配置可能でないオブジェクト (pkgmap ファイルに「絶対」パスが指定されているオブジェクト) は、必ず、非アクティブブート環境からの相対位置にインストールされます。\$BASEDIR からの相対位置ではありません。再配置可能なオブ

ジェクトがないパッケージは、絶対パッケージ(再配置不可)と呼ばれます。その場合、\$BASEDIR は未定義であるため、これをパッケージに添付されているスクリプトで使用することはできません。

たとえば、パッケージの pkgmap ファイルに次のエントリがあるとします。

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

さらに、pkginfo ファイルには、\$BASEDIR が次のように指定されているとします。

```
BASEDIR=/opt
```

このパッケージを次のコマンドでインストールすると、ls は、/a/opt/sbin/ls としてインストールされますが、ls2 は、/a/sbin/ls2 としてインストールされます。

```
pkgadd -R /a SUNWtest
```

## スクリプト作成のガイドライン

パッケージ処理のスクリプトを作成するときは、現在稼働中の OS への変更を防ぐために、OS に依存しないようにしてください。スクリプトには、パッケージのインストールや削除の実行中に行うアクションを定義します。事前に決められたプロシージャ名で作成できるスクリプトが4つあります。preinstall、postinstall、preremove、および postremove です。

表 B-2 スクリプト作成のガイドライン

ガイドライン	Solaris Live Upgrade への影響	非大域ゾーンへの影響
スクリプトは Bourne シェル (/bin/sh) で書き込む必要があります。pkgadd コマンドは、スクリプトの実行時にインタプリタとして Bourne シェルを使用します。	○	○
スクリプトはプロセスの開始や停止を行ったり、ps や truss などのコマンドの出力に依存したりしてはいけません。ps や truss はオペレーティングシステムに依存し、稼働中のシステムに関する情報を報告します。	○	○
スクリプトでは、expr、cp、ls などの標準的な UNIX コマンドや、シェルスクリプトの作成を容易にするそのほかのコマンドを自由に使用できます。	○	○

表 B-2 スクリプト作成のガイドライン (続き)

ガイドライン	Solaris Live Upgrade への影響	非大域ゾーンへの影響
<p>パッケージはサポートされているすべてのリリースで動作する必要があるため、スクリプトで呼び出すコマンドはこれらすべてのリリースで利用可能なものでなければなりません。したがって、Solaris 8 リリースのあとで追加または削除されたコマンドは使用できません。</p>	○	
<p>特定のコマンドまたはオプションが Solaris 8、9、または 10 リリースでサポートされているかどうかを調べるには、<a href="http://docs.sun.com">http://docs.sun.com</a> で該当するバージョンの Solaris Reference Manual Collection を参照してください。</p>		

## ディスクレスクライアントの互換性維持

パッケージでは、パッケージ自体が提供しているコマンドを実行してはいけません。これは、ディスクレスクライアントの互換性を維持するためであると同時に、まだインストールされていない共有ライブラリを必要とするコマンドの実行を避けるためです。

## パッケージの検証

すべてのパッケージは `pkgchk` の検証にパスしなければなりません。パッケージを作成したらインストールする前に、次のコマンドでパッケージをチェックする必要があります。

```
pkgchk -d dir_name pkg_name
```

`dir_name` パッケージがあるディレクトリの名前を指定します。

`pkg_name` パッケージの名前を指定します。

例 B-1 パッケージをテストする

パッケージを作成したら、`pkgadd` コマンドに `-R dir_name` オプションを指定して、これを代替ルート (*/*) ファイルシステムにインストールしてテストする必要があります。パッケージをインストールしたら、次のように、パッケージが正しいかどうかを `pkgchk` コマンドでチェックします。

```
pkgadd -d . -R /a SUNWvsvm
```

```
pkgchk -R /a SUNWvsvm
```

エラーが表示されないことを確認します。

例 B-2 /export/SUNWvxxm にあるパッケージをテストする

たとえば、パッケージが /export/SUNWvxxm にあるなら、コマンドを次のように指定します。

```
pkgchk -d /export SUNWvxxm
```

エラーが表示されないことを確認します。

ファイルの作成、変更、削除を行うときに、ほかのコマンドでパッケージを検証することもできます。次にコマンド例を示します。

- たとえば、パッケージが正しく動作するかどうかを `dircmp` や `fssnap` コマンドを使って検証できます。
- さらに、`ps` コマンドでは、パッケージによりデーモンの開始や停止が行われていないことを確認することによってデーモンに対する要件準拠を確認できます。
- `truss`、`pkgadd -v`、および `pkgrm` コマンドで、パッケージインストールの実行要件に準拠しているかどうかを確認できます。ただし、これが常に機能するとはかぎりません。次の例では、`truss` コマンドは、読み取り専用ディレクトリおよび `$TMPDIR` 以外のディレクトリへのアクセス情報をすべて除外し、指定された非アクティブブート環境以外のディレクトリへの読み取り専用でないアクセス情報のみを表示します。

```
TMPDIR=/a; export TMPDIR
truss -t open /usr/sbin/pkgadd -R ${TMPDIR} SUNWvxxm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open("'${TMPDIR}'
```

## インストール中およびアップグレード中のユーザー操作の回避

次の Solaris 標準ユーティリティを使用するときは、ユーザーの情報入力なしに、パッケージの追加や削除が行われる必要があります。

- カスタム JumpStart プログラム
- Solaris Live Upgrade
- Solaris インストール プログラム
- Solaris ゾーン

パッケージをテストして、ユーザー操作なしでインストールされるようにするには、`pkgadd -a` コマンドで新しい管理ファイルを設定します。`-a` オプションは、デフォルトの管理ファイルの代わりにユーザー定義の管理ファイルを使用することを意味します。デフォルトのファイルを使用すると、情報の入力が必要になることがあります。管理ファイルを作成すれば、`pkgadd` でこのようなチェックを省略し、ユーザーの確認なしで

パッケージをインストールすることができます。詳細については、`admin(4)` または `pkgadd(1M)` のマニュアルページを参照してください。

次の例では、`pkgadd` コマンドによる管理ファイルの扱いを示します。

- 管理ファイルを指定しないと、`pkgadd` は `/var/sadm/install/admin/default` を使用します。このファイルを使用すると、ユーザーの入力が必要になることがあります。

```
pkgadd
```

- コマンド行に相対的な管理ファイルを指定すると、`pkgadd` は `/var/sadm/install/admin` からこのファイル名を探して使用します。この例では相対的な管理ファイルの名前が `nocheck` であるため、`pkgadd` は `/var/sadm/install/admin/nocheck` を探します。

```
pkgadd -a nocheck
```

- 絶対パスでファイルを指定すると、`pkgadd` はこれを使用します。この例では、`pkgadd` は `/tmp` 内で `nocheck` 管理ファイルを検索します。

```
pkgadd -a /tmp/nocheck
```

#### 例 B-3 インストール管理ファイル

次に示すのは、`pkgadd` ユーティリティーでユーザーの入力をほとんど必要としないインストール管理ファイルの例です。パッケージがシステムで利用可能な容量を超えた容量を必要としない限り、`pkgadd` ユーティリティーはこのファイルを使用して、ユーザーに情報の入力を求めることなくインストールを実行します。

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
confiict=nocheck
action=nocheck
basedir=default
```

## ゾーンのパッケージパラメータの設定

パッケージが保持するパラメータにより、非大域ゾーンのインストールされたシステムで内容を配布および可視にする方法が制御されます。`SUNW_PKG_ALLZONES`、`SUNW_PKG_HOLLOW`、および `SUNW_PKG_THISZONE` パッケージパラメータは、ゾーンがインストールされているパッケージの特性を定義します。非大域ゾーンのインストールされたシステムでパッケージを管理できるようにするには、これらのパラメータを設定する必要があります。

次の表に、パッケージパラメータの設定に使用できる4つの有効な組み合わせを示します。次の表に示されていない設定の組み合わせは無効であり、そのような設定を選択するとパッケージのインストールは失敗します。

注-3つのパッケージパラメータをすべて設定したことを確認してください。3つのパッケージパラメータをすべて空のままにしてもかまいません。ゾーンのパッケージパラメータが見つからない場合、パッケージツールではfalseの設定として解釈されますが、パラメータの設定を省略しないように強くお勧めします。3つのパッケージパラメータをすべて設定することにより、パッケージをインストールまたは削除するときのパッケージツールの動作を正確に指定します。

表 B-3 ゾーンのパッケージパラメータの有効な設定

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
false	false	false	<p>ゾーンのパッケージパラメータのいずれにも値を指定しないパッケージの場合、これがデフォルト設定として使用されます。</p> <p>この設定を持つパッケージは、大域ゾーンまたは非大域ゾーンにインストールできます。</p> <ul style="list-style-type: none"> <li>■ 大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージは大域ゾーンおよびすべての非大域ゾーンにインストールされます。</li> <li>■ 非大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージはその非大域ゾーンだけにインストールされます。</li> </ul> <p>どちらの場合も、パッケージがインストールされたすべてのゾーンで、パッケージの内容全体が可視になります。</p>

表 B-3 ゾーンのパッケージパラメータの有効な設定 (続き)

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
false	false	true	<p>この設定を持つパッケージは、大域ゾーンまたは非大域ゾーンにインストールできます。インストール後に新しい非大域ゾーンを作成した場合、パッケージはこれらの新しい非大域ゾーンには伝達されません。</p> <ul style="list-style-type: none"> <li>■ 大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージは大域ゾーンだけにインストールされます。</li> <li>■ 非大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージはその非大域ゾーンだけにインストールされます。</li> </ul> <p>どちらの場合も、パッケージがインストールされたゾーンで、パッケージの内容全体が可視になります。</p>
true	false	false	<p>この設定を持つパッケージは、大域ゾーンだけにインストールできます。<code>pkgadd</code> コマンドを実行すると、パッケージは大域ゾーンおよびすべての非大域ゾーンにインストールされます。すべてのゾーンで、パッケージの内容全体が可視になります。</p> <p>注-パッケージを非大域ゾーンにインストールしようとするとう失敗します。</p>

表 B-3 ゾーンのパッケージパラメータの有効な設定 (続き)

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
true	true	false	<p>この設定を持つパッケージは、大域管理者が大域ゾーンだけにインストールできます。pkgadd コマンドを実行すると、パッケージの内容が大域ゾーンに完全にインストールされます。パッケージパラメータの値がこのように設定されている場合、パッケージの内容自体はどの非大域ゾーンにも提供されません。パッケージをインストール済みとして表示するために必要なパッケージインストール情報だけが、すべての非大域ゾーンにインストールされます。これにより、このパッケージに依存するほかのパッケージをインストールできるようになります。「hollow」パッケージの詳細については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 23 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチについて (概要)」を参照してください。</p> <p>パッケージの依存関係を検査できるように、パッケージはすべてのゾーンでインストール済みとして表示されます。</p> <ul style="list-style-type: none"> <li>■ 大域ゾーンでは、パッケージの内容全体が可視になります。</li> <li>■ 完全ルート非大域ゾーンでは、パッケージの内容全体が不可視になります。</li> <li>■ 非大域ゾーンが大域ゾーンからファイルシステムを継承する場合、このファイルシステムにインストールされているパッケージは非大域ゾーンで可視になります。パッケージで提供されるほかのすべてのファイルは、非大域ゾーン内では不可視になります。たとえば、疎ルート非大域ゾーンは、特定のディレクトリを大域ゾーンと共有します。これらのディレクトリは読み取り専用です。疎ルート非大域ゾーンは、/platform などのファイルシステムを共有します。もう 1 つの例は、起動するハードウェアだけに関連するファイルがパッケージで提供されている場合です。</li> </ul> <p>注-パッケージを非大域ゾーンにインストールしようとすると失敗します。</p>

説明	参照先
パッケージとゾーンの詳細	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 23 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチについて (概要)」
疎ルートゾーンと完全ルートゾーンの概要	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 16 章「Solaris ゾーンの紹介」
パッケージの特性およびパラメータの詳細	pkginfo(4)
パッケージパラメータ値の表示の詳細	pkgparam(1)

## 背景情報

パッケージの要件および特定のコマンドの構文については、次の情報を参照してください。

パッケージの要件の詳細および用語の定義	『Application Packaging Developer’s Guide』の第 6 章「Advanced Techniques for Creating Packages」
パッケージの追加と削除およびインストール管理ファイルに関する基本情報	『Solaris のシステム管理 (基本編)』の第 16 章「ソフトウェアの管理 (概要)」
この付録に記載されている個々のコマンドの詳細について参照するマニュアルページ	dircmp(1)、fssnap(1M)、ps(1)、または truss(1)、pkgadd(1M)、pkgchk(1M)、または pkgrm(1M)
Solaris Live Upgrade の概要	第 2 章
カスタム JumpStart の概要	『Solaris 10 11/06 インストールガイド (カスタム JumpStart/ 上級編)』の第 2 章「カスタム JumpStart (概要)」
Solaris ゾーンの概要	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 16 章「Solaris ゾーンの紹介」

## アップグレード時のパッチアナライザの使用(作業)

---

この章では、Solaris OS をアップグレードする前にパッチアナライザでパッチを検査する手順について説明します。最初の Solaris 10 3/05 リリースに続く次のいずれかのリリースにアップグレードする場合に、パッチアナライザはシステムの解析を実行します。

- Solaris 10 1/06 リリース
- Solaris 10 6/06 リリース

### Solaris Update リリースへのアップグレード

すでに Solaris 10 3/05 リリースを実行していて、個別のパッチをインストール済みの場合、以降の Solaris 10 リリースにアップグレードしたときの動作は次のとおりです。

- 上記のリリースの1つでその一部として提供されているすべてのパッチが、システムに再び適用されます。これらのパッチはバックアウトできません。
- システムにすでにインストールされていたパッチのうち、上記のリリースの1つに含まれていないものは、すべて削除されます。

パッチアナライザは、システムを解析し、上記のいずれかのリリースへのアップグレードによって削除されるパッチがあれば、それを特定します。パッチアナライザは次の形式で使用できます。

- Solaris インストールプログラムを使用してアップグレードを行なっている場合は、「パッチの解析」ダイアログボックスが表示されます。「はい」を選択して解析を実行します。
- テキストインストーラを使用してアップグレードを行なっている場合は、「パッチの解析」ダイアログボックスの「解析」を選択して解析を実行します。
- カスタム JumpStart または Solaris Live Upgrade を使用してアップグレードを行なっている場合は、`analyze_patches` スクリプトを実行して解析を実行します。詳しい手順については、214 ページの「`analyze_patches` スクリプトを実行する方法」を参照してください。

解析を実行した後、パッチの解析結果の詳細については、215 ページの「パッチアナライザの出力を確認する方法」を参照してください。

## ▼ analyze\_patches スクリプトを実行する方法

---

注-analyze\_patches スクリプトを実行するには、インストール済みシステムと Solaris Operating System DVD、Solaris SOFTWARE CD、またはネットワークインストールイメージに、NFS またはローカルでマウントされたメディアを通してスクリプトからアクセスできることが必要です。

---

### 1 Misc ディレクトリに移動します。

- SPARC: ローカルでマウントされたメディアにイメージが置かれている場合は、次のように入力します。

```
cd /cdrom/cdrom0/s0/Solaris_10/Misc
```

このコマンドの *Update* は、Update の実際の識別子です。

- x86: ローカルでマウントされたメディアにイメージが置かれている場合は、次のように入力します。

```
cd /cdrom/cdrom0/Solaris_10/Misc
```

このコマンドの *date* は、リリースの実際の日付です (06 など)。

- イメージが NFS ファイルシステム上にある場合は、次のように入力します。

```
cd /NFS_mount_directory/Solaris_10/Misc
```

### 2 analyze\_patches スクリプトを実行します。

```
./analyze_patches -R rootdir -N netdir -D databasedir
```

-R *rootdir*      *rootdir* はインストール済みシステムのルートです。デフォルトは / です。

-N *netdir*      *netdir* は、インストールする OS イメージのルートのパスです。デフォルトは /cdrom/cdrom0 です。*netdir* は、Solaris\_10 ディレクトリを含んでいるディレクトリのパスです。NFS マウントポイントから *patch\_analyzer* を実行している場合は、このオプションを使用する必要があります。

-D *databasedir*      OS イメージ内の Misc/ ディレクトリ以外のディレクトリからスクリプトを呼び出すと、プログラムはパッチの解析に使用するデータベースを見つけられません。-D オプションを使用してデータベースのパスを指定してください。このデータベースは OS イメージの Solaris\_10 /Misc/database に置かれています。このデータベースを使

用しなければ、スクリプトは正しく動作しません。

## ▼ パッチアナライザの出力を確認する方法

解析を実行した後、次の手順で出力を確認します。

### 1 パッチアナライザの出力を確認します。

パッチアナライザでは、ほかのパッチによって削除、ダウングレード、累積、または廃止されるパッチのリストが得られます。パッチの累積はパッチのアップグレードに似ています。累積されるパッチは削除され、その修正は新しいパッチによって提供されず。次のようなメッセージが表示されます。

```
Patch 105644-03 will be removed.
Patch 105925 will be downgraded from -02 to -01.
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.
```

パッチアナライザプログラムでリストが得られない場合は、システムにすでにインストールされていたどのパッチに対しても何のアクションも行われません。

### 2 パッチの置換と削除に同意するかどうかを決定します。

- する場合は、システムをアップグレードします。
- しない場合は、システムをアップグレードしないでください。



# 用語集

---

- 3DES** Triple-Data Encryption Standard (Triple DES) の略。168 ビットの鍵を提供する対称鍵暗号化方法。
- AES** Advanced Encryption Standard の略。対称 128 ビットブロックのデータ暗号技術。米国政府は、2000 年の 10 月に暗号化標準としてこのアルゴリズムの Rijndael 方式を採用しました。DES 暗号化に代わる米国政府の標準として、AES が採用されています。
- bootlog-cgi プログラム** WAN ブートインストール時に、リモートクライアントのブートおよびインストールのコンソールメッセージを Web サーバーで収集し保存できるようにする CGI プログラム。
- certstore ファイル** 特定のクライアントシステムに関するデジタル証明書を格納しているファイル。SSL ネゴシエーションの際、クライアントは証明書ファイルをサーバーに提供するように要求されることがあります。サーバーはこのファイルを使ってクライアントの識別情報を確認します。
- CGI** Common Gateway Interface の略。外部プログラムが HTTP サーバーと通信するためのインタフェース。CGI を使用するプログラムは、CGI プログラムまたは CGI スクリプトと呼ばれます。通常サーバーでは処理されないフォームや解析されない出力を、CGI プログラムが処理したり解析したりします。
- DES** Data Encryption Standard の略。対称鍵暗号化方法の 1 つ。1975 年に開発され、ANSI により 1981 年に ANSI X.3.92 として標準化されました。DES では 56 ビットの鍵を使用します。
- DHCP** Dynamic Host Configuration Protocol (動的ホスト構成プロトコル) の略。アプリケーション層のプロトコル。TCP/IP ネットワーク上の個々のコンピュータつまりクライアントが、中央管理を行なっている指定の DHCP サーバーから IP アドレスなどのネットワーク構成情報を抽出できるようにします。この機能は、大規模な IP ネットワークの保持、管理によるオーバーヘッドを削減します。
- /etc/netboot ディレクトリ** WAN ブートインストールに必要なクライアント構成情報とセキュリティーデータが格納されている、WAN ブートサーバー上のディレクトリ。
- /etc ディレクトリ** 重要なシステム構成ファイルや保守コマンドが収められているディレクトリ。

- /export ファイルシステム** OS サーバー上のファイルシステムで、ネットワーク上のほかのシステムと共有されます。たとえば、/export ファイルシステムには、ディスクレスクライアント用のルート (/) ファイルシステムとスワップ空間、それにネットワーク上のユーザーのホームディレクトリを収めることができます。ディスクレスクライアントは、起動と実行の際に OS サーバー上の /export ファイルシステムに依存します。
- fdisk パーティション** x86 ベースのシステム上にある特定のオペレーティングシステム専用のディスクドライブの論理パーティション。Solaris ソフトウェアをインストールするには、x86 システム上に 1 つ以上の Solaris fdisk パーティションを設定する必要があります。x86 ベースのシステムでは、1 台のディスクに最大 4 つの fdisk パーティションを作成できます。これらのパーティションは、個別のオペレーティングシステムをインストールして使用できません。各オペレーティングシステムは、独自の fdisk パーティション上に存在しなければなりません。個々のシステムの Solaris fdisk パーティションの数は、1 台のディスクにつき 1 つに限られます。
- GRUB** **x86 のみ:** GRUB (GNU GRand Unified Bootloader) は、簡単なメニューインタフェースを備えたオープンソースのブートローダーです。メニューには、システムにインストールされているオペレーティングシステムのリストが表示されます。GRUB を使用すると、Solaris OS、Linux、または Microsoft Windows などのさまざまなオペレーティングシステムを、簡単にブートすることができます。
- GRUB 編集メニュー** **x86 のみ:** GRUB メインメニューのサブメニューであるブートメニュー。このメニューには、GRUB コマンドが表示されます。これらのコマンドを編集して、ブート動作を変更できます。
- GRUB メインメニュー** **x86 のみ:** システムにインストールされているオペレーティングシステムがリストされたブートメニュー。このメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
- HMAC** メッセージ認証を行うためのキー付きハッシュ方法。HMAC は秘密共有鍵と併用して、MD5、SHA-1 などの繰り返し暗号化のハッシュ関数で使用します。HMAC の暗号の強さは、基になるハッシュ関数のプロパティによって異なります。
- HTTP** (Hypertext Transfer Protocol の略) リモートホストからハイパーテキストオブジェクトをフェッチするインターネットプロトコル。このプロトコルは TCP/IP にもとづいています。
- HTTPS** HTTP のセキュリティー保護されたバージョン。SSL (Secure Sockets Layer) を使って実装されます。
- IPv6** IPv6 は、現在のバージョン IPv4 (バージョン 4) から拡張されたインターネットプロトコル (IP) のバージョン (バージョン 6) です。定められた移行方法を使用して IPv6 を採用すると、現在の運用を中断する必要はありません。また、IPv6 には、新しいインターネット機能用のプラットフォームも用意されています。

IPv6の詳細については、『Solarisのシステム管理(IPサービス)』のパートI「システム管理の概要:IPサービス」を参照してください。

**IPアドレス**

インターネットプロトコル(Internet Protocol、IP)アドレス。TCP/IPでは、ネットワーク上の個々のホストを識別する32ビットの一意の数値。IPアドレスは、4つの数をピリオドで区切った形式になります(例:192.168.0.0)。通常、IPアドレスの各部分は0~225の番号ですが、最初の番号は224未満とし、最後の番号は0以外にする必要があります。

IPアドレスは、論理的に次の2つの部分に分割されます。ネットワーク(市外局番のようなもの)とネットワーク上のシステム(電話番号のようなもの)です。たとえば、クラスAのIPアドレス内の数字は「network.local.local.local」を表し、クラスCのIPアドレス内の数字は「network.network.network.local」を表します。

クラス	範囲(xxxは0から255までの数字)	使用できるIPアドレス数
クラスA	1.xxx.xxx.xxx - 126.xxx.xxx.xxx	1,600万以上
クラスB	128.0.xxx.xxx - 191.255.xxx.xxx	65,000以上
クラスC	192.0.0.xxx - 223.255.255.xxx	256

**JumpStartインストール**

インストール方法の1つ。出荷時にインストールされているJumpStartソフトウェアを使用することによって、Solarisソフトウェアをシステムに自動インストールできます。

**JumpStartディレクトリ**

カスタムJumpStartインストールの実行に必要なファイルが含まれているディレクトリ。プロファイルフロッピーディスクを使用してインストールする場合は、フロッピーディスク上のルートディレクトリがJumpStartディレクトリとなります。カスタムJumpStartインストール用にプロファイルサーバーを使用する場合、必要なカスタムJumpStartファイルをすべて格納するサーバー上のディレクトリがJumpStartディレクトリとなります。

**Kerberos**

強力な秘密鍵暗号方式を使用して、クライアントとサーバーが、セキュリティー保護されていないネットワーク接続で相互を認識できるようにするネットワーク認証プロトコル。

**keystoreファイル**

クライアントとサーバーとで共有される鍵を格納しているファイル。WANブートインストール時に、クライアントシステムは鍵を使って、サーバーから送信されるデータやファイルの整合性の確認と復号化を行います。

**LAN**

local area networkの略。接続用のハードウェアとソフトウェアを介して通信できる、近接したコンピュータシステムの集まり。

**LDAP**

Lightweight Directory Access Protocolの略。LDAPネームサービスクライアントとサーバー間の通信に使用される標準の拡張可能なディレクトリアクセスプロトコル。

**MDS**

Message Digest 5の略。デジタル署名などのメッセージ認証に使用する繰り返し暗号化のハッシュ関数。1991年にRivest氏によって開発されました。

- menu.lst ファイル** **x86** のみ: システムにインストールされているすべてのオペレーティングシステムがリストされたファイル。このファイルの内容は、GRUB メニューに表示されるオペレーティングシステムの一覧を記述したものです。GRUB のメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
- NIS** SunOS 4.0 (以上) のネットワーク情報サービス。ネットワーク上のシステムとユーザーに関する重要な情報が収められている分散型ネットワークデータベース。NIS データベースは、マスターサーバーとすべてのスレーブサーバーに格納されています。
- NIS+** SunOS 5.0 (以上) のネットワーク情報サービス。NIS+ は、SunOS 4.0 (以上) のネットワーク情報サービスである NIS に代わるものです。
- /opt ファイルシステム** Sun 以外のソフトウェア製品や別製品のソフトウェア用のマウントポイントが収められているファイルシステム。
- OS サーバー** ネットワーク上のシステムにサービスを提供するシステム。ディスクレスクライアントにサービスを提供するには、OS サーバーは、ディスクレスクライアントごとに、ルート (/) ファイルシステムとスワップ空間 (/export/root、/export/swap) 用のディスク容量が必要です。
- RAID-0 ボリューム** ストライプ方式または連結方式のボリューム。これらはサブミラーとも呼ばれます。ストライプや連結は、ミラーを構築する基本構成ブロックです。
- RAID-1 ボリューム** 同じデータのコピーを複数保持しているボリューム。RAID-1 ボリュームは、「サブミラー」と呼ばれる 1 つまたは複数の RAID-0 ボリュームから構成されます。RAID-1 ボリュームは「ミラー」と呼ばれることもあります。
- rules.ok ファイル** rules ファイルから生成されたファイル。カスタム JumpStart インストールソフトウェアは、rules.ok ファイルを使ってシステムとプロファイルを照合します。rules.ok ファイルは、check スクリプトを使用して作成してください。
- rules ファイル** 自動的にインストールするシステムの各グループまたは単一のシステムのルールを含んでいるテキストファイル。各ルールは 1 つ以上のシステム属性に基づいてシステムグループを識別します。rules ファイルは、各グループをプロファイル (Solaris ソフトウェアをどのようにしてグループ内の個々のシステムにインストールするかを定めたテキストファイル) にリンクします。rules ファイルは、カスタム JumpStart インストールで使用されます。「プロファイル」も参照してください。
- Secure Sockets Layer** (SSL) クライアントとサーバーの間にセキュリティー保護された接続を確立するソフトウェアライブラリ。HTTP のセキュリティー保護されたバージョンである HTTPS を実装するために使用されます。
- SHA1** Secure Hashing Algorithm の略。このアルゴリズムは、長さが  $2^{64}$  未満の入力に対して演算を行い、メッセージダイジェストを生成します。

<b>Solaris Live Upgrade</b>	アクティブブート環境が稼動している間に複製ブート環境のアップグレードを行うことにより、稼動中の環境のダウンタイムをなくすことを可能にするアップグレード方法。
<b>Solaris DVD または CD イメージ</b>	システムにインストールされる Solaris ソフトウェア。Solaris DVD や CD から、または Solaris DVD や CD イメージをコピーしたインストールサーバーのハードディスク上から利用できます。
<b>Solaris インストールプログラム</b>	グラフィカルユーザーインターフェース (GUI) とコマンド行インターフェース (CLI) を備えたインストールプログラム。ウィザードパネルに、Solaris ソフトウェアやサードパーティソフトウェアをインストールする手順が示されます。
<b>Solaris ゾーン</b>	ソフトウェアによるパーティション分割技術。オペレーティングシステムのサービスを仮想化し、隔離された安全なアプリケーション実行環境を提供します。非大域ゾーンを作成すると、そのアプリケーション実行環境で実行されるプロセスは、ほかのゾーンとは隔離されます。このように隔離することで、あるゾーンで実行中のプロセスが、ほかのゾーンで実行中のプロセスを監視したり操作したりすることを防ぐことができます。「大域ゾーン」と「非大域ゾーン」も参照してください。
<b>Solaris フラッシュ</b>	「マスターシステム」と呼ばれるシステムからファイルのアーカイブを作成する Solaris インストール機能。このアーカイブを使ってほかのシステムのインストールを行うと、そのシステムの構成はマスターシステムと同じになります。「アーカイブ」も参照してください。
<b>sysidcfg ファイル</b>	システムを事前設定する特殊な一連のシステム構成キーワードを指定するファイル。
<b>truststore ファイル</b>	1つ以上のデジタル証明書を格納しているファイル。WAN ブートインストール時に、クライアントシステムは truststore ファイル内のデータを参照して、インストールを実行しようとしているサーバーの識別情報を確認します。
<b>URL</b>	Uniform Resource Locator の略。サーバーやクライアントがドキュメントを要求するために使用するアドレス方式。URL はロケーションとも呼ばれます。URL の形式は <i>protocol://machine:port/document</i> です。  たとえば、 <a href="http://www.example.com/index.html">http://www.example.com/index.html</a> は URL の一例です。
<b>/usr ファイルシステム</b>	スタンドアロンシステムまたはサーバー上のファイルシステム。標準 UNIX プログラムの多くが格納されています。ローカルコピーを保持する代わりに、大きな /usr ファイルシステムをサーバーと共有することにより、システム上で Solaris ソフトウェアをインストールおよび実行するために必要なディスク容量を最小限に抑えることができます。
<b>/var ファイルシステム</b>	システムの存続期間にわたって変更または増大が予想されるシステムファイルが格納されている (スタンドアロンシステム上の) ファイルシステムまたはディレクトリ。これらのファイルには、システムログ、vi ファイル、メールファイル、UUCP ファイルなどがあります。

<b>WAN</b>	wide area network の略。複数のローカルエリアネットワーク (LAN) または地理的に異なる場所にあるシステムを、電話、光ファイバ、衛星などの回線を使って接続するネットワーク。
<b>wanboot-cgi</b> プログラム	WAN ブートインストールで使用されるデータとファイルの取得と転送を行う CGI プログラム。
<b>wanboot.conf</b> ファイル	WAN ブートインストールに必要な構成情報とセキュリティー設定値を指定するテキストファイル。
<b>wanboot</b> プログラム	WAN ブートインストールの実行に必要な、WAN ブートミニルート、クライアント構成ファイル、およびインストールファイルを読み込む、二次レベルのブートプログラム。WAN ブートインストールでは、 <b>wanboot</b> バイナリが、 <b>ufsboot</b> または <b>inetboot</b> 二次ブートプログラムと同様の処理を実行します。
<b>WAN</b> ブートインストール	HTTP または HTTPS を使って広域ネットワーク (WAN) を介してソフトウェアをブートしインストールできるインストール方式。WAN ブートインストールでは、暗号化された Solaris フラッシュアーカイブをパブリックネットワークを介して転送し、リモートクライアントに対してカスタム JumpStart インストールを実行できます。
<b>WAN</b> ブートサーバー	WAN ブートインストールで使用される構成ファイルとセキュリティーファイルを提供する Web サーバー。
<b>WAN</b> ブートミニルート	WAN ブートインストールを実行するために変更されたミニルート。WAN ブートミニルートには、Solaris ミニルートにあるソフトウェアのサブセットが格納されます。「ミニルート」も参照してください。
アーカイブ	マスターシステムからコピーされたファイルの集合体。このファイルには、アーカイブの名前や作成した日付など、アーカイブの識別情報が含まれています。アーカイブをシステムにインストールすると、システムはマスターシステムとまったく同じ構成になります。  更新前のマスターイメージと更新されたマスターイメージの相違部分のみを含む Solaris フラッシュアーカイブを、差分アーカイブとして使用することも可能です。差分アーカイブには、クローンシステムで保持、変更、または削除するファイルが含まれます。差分更新により、指定されたファイルだけが更新されます。また、差分更新を使用可能なシステムは、更新前のマスターイメージとの整合性を保持するソフトウェアを含むシステムのみに限定されます。
アップグレード	ファイルを既存のファイルとマージし、可能な場合には変更を保持するインストール。  Solaris OS のアップグレードでは、Solaris OS の新しいバージョンがシステムのディスク上の既存のファイルにマージされます。アップグレードでは、既存の Solaris OS に対して行なった変更は可能な限り保存されます。

アップグレードオプション	Solaris インストールによって提示されるオプション。アップグレード時には、新しいバージョンの Solaris とディスク上の既存のファイルが結合されます。前回 Solaris をインストールしてから加えられたローカルの変更内容は、できる限り残されます。
暗号化	認められたユーザー以外は情報を使用できないように、情報を判読不可能にして保護する処理。暗号化は鍵と呼ばれるコードに基づいて行われ、この鍵は情報の復号化に使用されます。「復号化」も参照してください。
一次ブートアーカイブ	システムで Solaris OS をブートするために使用されるブートアーカイブ。このブートアーカイブは、一次ブートアーカイブと呼ばれることもあります。「ブートアーカイブ」を参照してください。
インストールサーバー	インストール用に、Solaris DVD または CD のイメージをネットワーク上のほかのシステムに提供するサーバー（「メディアサーバー」とも呼ばれる）。Solaris DVD または CD のイメージをサーバーのハードディスクにコピーすることによってインストールサーバーを作成できます。
エンドユーザーシステムサポート	コアシステムサポートソフトウェアグループのほかに、エンドユーザーに推奨するソフトウェアが収められているソフトウェアグループ。これには共通デスクトップ環境 (CDE) や DeskSet ソフトウェアが含まれます。
開始スクリプト	ユーザーが定義する Bourne シェルスクリプト。rules ファイル内で指定され、Solaris ソフトウェアがシステムにインストールされる前に作業を実行します。このスクリプトは、カスタム JumpStart インストールでのみ使用できます。
開発者システムサポート	エンドユーザーシステムサポートソフトウェアグループのほかに、ソフトウェア開発用ライブラリ、インクルードファイル、マニュアルページ、およびプログラミングツールが収められているソフトウェアグループ。
鍵	データの暗号化および復号化に使用されるコード。「暗号化」も参照してください。
カスタム JumpStart	ユーザーが定義するプロファイルに基づいて、Solaris ソフトウェアをシステムに自動的にインストールする方法。ユーザーやシステムの種類ごとに、カスタマイズされたプロファイルを作成できます。カスタム JumpStart インストールは、ユーザーが作成する JumpStart インストールです。
カスタムプローブファイル	rules ファイルと同じ JumpStart ディレクトリに存在しなければならないファイルで、次の2つのタイプの関数を含む Bourne シェルスクリプト。含む2つのタイプは、プローブと比較です。プローブ関数は作業を実行して、必要な情報を収集したり、定義に対応した SI_環境変数を設定します。プローブ関数は、プローブキーワードになります。比較関数は、対応するプローブ関数を呼び出してプローブ関数の出力を比較し、キーワードが一致する場合は0、キーワードが一致しない場合は1を返します。比較関数はルールキーワードになります。「rules ファイル」も参照してください。
共有可能ファイルシステム	/export/home や /swap のようなユーザー定義のファイルシステム。Solaris Live Upgrade の使用時に、アクティブブート環境と非アクティブブート環境によって共有されます。共

有可能ファイルシステムは、アクティブブート環境と非アクティブブート環境の両方の `vfstab` ファイル内に同じマウントポイントを持ちます。このため、アクティブブート環境内の共有ファイルを更新すると、非アクティブブート環境のデータも更新されます。共有可能ファイルシステムはデフォルトで共有されますが、ユーザーが宛先スライスを指定することもできます。この場合、そのファイルシステムがコピーされます。

**クライアント** 通信用のクライアントサーバーモデルでは、計算機能や大容量のメモリーといったサーバーの資源にリモートアクセスするプロセスがクライアントに相当します。

**クラスタ** パッケージ(ソフトウェアモジュール)を論理的に集めたもの。Solaris ソフトウェアは「ソフトウェアグループ」に分割され、それぞれがクラスタと「パッケージ」から構成されています。

**クリティカルファイルシステム** Solaris OSが必要とするファイルシステム。Solaris Live Upgrade を使用するとき、これらのファイルシステムは、アクティブブート環境と非アクティブブート環境それぞれの `vfstab` ファイルでは独立したマウントポイントになります。`root (/)`、`/usr`、`/var`、`/opt` などがクリティカルファイルシステムの例です。これらのファイルシステムは、必ずソースブート環境から非アクティブブート環境にコピーされます。

**クローンシステム** Solaris フラッシュアーカイブを使用してインストールするシステム。クローンシステムは、マスターシステムと同一のインストール構成になります。

**限定ネットワークシステムサポート** ソフトウェアグループの1つ。Solaris システムのブートおよび実行に必要な最小限のコードが含まれ、ネットワークサービスのサポートも制限されます。限定ネットワークシステムサポートは、テキストベースのマルチユーザーコンソールと、システム管理ユーティリティを提供します。このソフトウェアグループを使用すると、システムでネットワークインタフェースを認識できますが、ネットワークサービスがアクティブになることはありません。

**コアソフトウェアグループ** システムで Solaris OS を起動して実行するのに必要な最小限のソフトウェアが収められているソフトウェアグループ。コアには共通デスクトップ環境(CDE)を実行するために必要ないくつかのネットワーク用ソフトウェアとドライバが含まれます。CDE ソフトウェアは、コアには含まれません。

**公開鍵** 公開鍵暗号方式で使用される暗号化鍵。

**公開鍵暗号化** 2つの鍵を使用する暗号方式。その1つは、全員が知っている公開鍵、もう1つは、メッセージの受取人だけが知っている非公開鍵です。

**更新** システムにインストールを実行して同じタイプのソフトウェアを変更することまたはそのインストール自体。アップグレードとは異なり、更新によりシステムがダウングレードされる場合があります。初期インストールとは異なり、更新を実行するには同じタイプのソフトウェアがあらかじめインストールされていなければなりません。

**コマンド行** コマンドで始まる文字列。多くの場合、コマンドの後には引数(オプション、ファイル名、式などの文字列)が続き、行末(EOL)文字で終わります。

サーバー	資源を管理し、クライアントにサービスを提供するネットワークデバイス。
サブネット	経路指定を簡単にするため、1つの論理ネットワークを小さな物理ネットワークに分割する方式。
サブネットマスク	サブネットアドレス指定のため、インターネットアドレスからビットを選択するために使用されるビットマスク。マスクは32ビット。インターネットアドレスのネットワーク部分と、ローカル部分の1個以上のビットを選択します。
サブミラー	「RAID-0 ボリューム」を参照してください。
差分アーカイブ	更新前のマスターイメージと更新されたマスターイメージの相違部分のみを含む Solaris フラッシュアーカイブ。差分アーカイブには、クローンシステムで保持、変更、または削除するファイルが含まれます。差分更新により、指定されたファイルだけが更新されます。また、差分更新を使用可能なシステムは、更新前のマスターイメージとの整合性を保持するソフトウェアを含むシステムのみに限定されます。
時間帯	グリニッジ標準時間を基準に地球の表面を24の地域に経度分割したもの。
システム構成ファイル	(system.conf) WAN ブートインストールで使用する sysidcfg ファイルおよびカスタム JumpStart ファイルの場所を指定するテキストファイル。
終了スクリプト	ユーザーが定義する Bourne シェルスクリプト。rules ファイル内で指定され、Solaris ソフトウェアがシステムにインストールされてから、システムがリブートされるまでの間に作業を実行します。このスクリプトは、カスタム JumpStart インストールで使用します。
状態データベース	Solaris ボリュームマネージャー構成の状態に関する情報を保存するデータベース。状態データベースは、複製された複数のデータベースコピーの集まりです。各コピーは「状態データベースの複製」と呼ばれます。状態データベースは、既知の状態データベースの複製の格納場所と状態をすべて記録しています。
状態データベースの複製	状態データベースのコピー。複製により、データベース内のデータの有効性が保証されます。
初期インストール	現在実行中のソフトウェアを上書きするか、空のディスクを初期化するインストール。  Solaris OS の初期インストールでは、システムのディスクが Solaris OS の新しいバージョンで上書きされます。システム上で Solaris OS が稼動していない場合は、初期インストールを行う必要があります。アップグレード可能な Solaris OS がシステム上で稼動している場合は、初期インストールを行うとディスクが上書きされ、OS やローカルの変更内容は保持されません。
ジョブ	コンピュータシステムで実行されるユーザー定義の処理。

スーパーユーザー	システム上ですべての管理タスクを実行する特権を持つ、特殊なユーザー。スーパーユーザーは全ファイルの読み取り権とアクセス権、全プログラムの実行権を持ち、任意のプロセスに終了シグナルを送ることができます。
スタンドアロン	ほかのマシンからのサポートを一切必要としないコンピュータ。
スライス	ソフトウェアごとに分割される、ディスク領域の区分。
スワップ空間	メモリーに再ロードできる状態になるまで、メモリー領域の内容を一時的に保持するスライスまたはファイル。/swap または swap ファイルシステムとも呼ばれます。
全体ディストリビューション	Solaris 10 11/06 のリリース全体が含まれているソフトウェアグループ。
全体ディストリビューションと OEM サポート	Solaris 10 11/06 のリリース全体と、OEM のための追加ハードウェアサポートを含むソフトウェアグループ。Solaris を SPARC 搭載サーバーシステムにインストールする場合は、このソフトウェアグループを推奨します。
ゾーン	「非大域ゾーン」を参照してください。
ソフトウェアグループ	Solaris ソフトウェアの論理グループ(クラスタとパッケージ)。Solaris のインストール時には、次のいずれかのソフトウェアグループをインストールできます。コアシステムサポート、エンドユーザーシステムサポート、開発者システムサポート、または全体ディストリビューションです。また、SPARC システムのみ、全体ディストリビューションと OEM サポートもインストールできます。
大域ゾーン	Solaris ゾーンでは、大域ゾーンはシステムのデフォルトのゾーンであり、システム全体での管理に使用されるゾーンでもあります。非大域ゾーンの構成、インストール、管理、およびアンインストールは、大域ゾーンからのみ行うことができます。物理デバイス、ルーティング、動的再構成 (DR) といったシステムインフラストラクチャーの管理は、大域ゾーンでのみ行うことができます。大域ゾーンで実行されるプロセスは、適切な権限が付与されていれば、ほかのゾーンに関連付けられているオブジェクトにもアクセスできます。「Solaris ゾーン」と「非大域ゾーン」も参照してください。
チェックサム	一連のデータ項目を合計した結果。一連のデータ項目を検査するために使用されます。データ項目は、数値でも、文字列でもよく、文字列の場合はチェックサム計算時に数値として扱われます。チェックサムの値から、2つのデバイス間の情報交換が正しく行われたかを確認できます。
ディスク (disc)	磁気ディスク (disk) に対する光学式ディスク。CD (コンパクトディスク) 業界では共通の綴りを使用します。たとえば、CD-ROM や DVD-ROM は光学式ディスクです。
ディスク (disk)	1 枚以上の磁性体の円盤から成るメディアであり、ファイルなどのデータを格納する同心トラックとセクターで構成されます。「ディスク (disc)」も参照してください。

ディスク構成ファイル	ディスクの構造(たとえば、バイト/セクター、フラグ、スライス)を表現するファイル。ディスク構成ファイルにより、単一システムから <code>pfinstall</code> コマンドを使用して、サイズの異なるディスクのプロファイルをテストできます。
ディスクレスクライアント	ディスク記憶装置を持たないためサーバーに依存するクライアント。
デジタル証明書	移転や偽造の不可能なデジタルファイルで、通信する両者によって信頼済みの第三者機関から発行されたもの。
電源管理システム	30 分間アイドル状態が続くとシステムの状態を自動的に保存し、電源を切断するソフトウェア。米国環境保護庁の省電力 (Energy Star) ガイドライン第 2 版に準拠したシステムに Solaris ソフトウェアをインストールすると、デフォルトで電源管理システムソフトウェアがインストールされます。デフォルトで電源管理がインストールされているシステムの例として、sun4u SPARC ベースのシステムがあります。リポート後、電源管理ソフトウェアを有効にするかどうかを確認するメッセージが表示されます。  Energy Star ガイドラインでは、システムまたはモニターを使用していない場合は、自動的に「休眠状態」(30 ワット以下の消費)に入ることが要求されます。
ドキュメントルートディレクトリ	Web サーバーにアクセスするユーザーに公開されるファイル、画像、およびデータが格納されている、Web サーバーマシン上の階層のルート。
ドメイン	インターネットのネーミング階層の一部。ドメインは管理ファイルを共有する、ローカルネットワーク上のシステムグループを表します。
ドメイン名	ローカルネットワーク上のシステムグループに割り当てられた名前であり、管理ファイルを共有します。ネットワーク情報サービス (NIS) のデータベースが正常に動作するためにはドメイン名が必要です。ドメイン名は、ピリオドで区切られた一連の構成要素名から構成されます(たとえば、 <code>tundra.mpk.ca.us</code> )。ピリオドで区切られた各構成要素名は右側に行くにしたがって、全体的な(リモートな)管理権限領域を表します。
認証局 (CA)	CA は、Certificate Authority の略。デジタル署名および公開鍵と非公開鍵のペアの作成に使用するデジタル証明書を発行する、公証された第三者機関または企業。CA は、一意の証明書を付与された個人が当該の人物であることを保証します。
ネームサーバー	ネットワーク上のシステムに対してネームサービスを提供するサーバー。
ネームサービス	ネットワーク上の全システムに関する重要なシステム情報が収められている分散型ネットワークデータベース。ネットワーク上のシステムは、これを利用して相互通信を行います。ネームサービスを使用することによって、ネットワーク全域にわたるシステム情報を保守、管理、または取得できます。ネームサービスを使用しないと、各システムはローカルの <code>/etc</code> ファイルにシステム情報のコピーを保持しなければなりません。Sun は次のネームサービスをサポートしています。LDAP、NIS、および NIS+ です。

ネットワークインストール	CD-ROM または DVD-ROM ドライブがあるシステムから CD-ROM または DVD-ROM ドライブがないシステムにネットワークを介してソフトウェアをインストールする方法。ネットワークインストールを行うには、「ネームサーバー」と「インストールサーバー」が必要です。
ネットワークに接続されていないシステム	ネットワークに接続されていない、またはほかのシステムに依存しないシステム。
ネットワークに接続されているシステム	ハードウェアやソフトウェアを介して接続されているシステムのグループ(ホスト)。通信や情報の共有が可能です。ローカルエリアネットワーク (LAN) とも呼ばれます。システムをネットワークに接続するには、通常、1 台以上のサーバーが必要です。
ハードリンク	ディスク上のファイルを参照するディレクトリエントリ。複数のハードリンクから同じ物理ファイルを参照することができます。
派生プロファイル	カスタム JumpStart インストール時に、開始スクリプトによって動的に作成されるプロファイル。
パッケージ	モジュール形式でのインストールを可能にするソフトウェアの集まり。Solaris ソフトウェアは「ソフトウェアグループ」に分割され、それぞれがクラスと「パッケージ」から構成されています。
ハッシュ	入力よりもかなり短い数値を生成する処理によって得られる数値。同じ入力に対しては、常に同じ値が出力されます。ハッシュ関数は、テーブル検索アルゴリズム、エラー検出、改ざん検出などに使用できます。改ざん検出に使用する場合は、同じ結果を生成する別の入力を見つけないようなハッシュ関数を選択します。1 方向のハッシュ関数の一例としては、MD5 および SHA-1 があります。たとえば、メッセージダイジェストはディスクファイルなどの可変長入力を受け取り、小さい値に変換します。
ハッシュ化	文字列を変換して、この元の文字列を表す値(キー)を得る処理。
パッチアナライザ	手作業でも、Solaris インストールプログラム内でも実行できるスクリプト。パッチアナライザは、システムを解析し、Solaris Update へのアップグレードを行うことで削除されるパッチがどれであるかを判断します。
パネル	ウィンドウ、ダイアログボックス、アプレットの内容を編成するコンテナ。パネルでは、ユーザーの入力をまとめて受け取り確認することができます。ウィザードでパネルを使用することで、正しい順序で操作を行い、目的の作業を完了することができます。
非公開鍵	公開鍵暗号方式で使用される復号化鍵。
非大域ゾーン	Solaris オペレーティングシステムの 1 つのインスタンス内で作成される、仮想化されたオペレーティングシステム環境。非大域ゾーンでは、システムのほかの部分と相互に作用することなく、1 つ以上のアプリケーションを実行できます。非大域ゾーンはゾーンとも呼ばれます。「Solaris ゾーン」と「大域ゾーン」も参照してください。

ファイルサーバー	ネットワーク上のシステムに対して、ソフトウェアやファイルの記憶領域を提供するサーバー。
ファイルシステム	SunOS™ オペレーティングシステムにおいて、ユーザーがアクセスできるファイルおよびディレクトリから成るツリー構造のネットワークのこと。
ファンクションキー	F1、F2、F3 などの名前が付いた 10 個以上のキーボードキー。これらのキーにはそれぞれ特定の機能が割り当てられています。
ブート	メモリーにシステムソフトウェアを読み込んで起動すること。
ブートアーカイブ	<p><b>x86</b> のみ: ブートアーカイブは、Solaris OS のブートに使用されるクリティカルなファイルの集まりです。これらのファイルは、ルート (/) ファイルシステムがマウントされる前、システムの起動中に必要です。システムは、2つのブートアーカイブを維持管理しています。</p> <ul style="list-style-type: none"> <li>■ システムで Solaris OS をブートするために使用されるブートアーカイブ。このブートアーカイブは、一次ブートアーカイブと呼ばれることもあります。</li> <li>■ 一次ブートアーカイブが損傷を受けたとき、回復のために使用されるブートアーカイブ。このブートアーカイブは、ルート (/) ファイルシステムをマウントすることなくシステムを起動します。GRUB メニューでは、このブートアーカイブはフェイルセーフと呼ばれます。アーカイブの重要な目的は一次ブートアーカイブを再生成することであり、通常、一次ブートアーカイブがシステムのブートに使用されます。</li> </ul>
ブート環境	<p>Solaris OS を操作する上で重要な必須ファイルシステム (ディスクスライスおよびマウントポイント) の集まり。ディスクスライスは、同じ 1 つのディスク上に存在することも、分散された複数のディスク上に存在することもあります。</p> <p>アクティブなブート環境とは、現在ブートしている環境を指します。単一のアクティブなブート環境からだけブートできます。アクティブでないブート環境とは、現在ブートしていないが、次のリブート時にアクティブ化できる状態にある環境のことを指します。</p>
ブートサーバー	同じネットワークのサブネット上のクライアントシステムに、起動に必要なプログラムと情報を提供するサーバーシステム。インストールサーバーの存在するサブネットが、Solaris ソフトウェアをインストールする必要があるシステムと異なる場合、ネットワークを介してインストールするにはブートサーバーが必要です。
ブートローダー	<b>x86</b> のみ: ブートローダーは、システムの電源を入れた後に最初に実行されるソフトウェアプログラムです。このプログラムがブートプロセスを開始します。
フェイルセーフブートアーカイブ	<b>x86</b> のみ: 一次ブートアーカイブが損傷を受けたとき、回復のために使用されるブートアーカイブ。このブートアーカイブは、ルート (/) ファイルシステムをマウントすることなくシステムを起動します。このブートアーカイブは、GRUB メニューではフェイル

セーフと呼ばれます。アーカイブの重要な目的は一次ブートアーカイブを再生成することであり、通常、一次ブートアーカイブがシステムのブートに使用されます。「ブートアーカイブ」を参照してください。

**フォーマット** データを一定の構造にしたり、データを保存できるようにディスクをセクターに分割したりすること。

**フォールバック** 以前に動作していた環境に戻すこと。ブート環境のアクティブ化の処理中や、ブート対象として指定されたブート環境に問題または望ましくない動作が発生する場合にはフォールバックを行います。

**復号化** 符号化されたデータを平文に変換する処理。「暗号化」も参照してください。

**プラットフォームグループ** 特定のソフトウェア用にベンダーが定義するハードウェアプラットフォームのグループ。たとえば i86pc や sun4c などです。

**プラットフォーム名** uname -i コマンドによって出力される情報。たとえば Ultra 60 のプラットフォーム名は、SUNW,Ultra-60 です。

**プローブキーワード** インストールにカスタム JumpStart を使用する場合、システムに関する属性情報を抽出する構文要素。プローブキーワードでは、ルールに必要な一致条件の設定およびプロファイルの実行は必要ありません。「ルール」も参照してください。

**プロファイル** カスタム JumpStart を使用する場合に、Solaris ソフトウェアのインストール方法を定義するテキストファイル。たとえば、プロファイルでインストールするソフトウェアグループを定義します。各ルールは、そのルールが一致したときにシステムがインストールされる方法を定義してあるプロファイルを指定します。通常は、ルールごとに異なるプロファイルを作成します。しかし、複数のルールで同じプロファイルを使用することも可能です。「rules ファイル」も参照してください。

**プロファイルサーバー** すべての重要なカスタム JumpStart ファイルを JumpStart ディレクトリに持つサーバー。

**プロファイルフロッピーディスク** すべての重要なカスタム JumpStart ファイルを、そのルートディレクトリ (JumpStart ディレクトリ) に持つフロッピーディスク。

**ホスト名** システムがネットワーク上のほかのシステムから識別される名前。この名前は、特定のドメイン (通常、これは1つの組織内にあることを意味する) 内にある全システム間で固有でなければなりません。ホスト名は、文字、数字、マイナス符号 (-) を任意に組み合わせることで作成できますが、先頭と末尾にマイナス符号は使用できません。

**ボリューム** システムで単一の論理デバイスとして扱われる、物理スライスやボリュームの集まり。ボリュームは、アプリケーションやファイルシステムにとって物理ディスクと同じように機能します。

一部のコマンド行ユーティリティーでは、ボリュームはメタデバイスと呼ばれます。一般的な UNIX 用語では、ボリュームは、「擬似デバイス」または「仮想デバイス」とも呼ばれます。

ボリュームマネージャー	DVD-ROM、CD-ROM、およびフロッピーディスク上のデータへのアクセスを管理および実行するための手段を提供するプログラム。
マウント	マウント要求を行うマシンのディスクまたはネットワーク上のリモートディスクから、ディレクトリにアクセスするプロセス。ファイルシステムをマウントするには、ローカルシステム上のマウントポイントと、マウントするファイルシステム名(たとえば /usr)が必要です。
マウント解除	マシンに接続されたディスクまたはネットワーク上のリモートディスク上のディレクトリへのアクセスを解除するプロセス。
マウントポイント	リモートマシン上に存在するファイルシステムのマウント先となる、ワークステーション上のディレクトリ。
マスターシステム	Solaris フラッシュアーカイブを作成するシステム。このシステム構成がアーカイブに保存されます。
マニフェストセクション	クローンシステムの検証に使用される Solaris フラッシュアーカイブのセクション。マニフェストセクションには、クローンシステムに保持、追加または削除されるシステム上のファイルが一覧表示されます。このセクションは、情報提供専用です。このセクションは、ファイルを内部形式でリストするため、スクリプトの記述には使用できません。
ミニルート	Solaris インストールメディアに収録されている、起動可能な最小のルート (/) ファイルシステム。ミニルートには、システムをインストールおよびアップグレードするために必要な Solaris ソフトウェアが含まれます。x86 システムでは、ミニルートはシステムにコピーされ、フェイルセーフブートアーカイブとして使用されます。「フェイルセーフブートアーカイブ」を参照してください。
ミラー	「RAID-1 ボリューム」を参照してください。
メタデバイス	「ボリューム」を参照してください。
メディアサーバー	「インストールサーバー」を参照してください。
矢印キー	数値キーパッド上にある方向を示す 4 つのキーの 1 つ。
ユーティリティー	コンピュータを購入すると通常無料で提供される、標準プログラム。
ルート	複数の項目から成る階層構造の最上位。ルートは、ほかのすべての項目を子孫として持つ唯一の項目です。「ルートディレクトリ」または「ルート (/) ファイルシステム」を参照してください。

---

ルート (/) ファイルシステム	ほかのすべてのファイルシステムの元となる最上位ファイルシステム。ルート (/) ファイルシステムはほかのすべてのファイルシステムがマウントされる元となり、マウント解除されることはありません。ルート (/) ファイルシステムには、カーネル、デバイスドライバ、システムの起動(ブート)に使用されるプログラムなど、システムの稼働に不可欠なディレクトリやファイルが含まれています。
ルートディレクトリ	ほかのすべてのディレクトリの元となる最上位ディレクトリ。
ルール	1つ以上のシステム属性をプロファイルに割り当てる一連の値。ルールは、カスタム JumpStart インストールで使用されます。
連結	RAID-0 ボリューム。複数のスライスが連結された方式では、利用可能な最初のスライスがいっぱいになるまでそのスライスにデータが書き込まれます。そのスライスがいっぱいになると次のスライスに連続してデータが書き込まれます。ミラーに含まれている場合を除き、連結にはデータの冗長性はありません。「RAID-0 ボリューム」も参照してください。
ロケール	同一の言語、風俗、慣習、文化などを共有する地理上または政治上の地域圏(コミュニティー)。たとえば、米国英語のロケールは en_US、英国英語のロケールは en_UK です。
論理デバイス	システムで単一のデバイスとして扱われる、1つまたは複数のディスク上にある物理スライスの集まり。論理デバイスは、Solaris ボリュームマネージャーではボリュームと呼ばれます。ボリュームは、アプリケーションやファイルシステムにとって物理ディスクと同じように機能します。

# 索引

---

## B

boot: cannot open /kernel/unix メッセージ, 184  
bootparams ファイル, 更新, 189

## C

Can't boot from file/device メッセージ, 184  
CHANGE DEFAULT BOOT DEVICE メッセージ, 190  
CLIENT MAC ADDR エラーメッセージ, 189  
clock gained xxx days メッセージ, 184

## E

/etc/bootparams ファイル, JumpStart ディレクトリア  
クセスの有効化, 189

## G

GRUB ベースのブート, menu.lst ファイルの検  
出, 155

## L

le0: No carrier - transceiver cable problem メッセー  
ジ, 184  
Live Upgrade, 「Solaris Live Upgrade」を参照  
Live Upgrade ブート環境を比較する, 145

## M

menu.lst ファイル, 検出, 155

## N

No carrier - transceiver cable problem メッセージ, 184  
Not a UFS filesystem メッセージ, 184

## R

RAID-0 ボリューム, 説明, 29  
RAID-1 ボリューム, Solaris Live Upgrade, 説明, 29  
RAID-1 ボリューム (ミラー), Solaris Live Upgrade  
Solaris ボリュームマネージャーボリュームへの移  
行例, 172  
作成およびアップグレードの例, 168  
作成の例, 88, 89, 90  
説明, 27, 29  
要件, 49  
RPC Timed out メッセージ, 189

## S

Solaris Live Upgrade  
RAID-1 ボリューム (ミラー) 用スライスの選  
択, 49  
アップグレード  
作業マップ, 95-96  
ブート環境, 96  
アップグレードの障害回復, 129

## Solaris Live Upgrade (続き)

- インストール
  - Solaris フラッシュアーカイブ, 112
  - Solaris フラッシュアーカイブでプロファイルを使用, 118
  - パッケージ, 59
- キーワード
  - プロファイル, 106, 107
- 起動, 63
- コマンド, 179
- 作成
  - RAID-1 ボリューム(ミラー)、説明, 27
  - 作業マップ, 59
  - ブート環境、作業, 64
  - ブート環境、説明, 22
- 処理を取り消す, 144
- 説明, 20
- 停止, 63
- ディスク容量の要件, 45
- 内容のカスタマイズ, 52
- 必要なパッケージ, 43
- 表示
  - ブート環境の構成, 153
  - リモートシステムの画面, 56
  - ファイルシステムの構成, 65
  - ファイルシステムのスライス, 66
  - ファイルへ出力, 66
  - ブート環境のアクティブ化, 119
  - ブート環境の削除, 146
  - ブート環境の名前の表示, 148
  - ブート環境の名前の変更, 149
  - ブート環境を比較する, 145
  - プロファイル、差分アーカイブ用の例, 109
  - プロファイル、例, 109
  - ボリュームのキーワード, 87
- 例, 161
  - RAID-1 ボリュームのアップグレード, 168, 172
  - RAID-1 ボリュームの作成, 88, 89
  - 完全なプロセス, 161
  - 内容のカスタマイズ, 93
  - ミラーの作成, 90
- Solaris Live Upgrade 処理を取り消す, 144
- Solaris Live Upgrade のコマンド, 179
- Solaris Live Upgrade ブート環境の名前の変更, 149
- Solaris ボリュームマネージャー
  - Solaris Live Upgrade で使用されるコマンド, 50

## Solaris ボリュームマネージャー (続き)

- Solaris Live Upgrade の例
  - RAID-1 ボリュームの切り離しとアップグレード, 168
  - RAID-1 ボリュームへの移行, 172

## T

- timed out RPC エラー, 189
- transceiver cable problem メッセージ, 184

## U

- Unknown client error メッセージ, 183

## W

- WARNING: CHANGE DEFAULT BOOT DEVICE, 190
- WARNING: clock gained xxx days メッセージ, 184

## あ

## アーカイブ

- Solaris Live Upgrade ブート環境へのインストール, 112
- インストールの例, 35
- 空のブート環境の作成, 83

## アップグレード

- Solaris Live Upgrade
  - アップグレードの障害回復, 129
  - ガイドライン, 96
  - 作業, 96
  - 説明, 33
  - 手順, 112
  - 例, 161, 168, 172
- Solaris Update リリースへの, 213-215
- アップグレードの失敗, 195
- アップグレードの失敗, リブートの問題, 195
- アップグレードの障害, Solaris Live Upgrade による回復, 129

## き

## キーワード

## Solaris Live Upgrade

プロファイル, 106, 107

ボリューム, 87

共有可能ファイルシステム, 説明, 22

## く

クリティカルファイルシステム, 説明, 22

## け

計画, Solaris Live Upgrade, 41

## こ

構成, Solaris Live Upgrade のファイルシステム, 65

コピー, Solaris Live Upgrade のファイルシステム, 142

## さ

削除, Live Upgrade ブート環境, 146

## 作成

## Solaris Live Upgrade

ブート環境, 作業, 64, 72, 74

ブート環境, 説明, 22

サブミラー, 説明, 29

## し

状態データベース, 説明, 29

## す

ステータス, ブート環境の表示, 140

## スライス

## Solaris Live Upgrade

選択のための指針, 48

ファイルシステムのカスタマイズ, 66

## スワップファイルシステム

## Solaris Live Upgrade

カスタマイズ, 68

スライスを選択するための指針, 51

## て

## ディスク容量

## 要件

Solaris Live Upgrade, 45

テスト, Solaris Live Upgrade, プロファイル, 110

## と

トークンリングカード, ブートエラー, 188

## トラブルシューティング

DHCP によるネットワークブート, 189

インストールの一般的な問題

DHCP によるネットワークブート, 189

システムのブート, 189

間違ったサーバーからのブート, 189

## は

## パッケージ

## Solaris Live Upgrade

追加, 46, 97

要件, 203

カスタム JumpStart を使用する要件, 203

## パッチ

Solaris Live Upgrade で追加, 46, 97

パッチレベルの確認, 43, 60

パッチアナライザ, 213-215

## ひ

表示, Solaris Live Upgrade ブート環境の名前, 148

## ふ

### ファイルとファイルシステム

#### Solaris Live Upgrade

RAID-1 ボリューム (ミラー) の作成、説明, 27

カスタマイズ, 65

サイズの見積もり, 45

作成のための指針, 47

スライスを選択するための指針, 48

説明, 22

ブート環境間でのファイルシステムの共有, 51

### ブート環境、Solaris Live Upgrade

構成の表示, 153

失敗、説明, 37

### ブート環境のアクティブ化

作業, 119

失敗、説明, 37

説明, 36

ファイルの、説明, 53

### プロファイル

#### Solaris Live Upgrade

差分アーカイブ用の例, 109

例, 109

### プロファイルキーワード

`forced_deployment`

説明と値, 108

`local_customization`

説明と値, 108

## ほ

### ボリューム

RAID-0、説明, 29

RAID-1、説明, 29

ボリュームマネージャー、「Solaris ボリュームマネージャー」を参照

## み

ミラー、「RAID-1 ボリューム」を参照

## よ

要件, Live Upgrade の使用, 41

## る

ルートファイルシステム, 非アクティブブート環境  
のパッケージ要件, 203

## れ

連結、説明, 29