

Oracle® Solaris 11.2 でのファイルシステムの 管理

ORACLE®

Part No: E53911
2014 年 7 月

Copyright © 2004, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

目次

このドキュメントの使用	7
1 ファイルシステムの管理について	9
ファイルシステムの概要	9
Oracle Solaris ファイルシステムのタイプ	10
デフォルトの Oracle Solaris ファイルシステム	16
ファイルシステムのマウントおよびマウント解除の概要	18
マウントされたファイルシステムテーブル	18
仮想ファイルシステムテーブル	19
NFS 環境	20
ブート環境間でのファイルシステムデータの共有	21
自動マウント (autofs)	21
Oracle Solaris SMB サービス	22
ファイルシステムのタイプを調べる	23
ファイルシステムモニタリングツール (fsstat)	24
ファイルシステム管理タスクについての参照先	24
2 ファイルシステムの作成およびマウント	27
Oracle Solaris ファイルシステムの作成およびマウント	27
▼ ZFS ファイルシステムの作成方法	27
▼ レガシー UFS ファイルシステムを作成およびマウントする方法	28
▼ TMPFS ファイルシステムを作成およびマウントする方法	29
▼ LOFS ファイルシステムを作成およびマウントする方法	30
Oracle Solaris ファイルシステムのマウントおよびマウント解除	32
▼ /etc/vfstab ファイルにエントリを追加する方法	36
▼ 1 つのファイルシステム (/etc/vfstab ファイル) をマウントする方法	37
▼ NFS ファイルシステムのマウント方法 (mount コマンド)	38
▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方 法 (mount コマンド)	39
▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法	40

▼ 1 つのファイルシステムをマウント解除する方法	41
3 追加スワップ空間の構成	43
スワップ空間について	43
スワップ空間と仮想メモリー	43
スワップ空間と TMPFS ファイルシステム	44
スワップ空間とダンプデバイスの構成	45
スワップ空間と動的再構成	46
SAN 環境でのスワップ空間の構成	46
スワップ空間の追加が必要かどうかを調べる方法	47
スワップ関連のエラーメッセージ	47
TMPFS 関連のエラーメッセージ	47
スワップ空間の割り当て方法	48
スワップ空間と /etc/vfstab ファイル	48
スワップ空間の計画	48
ZFS ベースのシステムでのスワップ空間の割り当て	49
スワップリソースのモニタリング	50
Oracle Solaris ZFS ルート環境でのスワップ空間の追加または変更	51
▼ Oracle Solaris ZFS ルート環境でスワップ空間を追加する方法	52
4 ファイルとファイルシステムのコピー	55
ファイルシステムをコピーするためのコマンド	55
cpio を使用してファイルシステム間でディレクトリをコピーする	57
▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)	57
ファイルとファイルシステムをテープにコピーする	58
tar を使用してファイルをテープにコピーする	59
▼ ファイルをテープにコピーする方法 (tar)	59
▼ テープ上のファイルのリストを表示する方法 (tar)	60
▼ テープからファイルを取り出す方法 (tar)	61
pax コマンドを使用してファイルをテープにコピーする	62
▼ ファイルをテープにコピーする方法 (pax)	62
cpio コマンドを使用してファイルをテープにコピーする	63
▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)	63
▼ テープ上のファイルのリストを表示する方法 (cpio)	64
▼ テープからすべてのファイルを取り出す方法 (cpio)	65
▼ テープから特定のファイルを取り出す方法 (cpio)	66
ファイルをリモートテープデバイスにコピーする	67
▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd)	67

▼ ファイルをリモートテープデバイスから抽出する方法	68
索引	71

このドキュメントの使用

- **概要** - Oracle Solaris の概要を説明します。1 つ以上のファイルシステムを管理し、ファイルシステム管理タスクを実行する方法に関する情報が含まれます。
- **対象読者** - システム管理者。
- **前提知識** - Oracle Solaris または UNIX の基本的なシステム管理の経験、および一般的なファイルシステム管理の経験。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=E36784>) に含まれています。

Oracle サポートへのアクセス

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

フィードバック

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

◆◆◆ 第 1 章

ファイルシステムの管理について

ファイルシステムの管理は、もっとも重要なシステム管理タスクの 1 つです。

この章の内容は以下のとおりです。

- 9 ページの「ファイルシステムの概要」
- 16 ページの「デフォルトの Oracle Solaris ファイルシステム」
- 18 ページの「ファイルシステムのマウントおよびマウント解除の概要」
- 23 ページの「ファイルシステムのタイプを調べる」
- 24 ページの「ファイルシステムモニタリングツール (fsstat)」
- 24 ページの「ファイルシステム管理タスクについての参照先」

ファイルシステムの概要

ファイルシステムは、ファイルを編成して格納するためのディレクトリ構造です。

「ファイルシステム」という用語は、次のような場合に使用されます。

- 特定の種類のファイルシステム (ディスクベース、ネットワークベース、または仮想) を指す場合
- ルートディレクトリ (/) から始まるファイルツリー全体を指す場合
- ディスクスライスやほかの記憶メディアデバイスのデータ構造を指す場合
- ファイルツリー構造のうち、ファイルがアクセスできるように主なファイルツリー上のマウントポイントに接続されている部分を指す場合

通常、その意味は状況に応じて判断できます。

Oracle Solaris OS は、各種ファイルシステムへの標準インタフェースを提供する仮想ファイルシステム (VFS) アーキテクチャーを使用します。VFS アーキテクチャーによって、カーネルはファ

イルの読み取り、書き込み、一覧表示などの基本操作を処理できます。また、VFS アーキテクチャーにより新しいファイルシステムの追加も簡単になります。

Oracle Solaris ファイルシステムのタイプ

Oracle Solaris OS では、次の 3 種類のファイルシステムがサポートされます。

- ディスクベース
- ネットワークベース
- 仮想

ファイルシステムのタイプを確認するには、[23 ページの「ファイルシステムのタイプを調べる」](#)を参照してください。

Oracle Solaris のディスクベースのファイルシステム

「ディスクベースのファイルシステム」は、ハードディスクや DVD などの物理メディアに格納されます。ディスクベースのファイルシステムは、さまざまな形式で作成できます。次の表で、作成できる形式について説明します。

ディスクベースのファイルシステム	形式の説明
ZFS	ZFS はデフォルトのディスクベースのルートファイルシステムです。
UFS	レガシー UNIX ファイルシステム (4.3 Tahoe リリースで提供されていた BSD Fat Fast ファイルシステムに基づく)。
PxFS	Oracle Solaris Cluster ソフトウェアは、Oracle Solaris Cluster Proxy File System (PxFS) に基づくクラスタファイルシステムを提供します。
SAM-QFS	SAM-QFS は、統合された階層型ストレージマネージャー (HSM) およびストレージエリアネットワーク (SAN) ファイルシステムです。SAM は、HSM のストレージおよびアーカイブ管理コンポーネントです。QFS は、SAN のスケーラブルな高パフォーマンスファイルシステムコンポーネントです。SAM-QFS には、ディスクボリューム管理およびテープボリューム管理も統合されています。QFS には、ライトワンスリードメニー (WORM) ファイルシステム機能もあります。QFS は、ファイルシステムが必要な際に、SAM とは独立して使用できます。SAM は QFS を必要としており、QFS とは独立して使用できません。
HSFS	High Sierra, Rock Ridge、および ISO 9660 のファイルシステム。High Sierra は、はじめての CD-ROM ファイルシステムです。ISO 9660 は、High Sierra ファイルシステムの公式の標準バージョンです。HSFS ファイルシステムは CD-ROM 上で使用される読み取り専用ファイルシステムです。Oracle Solaris HSFS では ISO 9660 の Rock Ridge 拡張がサポートされます。CD-ROM 上に存在する場

ディスクベースのファイルシステム	形式の説明
	合、これらの拡張機能は、書き込み可能性およびハードリンクを除くすべてのファイルシステム機能およびファイルタイプを提供します。
PCFS	PC ファイルシステム。DOS ベースのパーソナルコンピュータ用に作成された DOS フォーマットのディスク上のデータとプログラムに読み取りと書き込みのアクセスが可能です。
UDFS	UDFS (Universal Disk Format) ファイルシステム。DVD (Digital Versatile Disc または Digital Video Disc) と呼ばれる光学式メディアテクノロジーに情報を格納するための業界標準形式です。

ディスクベースの各種ファイルシステムは、次のように特定のメディアのタイプに対応しています。

- ZFS または UFS とハードディスク
- HSFS と CD-ROM
- PCFS と USB フロッピーディスク
- UDF と DVD
- SAM-QFS とハードディスクまたはテープ
- PxFs とハードディスク - クラスタファイルシステムが高可用であるには、基になるディスクストレージが複数の Oracle Solaris ホストに接続されている必要があります。そのため、クラスタファイルシステム内に構築されたローカルファイルシステム (ホストのローカルディスク上に格納されたファイルシステム) は高可用ではありません。詳細は、『[Oracle Solaris Cluster Concepts Guide](#)』の「[Cluster File Systems](#)」を参照してください。

ただし、これらの対応関係は制限的なものではありません。たとえば、DVD 上に ZFS または UFS ファイルシステムを作成できます。

UDFS (Universal Disk Format) ファイルシステム

リムーバブルメディア上で UDFS ファイルシステムを作成する方法については、『[Oracle Solaris 11.2 でのデバイスの管理](#)』の「[リムーバブルメディア上にファイルシステムを作成する方法](#)」を参照してください。

UDF ファイルシステムは、「*DVD*」(Digital Versatile Disc または Digital Video Disc) 光学式メディアに情報を格納するための業界標準形式です。

UDF ファイルシステムは、SPARC と x86 の両方のプラットフォームにおいて、動的にロード可能な 32 ビットと 64 ビットのモジュールとして提供されます。また、ファイルシステムを作成、マウント、および検査するシステム管理ユーティリティも同時に提供されます。Oracle Solaris

の UDF ファイルシステムは、サポートされている ATAPI と SCSI の DVD ドライブ、CD-ROM デバイス、およびディスクドライブで機能します。さらに、Oracle Solaris の UDF ファイルシステムは UDF 1.50 仕様に完全に準拠しています。

UDF ファイルシステムには次のような機能があります。

- UDF ファイルシステムが入っている業界標準の CD-ROM や DVD-ROM のメディアにアクセスできます
- さまざまなプラットフォームやオペレーティングシステムで情報を交換できます
- UDF 形式に基づく DVD ビデオ仕様を使用することで、放送品質並みの映像、高品質のサウンド、すぐれた対話性という特長を備えた新しいアプリケーションを実装できます

次の機能は、UDF ファイルシステムにはありません。

- 書き込み可能なメディア (CD-RW) へのディスクアットワンス (Disk At Once) 記録、およびインクリメンタル記録

次に、UDF ファイルシステムの要件を示します。

- SPARC または x86 プラットフォームがサポートされていること
- CD-ROM または DVD-ROM ドライブがサポートされていること

Oracle Solaris で実装された UDF ファイルシステムには、次のような互換性があります。

- 業界標準の読み取り / 書き込み UDF Version 1.50 のサポート
- 完全に国際化されたファイルシステムのユーティリティ

ネットワークベースのファイルシステム

「ネットワークベースのファイルシステム」は、ネットワークからアクセスされるファイルシステムです。ネットワークベースのファイルシステムは通常、1 つのシステム (通常はサーバー) 上にあり、ほかのシステムからネットワーク経由でアクセスされます。

NFS サービスで分散されたリソース (ファイルやディレクトリ) を提供するには、サーバーからそれらのリソースを共有して個々のクライアントでマウントします。詳細は、[20 ページの「NFS 環境」](#)を参照してください。

Oracle SMB サービスで分散されたリソース (ファイルやディレクトリ) を Windows および Mac OS システムに提供するには、サーバーからそれらのリソースを共有して個々のクライアントでマウントします。詳細は、[22 ページの「Oracle Solaris SMB サービス」](#)を参照してください。

仮想ファイルシステム

「仮想ファイルシステム」は、特殊なカーネル情報と機能へのアクセスを提供するメモリーベースのファイルシステムです。ほとんどの仮想ファイルシステムは、ディスク領域を使用しません。また、一時ファイルシステム (TMPFS) などの一部の仮想化ファイルシステムは、ディスク上のスワップ空間を使用します。

一時ファイルシステム

一時ファイルシステム (TMPFS) は、ファイルシステムの読み取りと書き込みにローカルメモリーを使用します。TMPFS を使用すると、ローカルディスク上で、あるいはネットワーク経由で一時的ファイルの読み書きを行う際のオーバーヘッドを軽減でき、システムパフォーマンスを改善できます。一時ファイルは、たとえば、プログラムのコンパイル時に作成されます。OS は、一時ファイルを操作しているとき、多くのディスク処理またはネットワーク処理を行います。TMPFS を使用してこれらの一時ファイルを格納すると、その作成、処理、または削除が大幅に高速になります。

TMPFS ファイルシステムのファイルは、永続的に保存されるわけではありません。これらのファイルは、ファイルシステムのマウントが解除されるとき、システムがシャットダウンまたはリブートされるときに削除されます。

TMPFS は、Oracle Solaris OS 内の `/tmp` ディレクトリのデフォルトのファイルシステムタイプです。ZFS または UFS ファイルシステムの場合と同様に、`/tmp` ディレクトリとの間でファイルをコピーまたは移動できます。TMPFS ファイルシステムは、一時的な退避場所としてスワップ空間を使用します。

TMPFS ファイルシステムがマウントされたシステムのスワップ空間が足りないと、次の 2 つの問題が発生する可能性があります。

- TMPFS ファイルシステムは、通常のファイルシステムと同様に容量不足になる可能性があります。
- TMPFS はスワップ空間を割り当ててファイルのデータを保存するので (必要な場合)、一部のプログラムがスワップ空間不足のために実行できなくなる可能性があります。

TMPFS ファイルシステムの作成方法については、[第2章「ファイルシステムの作成およびマウント」](#)を参照してください。スワップ空間を追加する方法については、[第3章「追加スワップ空間の構成」](#)を参照してください。

ループバックファイルシステム

ループバックファイルシステム (LOFS) を使用すると、代替パス名を使用してファイルにアクセスできるように、新しい仮想ファイルシステムを作成できます。たとえば、`/tmp/newroot` 上にルート (`/`) ディレクトリのループバックマウントを作成できます。このループバックマウントでは、NFS サーバーからマウントされたファイルシステムを含むファイルシステム階層全体が、`/tmp/newroot` の下に複製されたように見えます。どのファイルにも、ルート (`/`) で始まるパス名または `/tmp/newroot` で始まるパス名を使用してアクセスできます。

LOFS ファイルシステムを作成する方法については、[第2章「ファイルシステムの作成およびマウント」](#)を参照してください。

プロセスファイルシステム

プロセスファイルシステム (PROCFS) はメモリー内に存在し、`/proc` ディレクトリ内にアクティブなプロセスのプロセス番号別リストが格納されます。`/proc` ディレクトリの内容は、`ps` などのコマンドで使用されます。デバッガやほかの開発ツールも、ファイルシステムコールを使用して、プロセスのアドレス空間にアクセスできます。



注意 - `/proc` ディレクトリ内のファイルは削除しないでください。`/proc` ディレクトリからプロセスを削除しても、そのプロセスは強制終了されません。`/proc` ファイルはディスク容量を消費しないため、このディレクトリからファイルを削除しても無意味です。

`/proc` ディレクトリは、管理が不要です。

その他の仮想ファイルシステム

次のタイプの仮想ファイルシステムは、参考のために掲載してあります。管理は不要です。

仮想ファイルシステム	説明
CTFS	CTFS (契約ファイルシステム) は、契約の作成、制御、および監視のためのインタフェースです。契約は、豊富なエラー報告機能とリソースの削除を延期する手段 (オプション) を提供することにより、プロセスと、このプロセスが依存するシステムとの関係を拡張します。 サービス管理機能 (SMF) は、プロセス契約 (契約の一種) を使用して、サービスを構成するプロセス群を追跡します。このため、マルチプロセスサービスの一部分での障害をそのサービスの障害として識別できます。

仮想ファイルシステム	説明
FIFOFS (先入れ先出し)	プロセスにデータへの共通アクセスを提供する名前付きパイプファイル。
FDFS (ファイル記述子)	ファイル記述子を使用してファイルを開くための明示的な名前を提供します。
MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供します。
NAMEFS	ファイル記述子をファイルの先頭に動的にマウントするために、主に STREAMS によって使用されます。
OBJFS	OBJFS (オブジェクト) ファイルシステムは、現在カーネルによってロードされているすべてのモジュールの状態を説明します。デバッガは、この仮想ファイルシステムを使って、カーネルに直接アクセスすることなくカーネルシンボルの情報にアクセスできます。
SHAREFS	ローカルシステムに、共有ファイルシステムのテーブルへの読み取り専用アクセスを提供します。
SPECFS (特殊)	キャラクタ型の特殊デバイスおよびブロック型デバイスへのアクセスを提供します。
SWAPFS	スワップ用にカーネルによって使用されます。

拡張ファイル属性

ZFS、UFS、NFS、および TMPFS ファイルシステムは、拡張ファイル属性を含むように機能拡張されました。アプリケーション開発者は、拡張ファイル属性を使って、ファイルに特定の属性を関連付けることができます。たとえば、ウィンドウシステムの管理アプリケーションの開発者は、表示アイコンとファイルを関連付けることができます。拡張ファイル属性は、論理的には、ターゲットファイルに関連付けられている隠しディレクトリ内のファイルとして表されます。

属性を追加し、拡張属性の名前空間内に入っているシェルコマンドを実行するには、`runat` コマンドを使用します。拡張属性の名前空間とは、特定のファイルに関連付けられた、非表示の属性ディレクトリです。

`runat` コマンドを使用して属性をファイルに追加するには、最初に属性ファイルを作成する必要があります。

```
$ runat filea cp /tmp/attrdata attr.1
```

次に、`runat` コマンドを使用して、ファイルの属性をリストに表示します。

```
$ runat filea ls -l
```

詳細は、[runat\(1\)](#)を参照してください。

属性認識オプションの追加により、多くの Oracle Solaris ファイルシステムコマンドがファイルシステム属性をサポートするようになりました。属性認識オプションを使って、ファイル属性を照会したり、コピーしたり、検索したりできます。詳細は、各ファイルシステムコマンドのマニュアルページを参照してください。

スワップ空間

Oracle Solaris OS は、一部のディスクスライスをファイルシステムではなく一時ストレージとして使用します。これらのスライスを「スワップスライス」または「スワップ空間」と呼びます。スワップ空間は、現在のプロセスを処理するだけの十分な物理メモリーがシステムにない場合に、仮想メモリー記憶域として使用されます。

多くのアプリケーションは十分なスワップ空間が使用できることを前提に作成されているため、スワップ空間を割り当て、その使われ方をモニターして、必要に応じてスワップ空間を追加する方法を知っておく必要があります。スワップ空間の概要とスワップ空間の追加方法については、[第3章「追加スワップ空間の構成」](#)を参照してください。

デフォルトの Oracle Solaris ファイルシステム

革新的な新しいファイルシステムである Oracle Solaris ZFS は、管理の簡素化、トランザクションのセマンティクス、エンドツーエンドのデータの整合性、および大きなスケーラビリティを提供します。

ZFS ファイルシステムは階層構造になっており、ルートディレクトリ (/) から始まり、下位に多数のディレクトリが続いています。Oracle Solaris のインストールプロセスは、デフォルトのディレクトリセットをインストールし、一連の規則を適用して類似するタイプのファイルをグループ化します。

さらに、ZFS は次の管理機能を提供します。

- デバイス管理のサポート
- 持続的なスナップショットの作成と複製の機能
- ファイルシステムの割り当て制限の設定機能
- ACL ベースのアクセス制御
- ストレージプールによるファイルシステムの容量の予約機能
- ゾーンがインストールされた Oracle Solaris システムのサポート

ZFS ファイルを使用する方法の詳細は、『[Oracle Solaris 11.2 での ZFS ファイルシステムの管理](#)』を参照してください。

Oracle Solaris のファイルシステムやディレクトリの簡単な概要については、[filesystem\(5\)](#)を参照してください。

次の表は、デフォルト Oracle Solaris ファイルシステムのサマリーを示しています。

表 1-1 デフォルトの Oracle Solaris ファイルシステム

ファイルシステムまたはディレクトリ	ファイルシステムのタイプ	説明
ルート (/)	ZFS	階層ファイルツリーの最上位。ルート (/) ディレクトリには、カーネル、デバイスドライバ、システムのブートに使用されるプログラムなど、システム処理に欠かせないディレクトリとファイルが入っています。また、ローカルとリモートのファイルシステムをファイルツリーに接続できるマウントポイントディレクトリも入っています。
/usr	ZFS	ほかのユーザーと共有できるシステムファイルとディレクトリ。特定のタイプのシステム上でのみ実行できるファイルは、/usr ディレクトリにあります (SPARC 実行可能ファイルなど)。どのタイプのシステム上でも使用できるファイル (マニュアルページなど) は、/usr/share ディレクトリに配置できます。
/export/home または /home	NFS または ZFS	ユーザーのホームディレクトリのマウントポイント。ホームディレクトリには、そのユーザーの作業ファイルが格納されます。デフォルトでは、/home ディレクトリは自動マウントされるファイルシステムです。
/var	ZFS	ローカルシステムの使用中に変化または拡大する可能性のあるシステムファイルとディレクトリ。これには、システムログ (vi や ex のバックアップファイルなど) が含まれます。
/opt	NFS または ZFS	オプションの他社製のソフトウェア製品のマウントポイント。一部のシステムでは、/opt ディレクトリが UFS ファイルシステムまたは ZFS ファイルシステムである場合があります。
/tmp	TMPFS	システムがブートされるか、/tmp ファイルシステムがマウント解除されるたびに削除される一時ファイル。
/proc	PROCFS	アクティブなプロセスのプロセス番号別リスト。
/etc/mnttab	MNTFS	ローカルシステムに、マウント済みファイルシステムのテーブルへの読み取り専用アクセスを提供する仮想ファイルシステム。
/system/volatile	TMPFS	システムのブート後は不要になる一時ファイルを格納するメモリーベースのファイルシステム。
/system/contract	CTFS	契約情報を保持する仮想ファイルシステム。
/system/object	OBJFS	カーネルに直接アクセスすることなくカーネルシンボルの情報にアクセスする際にデバッグによって使用される仮想ファイルシステム。

ファイルシステムのマウントおよびマウント解除の概要

ファイルシステム上のファイルにアクセスするには、ファイルシステムをマウントする必要があります。ファイルシステムのマウントとは、ファイルシステムを特定のディレクトリ (マウントポイント) に接続し、システムで使用可能にすることです。ルート (/) ファイルシステムは、常にマウントされています。ほかのファイルシステムは、ルート (/) ファイルシステムに接続したり切り離したりできます。

ほとんどのファイルシステムは、システムブート時に SMF サービスにより自動的にマウントされます。一般に、ファイルシステムのマウントやマウント解除を手動で行う必要はありません。さまざまなファイルシステムタイプをマウントする方法の詳細については、[32 ページの「Oracle Solaris ファイルシステムのマウントおよびマウント解除」](#)を参照してください。

ファイルシステムをマウントすると、そのファイルシステムがマウントされている間は、マウントポイントのディレクトリ内に実際に存在しているファイルやディレクトリは使用できなくなります。これらのファイルは、永続的にマウントプロセスの影響を受け続けるわけではありません。ファイルシステムのマウントが解除されると、再び使用可能な状態になります。ただし、通常は存在するがアクセスできないファイルは混乱の原因となるので、通常マウントディレクトリは空になります。

ファイルシステムのマウント手順については、[32 ページの「Oracle Solaris ファイルシステムのマウントおよびマウント解除」](#)を参照してください。

マウントされたファイルシステムテーブル

ファイルシステムをマウントまたはアンマウントすると、現在マウントされているファイルシステムのリストを使用して、`/etc/mnttab` (マウントテーブル) ファイルが変更されます。このファイルの内容を表示するには、`cat` または `more` コマンドを使用します。ただし、このファイルを編集することはできません。次に `/etc/mnttab` ファイルの例を示します。

```
$ more /etc/mnttab
rpool/ROOT/zfsBE /      zfs      dev=3390002  0
/devices         /devices  devfs     dev=8580000  1337114941
/dev             /dev     dev      dev=85c0000  1337114941
ctfs             /system/contract  ctfs     dev=8680001  1337114941
proc            /proc    proc     dev=8600000  1337114941
mnttab          /etc/mnttab  mntfs   dev=86c0001  1337114941
swap           /system/volatile  tmpfs   xattr,dev=8700001  1337114941
objfs          /system/object  objfs   dev=8740001  1337114941
sharefs        /etc/dfs/sharetab  sharefs dev=8780001  1337114941
```

```

/usr/lib/libc/libc_hwcaps2.so.1 /lib/libc.so.1 lofs dev=3390002 13371149
fd /dev/fd fd rw,dev=8880001 1337114969
rpool/ROOT/zfsBE/var /var zfs rw,devices,
setuid,nonbmand,exec,
rstchown,xattr,atime,dev=3390003 1337114969
swap /tmp tmpfs xattr,dev=8700002 1337114969
rpool/VARSHARE /var/share zfs rw,devices,setuid,nonbmand,exec,
rstchown,xattr,atime,dev=3390004 1337114969

```

仮想ファイルシステムテーブル

ほとんどのファイルシステムは、システムブート時に SMF サービスにより自動的にマウントされます。

レガシーまたはリモートファイルシステムをマウントしたり、ZFS スワップボリュームを変更したりする場合は、`/etc/vfstab` ファイルの編集が必要ことがあります。ZFS スワップボリュームの変更については、[第3章「追加スワップ空間の構成」](#)を参照してください。

レガシーまたはリモートファイルシステムをマウントするためのエントリを追加するには、次の情報を指定する必要があります。

- ファイルシステムが存在するデバイスまたは NFS サーバー
- ファイルシステムのマウントポイント
- ファイルシステムのタイプ
- システムのブート時に、`mountall` コマンドを使ってファイルシステムを自動的にマウントするかどうか
- マウントオプション

次に、ZFS ルートファイルシステムがあるシステムの `vfstab` の例を示します。さらに、システムは、NFS サーバー `neo` からリモートファイルシステム `/users/data` をマウントしています。

```

# cat /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot  options
#
fd           -           /dev/fd    fd      -         no       -
/proc       -           /proc      proc    -         no       -
/dev/zvol/dsk/rpool/swap -           -          swap    -         no       -
/devices    -           /devices   devfs   -         no       -
sharefs     -           /etc/dfs/sharetabsharefs -         no       -
ctfs       -           /system/contract ctfs    -         no       -
objfs      -           /system/object objfs   -         no       -
swap       -           /tmp       tmpfs   -         yes      -
neo:/users/data -           /data      nfs     -         yes      -

```

ZFS ファイルシステムは、ブート時に SMF サービスにより自動的にマウントされます。レガシーのマウント機能を使用し、`vfstab` に従って ZFS ファイルシステムをマウントできます。

`/etc/vfstab` の各フィールドの説明、およびこのファイルの編集方法と使用方法については、[36 ページの「/etc/vfstab ファイルにエントリを追加する方法」](#)を参照してください。

NFS 環境

NFS は、1 つのシステム (通常はサーバー) のリソース (ファイルやディレクトリ) をネットワーク上のほかのシステムと共有するための分散型ファイルシステムサービスです。たとえば、他社製のアプリケーションやソースファイルをほかのシステム上のユーザーと共有できます。

NFS は、リソースの実際の物理的な位置をユーザーが意識しなくてすむようにします。よく使用されるファイルのコピーをシステムごとに配置しなくても、あるシステムのディスク上にコピーを 1 つ配置することによって NFS は、ほかのすべてのシステムがそのコピーにネットワークからアクセスできるようにします。NFS の環境では、リモートファイルはローカルファイルと区別が付きません。

詳細は、『[Oracle Solaris 11.2 でのネットワークファイルシステムの管理](#)』を参照してください。

システムは、ネットワーク上で共有するリソースがあるときに、NFS サーバーになります。サーバーは、現在共有されているリソースとそのアクセス制限 (読み取り / 書き込み、読み取り専用アクセスなど) のリストを管理します。

リソースを共有する場合は、リモートシステムにマウントできるように、そのリソースを使用可能な状態にします。

リソースを共有するには、次の方法があります。

- ZFS の `share.nfs` プロパティを設定して ZFS 共有を作成します。例:

```
# zfs set share.nfs=on tank/home
```

- `share` コマンドを使用してレガシー共有を作成します。

```
# share -F nfs /ufsfs
```

NFS の詳細は、『[Oracle Solaris 11.2 でのネットワークファイルシステムの管理](#)』を参照してください。

NFS Version 4

Oracle Solaris リリースには、Oracle の実装の NFS version 4 分散ファイルアクセスプロトコルが含まれています。

NFS version 4 では、ファイルアクセス、ファイルロック、およびマウントプロトコルが 1 つのプロトコルに統合されるので、ファイアウォールの通過が容易になり、セキュリティが向上します。Oracle Solaris の NFS version 4 実装は、SEAM としても知られている Kerberos V5 と完全に統合されていますので、認証、整合性、およびプライバシーの機能を備えています。NFS version 4 を使用して、クライアントとサーバーとの間で使用するセキュリティの種類の変換を行うこともできます。NFS version 4 を実装しているサーバーは、さまざまなセキュリティ種類とファイルシステムに対応できます。

NFS Version 4 の機能の詳細は、『[Oracle Solaris 11.2 でのネットワークファイルシステムの管理](#)』を参照してください。

ブート環境間でのファイルシステムデータの共有

Oracle Solaris 11.1 リリースでは、異なるブート環境間での自動データ共有を可能にするメカニズムを使用できます。これらの共有ディレクトリは、`/var/share` にマウントされている `rpool/VARSHARE` ファイルシステムに格納されます。`/var` ディレクトリ内の共有データの配置により、すべてのブート環境に必要な領域の量が削減されます。

例:

```
# ls /var/share
audit cores crash mail
```

`/var` コンポーネントによってルートファイルシステムがいっぱいにならないようにすることを除いて、`/var/share` ファイルシステムは一般に管理を必要としません。

互換性のために、上記の `/var` コンポーネントから `/var/share` コンポーネントへのシンボリックリンクが自動的に作成されます。詳細については、[datasets\(5\)](#)を参照してください。

自動マウント (autofs)

NFS ファイルシステムリソースをマウントするには、「自動マウント」(*autofs*) というクライアント側のサービスを使用します。*autofs* サービスにより、システムから NFS リソースにアクセスする

たびに、これらのリソースを自動的にマウントしたりマウント解除したりできます。ユーザーがこのディレクトリ内で、このディレクトリに格納されているファイルを使用している間、ファイルシステムリソースはマウントされたままになります。リソースが一定の時間アクセスされなかった場合、リソースは自動的にマウント解除されます。

autofs サービスには、次の機能があります。

- システムブート時に NFS リソースをマウントする必要がないために、ブート時間が短くなります。
- NFS リソースをマウントまたはマウント解除するために、スーパーユーザーのパスワードを知っている必要はありません。
- NFS リソースは使用されるときにだけマウントされるために、ネットワークトラフィックが軽減されます。

autofs サービスは `automount` ユーティリティによって初期化されます。このコマンドは、システムのブート時に自動的に実行されます。`automountd` デーモンは永続的に動作し、必要に応じて NFS ファイルシステムをマウントまたはアンマウントします。デフォルトでは、`/home` ファイルシステムは `automount` デーモンによってマウントされます。

autofs では、同じファイルシステムを提供するサーバーを複数指定できます。このような方法では、1 つのサーバーがダウンしても、autofs がその他のマシンからファイルシステムのマウントを試みることができます。

autofs を設定および管理する方法の詳細は、『[Oracle Solaris 11.2 でのネットワークファイルシステムの管理](#)』を参照してください。

Oracle Solaris SMB サービス

Oracle Solaris OS は、サーバーメッセージブロック (SMB) プロトコルのサーバーおよびクライアント実装を提供します。これには、NT LM 0.12 や共通インターネットファイルシステム (CIFS) などの多数の SMB ダイアレクトのサポートが含まれます。CIFS と SMB という用語は、置き換え可能と考えることができます。

Solaris SMB サーバーを使用すると、ネイティブの Oracle Solaris システムが、ファイルシステム共有をマウントする SMB 対応クライアントに対してファイルを SMB 共有として提供できます。Windows、Mac OS、または Solaris クライアントは、Windows サーバーの場合と同様に、Solaris SMB サーバーと相互運用できます。Solaris SMB サーバーは、ワークグループモードでも、ドメインモードでも動作が可能です。ワークグループモードでは、共有リソースへのアクセ

スが要求されたときに Solaris SMB サーバーがユーザーのローカル認証を担当します。この認証プロセスは、ローカルログインと呼ばれます。ドメインモードでは、Solaris SMB サーバーはパススルー認証を使用します。パススルー認証では、ユーザー認証がドメインコントローラに委任されます。

詳細は、*Oracle Solaris 11.2* での *SMB ファイル共有と Windows 相互運用性の管理に関する説明*を参照してください。

ファイルシステムのタイプを調べる

これらのコマンドは、ファイルシステムがマウントされているかどうかにかかわらず動作します。

ディスクスライスの raw デバイス名がある場合は、`fstyp` または `df` コマンドを使用してファイルシステムのタイプを調べることができます。詳細については、[fstyp\(1M\)](#) または [df\(1M\)](#) を参照してください。

例 1-1 ファイルシステムのタイプを調べる方法

次の例は、`fstyp` コマンドを使ってファイルシステムのタイプを確認しています。

```
# fstyp /dev/rdisk/c0t0d0s0
zfs
```

次の例は、`df -n` コマンドを使用してシステムのファイルシステムタイプを表示します。

```
# df -n
/                : zfs
/devices        : devfs
/dev            : dev
/system/contract : ctfs
/proc           : proc
/etc/mnttab     : mntfs
/system/volatile : tmpfs
/system/object  : objfs
/etc/dfs/sharetab : sharefs
/dev/fd         : fd
/var            : zfs
/tmp           : tmpfs
/var/share     : zfs
/export        : zfs
/export/home   : zfs
/rpool         : zfs
/media/cdrom   : ufs
/media/cdrom-1 : ufs
/media/cdrom-2 : ufs
```

```

/media/cdrom-3      : ufs
/media/sol_10_811_sparc : hsfs
/media/cdrom-4      : ufs
/pond               : zfs
/pond/amy           : zfs
/pond/dr            : zfs
/pond/rory          : zfs

```

ファイルシステムモニタリングツール (fsstat)

Oracle Solaris 11 リリースからファイルシステムの動作を報告するための新しいファイルシステムモニタリングツール `fsstat` を使用できます。いくつかのオプションを使用して、マウントポイントごとまたはファイルシステムのタイプごとなどの活動を報告できます。

たとえば、次の `fsstat` コマンドは、ZFS モジュールがロードされてからのすべての ZFS ファイルシステムの操作を表示します。

```

$ fsstat zfs
new name name attr attr lookup rmdir read read write write
file remov chng get set ops ops ops bytes ops bytes
268K 145K 93.6K 28.0M 71.1K 186M 2.74M 12.9M 56.2G 1.61M 9.46G zfs

```

たとえば、次の `fsstat` コマンドは、`/export/ws` ファイルシステムがマウントされてからのすべてのファイルシステムの操作を表示します。

```

$ fsstat /export/ws
new name name attr attr lookup rmdir read read write write
file remov chng get set ops ops ops bytes ops bytes
0 0 0 18.1K 0 12.6M 52 0 0 0 0 /export/ws

```

デフォルトのフォームは、G バイト、K バイト、M バイトなど、理解しやすい値を使用して統計情報を報告します。

詳細は、[fsstat\(1M\)](#)を参照してください。

ファイルシステム管理タスクについての参照先

ファイルシステム管理の手順については、次を参照してください。

ファイルシステム管理タスク	参照先
新しいディスクデバイスを接続して構成します。	『Oracle Solaris 11.2 でのデバイスの管理』の第 4 章「Oracle Solaris でのディスクの管理」

ファイルシステム管理タスク	参照先
新規ファイルシステムを作成およびマウントします。	第2章「ファイルシステムの作成およびマウント」
リモートファイルをユーザーが利用できるようにします。	『Oracle Solaris 11.2 でのネットワークファイルシステムの管理』

◆◆◆ 第 2 章

ファイルシステムの作成およびマウント

この章では、ZFS ファイルシステム、ファイルシステム、一時ファイルシステム (TMPFS)、およびループバックファイルシステム (LOFS) を作成およびマウントする方法について説明します。TMPFS と LOFS は仮想ファイルシステムであるため、これらのファイルシステムを「実際に使用する」には、ファイルシステムをマウントします。さらに、レガシー UFS ファイルシステムの作成およびマウントについても説明します。

この章の内容は次のとおりです。

- [27 ページの「Oracle Solaris ファイルシステムの作成およびマウント」](#)
- [32 ページの「Oracle Solaris ファイルシステムのマウントおよびマウント解除」](#)

Oracle Solaris ファイルシステムの作成およびマウント

このセクションでは、Oracle Solaris ファイルシステムを作成およびマウントする方法の具体例を示します。

▼ ZFS ファイルシステムの作成方法

1. **管理者になります。**

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. **ZFS ストレージプールを作成します。**

次の例は、tank という名前の単純なミラー化ストレージプールと tank という名前の ZFS ファイルシステムを、1 つのコマンドで作成する方法を示しています。`/dev/dsk/c1t0d0` ディスク全体と `/dev/dsk/c2t0d0` ディスク全体を使用することを前提としています。

```
# zpool create tank mirror c1t0d0 c2t0d0
```

3. ZFS ファイルシステムを作成します。

```
# zfs create tank/fs
```

この新規 ZFS ファイルシステム tank/fs では、ディスク領域を必要なだけ使用でき、/tank/fs に自動的にマウントされます。

4. ファイルシステムが作成されたことを確認します。

```
# zfs list -r tank
NAME      USED  AVAIL  REFER  MOUNTPOINT
tank      117K  268G   21K    /tank
tank/fs   21K   268G   21K    /tank/fs
```

▼ レガシー UFS ファイルシステムを作成およびマウントする方法

始める前に 次の前提条件を満たしているかどうかを確認します。

- ディスクがフォーマットされ、スライスに分割されている。
- 既存のレガシー UFS ファイルシステムを再作成する場合は、そのマウントを解除します。
- ファイルシステムを格納するスライスのデバイス名がわかっている。

ディスクおよびディスクスライス番号の検索方法については、『[Oracle Solaris 11.2 でのデバイスの管理](#)』の第 6 章「システムのディスクの管理」を参照してください。

1. 管理者になります。

詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「割り当てられている管理権利の使用」を参照してください。

2. レガシー UFS ファイルシステムを作成します。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name
```

システムから、確認を促すプロンプトが表示されます。



注意 - この手順を実行する前に、スライスのデバイス名が正しく指定されていることを確認してください。間違ったスライスを指定すると、その内容は新しいファイルシステムの作成時に消去されます。そして、システムがパニックを起こす原因となる可能性があります。

- レガシー UFS ファイルシステムが作成されていることを確認するには、新しいファイルシステムを検査します。

```
# fsck /dev/rdisk/device-name
```

`device-name` 引数は、新しいファイルシステムを格納するディスクデバイスの名前を指定します。

`fsck` コマンドは、新しいファイルシステムの整合性を検査して、問題があれば通知し、問題を修復する前にプロンプトを表示します。`fsck` コマンドの詳細は、[fsck\(1M\)](#) を参照してください。

- レガシー UFS ファイルシステムをマウントします。

```
# mkdir /directory-name
# mount /dev/dsk/device-name /directory-name
```

例 2-1 レガシー UFS ファイルシステムの作成およびマウント

次の例は、UFS ファイルシステム `/dev/rdisk/c0t1d0s0` を作成して `/legacy` にマウントする方法を示しています。

```
# newfs /dev/rdisk/c0t1d0s0
newfs: construct a new file system /dev/rdisk/c0t1d0s0: (y/n)? y
/dev/rdisk/c0t1d0s0: 286722656 sectors in 46668 cylinders of 48 tracks, 128 sectors
140001.3MB in 2917 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
.....
super-block backups for last 10 cylinder groups at:
285773216, 285871648, 285970080, 286068512, 286166944, 286265376, 286363808,
286462240, 286560672, 286659104
# fsck /dev/rdisk/c0t1d0s0
# mkdir /legacy
# mount /dev/dsk/c0t1d0s0 /legacy
```

次の手順 ブート時にレガシー UFS ファイルシステムを自動的にマウントする場合は、作成後、[36 ページの「/etc/vfstab ファイルにエントリを追加する方法」](#)に進んでください。

▼ TMPFS ファイルシステムを作成およびマウントする方法

- 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 必要に応じて、TMPFS ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir /mount-point
```

/mount-point は、TMPFS ファイルシステムがマウントされるディレクトリです。

3. TMPFS ファイルシステムをマウントします。

```
# mount -F tmpfs [-o size=number] swap mount-point
```

-o size=number TMPFS ファイルシステムのサイズ制限を M バイト単位で示します。

mount-point TMPFS ファイルシステムがマウントされるディレクトリを示します。

ブート時に TMPFS ファイルシステムが自動的にマウントされるようにシステムを設定する方法については、[例2-3「ブート時に TMPFS ファイルシステムをマウントする」](#)を参照してください。

4. TMPFS ファイルシステムが作成されていることを確認します。

```
# mount -v
```

例 2-2 TMPFS ファイルシステムを作成およびマウントする

次の例は、TMPFS ファイルシステム /export/reports を作成およびマウントし、そのサイズを 50M バイトに制限する方法を示しています。

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50m swap /export/reports
# mount -v
```

例 2-3 ブート時に TMPFS ファイルシステムをマウントする

ブート時にシステムが自動的に TMPFS ファイルシステムをマウントするように設定するには、/etc/vfstab のエントリを追加します。次の例は、ブート時に /export/test を TMPFS ファイルシステムとしてマウントする /etc/vfstab ファイルのエントリを示しています。size=number オプションを指定していないため、/export/test の TMPFS ファイルシステムのサイズは利用できるシステムリソースによってのみ制限されます。

```
swap - /export/test tmpfs - yes -
```

▼ LOFS ファイルシステムを作成およびマウントする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 必要に応じて、LOFS ファイルシステムとしてマウントするディレクトリを作成します。

```
# mkdir loopback-directory
```

3. 新しく作成したディレクトリに対して、適切なアクセス権と所有権を設定します。

4. 必要に応じて、LOFS ファイルシステムをマウントするマウントポイントを作成します。

```
# mkdir /mount-point
```

5. LOFS ファイルシステムをマウントします。

```
# mount -F lofs loopback-directory /mount-point
```

loopback-directory ループバックマウントポイントにマウントするファイルシステムを指定します。

/mount-point LOFS ファイルシステムをマウントするディレクトリを指定します。

6. LOFS ファイルシステムがマウントされていることを確認します。

```
# mount -v
```

例 2-4 LOFS ファイルシステムを作成およびマウントする

次の例は、新しいソフトウェアを、実際にはインストールしないで、ループバックファイルシステムとして */new/dist* ディレクトリに作成、マウント、およびテストする方法を示しています。

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

例 2-5 ブート時に LOFS ファイルシステムをマウントする

ブート時にシステムが自動的に LOFS ファイルシステムをマウントするように設定するには、*/etc/vfstab* ファイルの最後にエントリを追加します。次の例は、ルート (*/*) ファイルシステムの LOFS ファイルシステムを */tmp/newroot* にマウントする */etc/vfstab* ファイルのエントリを示しています。

```
/ - /tmp/newroot lofs - yes -
```

ループバックファイルシステムのエントリーは、`/etc/vfstab` ファイル内の最後のエントリーでなければなりません。ループバックファイルシステムの `/etc/vfstab` エントリーが、そこに組み込まれるファイルシステムよりも前にあると、ループバックファイルシステムをマウントできません。

Oracle Solaris ファイルシステムのマウントおよびマウント解除

ZFS ファイルシステムのマウントやマウント解除は、自動的に行われます。レガシー UFS ファイルシステムを使用できるようにするには、マウントします。マウントしたファイルシステムは、システムのディレクトリツリー内の指定したマウントポイントに接続されます。ルート (`/`) ファイルシステムは、常にマウントされています。

次の表に、ファイルシステムをその用途に応じてマウントするためのガイドラインを示します。

必要なマウントの種類	推奨されるマウント方法
ローカルまたはリモートのファイルシステムをときどきマウントする。	コマンド行から手動で <code>mount</code> コマンドを入力します。
ローカルのレガシー UFS ファイルシステムを頻繁にマウントする。ローカルの ZFS ファイルシステムは、SMF サービスにより自動的にマウントされます。	<code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザー状態でブートされたときに、自動的にファイルシステムをマウントします。
ホームディレクトリなどのリモートのレガシー UFS ファイルシステムを頻繁にマウントする。	<ul style="list-style-type: none"> ■ <code>/etc/vfstab</code> ファイルを使用して、システムがマルチユーザーモードでブートされたときに、自動的にファイルシステムをマウントします。 ■ <code>autofs</code> を使用して、ユーザーがアクセスするときに自動的にマウントします。その後、ファイルシステムから別のディレクトリに移動するときに自動的にマウント解除します。

リムーバブルメディアのマウント方法については、『[Oracle Solaris 11.2 でのデバイスの管理](#)』の第 14 章「リムーバブルメディアの管理」を参照してください。

どのファイルシステムがすでにマウント済みであるかを調べるには、`mount` コマンドを使用します。

```
$ mount [ -v ]
```

`-v` は、マウントされているファイルシステムのリストを冗長モードで表示します。

例 2-6 どのファイルシステムがマウントされているかを調べる

この例は、`mount` コマンドを使用して、現在マウントされているファイルシステムに関する情報を表示する方法を示しています。

```
$ mount
/ on rpool/ROOT/zfsBE read/write/setuid/devices/rstchown/dev=3390002 on Tue ...
/devices on /devices read/write/setuid/devices/rstchown/dev=8580000 on Tue May 15 ...
/dev on /dev read/write/setuid/devices/rstchown/dev=85c0000 on Tue May 15 14:49:01 2012
/system/contract on ctfs read/write/setuid/devices/rstchown/dev=8680001 on Tue May 15 ...
/proc on proc read/write/setuid/devices/rstchown/dev=8600000 on Tue May 15 14:49:01 2012
/etc/mnttab on mnttab read/write/setuid/devices/rstchown/dev=86c0001 on Tue May 15
14:49:01 ...
/system/volatile on swap read/write/setuid/devices/rstchown/xattr/dev=8700001 on Tue May
15 ...
/system/object on objfs read/write/setuid/devices/rstchown/dev=8740001 on Tue May 15 ...
/etc/dfs/sharetab on sharefs read/write/setuid/devices/rstchown/dev=8780001 on Tue May 15 ...
/lib/libc.so.1 on /usr/lib/libc/libc_hwcaps2.so.1 read/write/setuid/devices/rstchown/dev ...
/dev/fd on fd read/write/setuid/devices/rstchown/dev=8880001 on Tue May 15 14:49:29 2012
/var on rpool/ROOT/zfsBE/var read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/tmp on swap read/write/setuid/devices/rstchown/xattr/dev=8700002 on Tue May 15 14:49:29 2012
/var/share on rpool/VARSHARE read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
```

この例は、`zfs mount` コマンドを使用して、現在マウントされている ZFS ファイルシステムに関する情報を表示する方法を示しています。

```
$ zfs mount
rpool/ROOT/zfsBE          /
rpool/ROOT/zfsBE          /var
rpool/VARSHARE            /var/share
tank/home                 /tank/home
```

/etc/vfstab ファイルのフィールドの説明

`/etc/vfstab` ファイル内のエントリには、次の表に示すように 7 つのフィールドがあります。

表 2-1 `/etc/vfstab` ファイルのフィールドの説明

フィールド名	説明
device to mount	このフィールドは、次のいずれかを指定します。 <ul style="list-style-type: none"> ■ ローカルのレガシー UFS ファイルシステム用のブロック型デバイス名 (<code>/dev/dsk/c8t1d0s7</code> など)。 ■ リモートファイルシステム用のリソース名 (<code>myserver:/export/home</code> など)。 リモートシステムリソースにエントリを追加した後に、次のサービスが有効になっていることを確認してください。

フィールド名	説明
	<pre># svcs -a grep nfs/client disabled May_14 svc:/network/nfs/client:default # svcadm enable svc:/network/nfs/client:default</pre> <p>NFS の詳細については、『Oracle Solaris 11.2 でのネットワークファイルシステムの管理』を参照してください。</p> <ul style="list-style-type: none"> ■ スワップボリューム (/dev/zvol/dsk/rpool/swap など)。 ■ 仮想ファイルシステム用のディレクトリ。
device to fsck	<p>「device to mount」フィールドで識別したレガシー UFS ファイルシステムに対応する raw (キャラクタ型) デバイス名 (/dev/rdisk/c8t1d0s7 など)。このフィールドによって、fsck コマンドが使用する raw インタフェースが決まります。読み取り専用ファイルシステムやリモートファイルシステムなど、適用できるデバイスがない場合は、ダッシュ (-) を使用します。</p>
mount point	レガシーまたはリモートファイルシステムをマウントする場所 (/data など) を指定します。
FS type	ファイルシステムのタイプを指定します。
fsck pass	<p>fsck コマンドがレガシー UFS ファイルシステムを検査するか決めるために使用するパス番号。このフィールドでダッシュ (-) を指定すると、ファイルシステムは検査されません。現在、/etc/vfstab ファイルの fsck pass 値はブートプロセスで無視されます。</p> <p>このフィールドにゼロが含まれるときは、レガシー UFS ファイルシステムは検査されません。このフィールドに 0 より大きい値が指定されている場合、UFS ファイルシステムは常に検査されます。</p> <p>このフィールドに 1 が指定されている場合、すべてのレガシー UFS ファイルシステムは vfstab ファイル内の順番どおりに 1 つずつ検査されます。このフィールドに 1 より大きな値が指定され、さらに preen (修復) オプション (-o p) が指定されている UFS ファイルシステムが複数ある場合、効率を最大限に高めるために、fsck コマンドは複数のディスク上のファイルシステムを自動的に並行して検査します。それ以外の場合、このフィールドの値は意味を持ちません。</p>
mount at boot	システムのブート時にファイルシステムが mountall コマンドによって自動的にマウントされるかどうかを yes または no で設定します。このフィールドは autofs とは連動していません。/proc や /dev/fd などのような仮想ファイルシステムの場合、このフィールドは常に no に設定するべきです。
mount options	ファイルシステムのマウントに使用されるオプションを (空白を入れずに) コマンドで区切ったリスト。オプションなしを示すにはダッシュ (-) を使用します。詳細は、 vfstab(4) を参照してください。

注記 - /etc/vfstab ファイル内の各フィールドには必ずエントリが必要です。フィールドに値を指定しない場合は、必ずダッシュ (-) を入力してください。ダッシュを入力しないと、システムが正常にブートしない可能性があります。同様に、フィールドの値に空白文字を使用しないでください。

Oracle Solaris ファイルシステムをマウント解除するための前提条件

ファイルシステムをマウント解除する場合の前提条件は次のとおりです。

- 管理者である必要があります。
- 使用中のファイルシステムはマウント解除できません。ユーザーがそのファイルシステム内のディレクトリにアクセスしているとき、プログラムがそのファイルシステム上のファイルを開いているとき、またはファイルシステムが共有されているときには、ファイルシステムは使用中とみなされます。

次の方法でファイルシステムをマウント解除が可能な状態にできます。

- 別のファイルシステム内のディレクトリにカレントディレクトリを変更する。
- システムからログアウトする。
- fuser コマンドを使用して、そのファイルシステムを使用中のすべてのプロセスを表示し、必要に応じて終了させる。詳細は、[40 ページの「ファイルシステムを使用中のすべてのプロセスを終了させる方法」](#)を参照してください。

ほかのユーザーが使用しているファイルシステムをマウント解除する必要があるときは、各ユーザーに通知します。

- ファイルシステムの共有を解除する。
 - 例:

```
# zfs set share.nfs=off tank/fs1
```

- レガシー unshare メソッドを使用します。詳細は、[unshare\(1M\)](#)を参照してください。

ファイルシステムをマウント解除したことを確認するには、mount コマンドからの出力を調べます。

```
$ mount | grep unmounted-file-system
```

▼ /etc/vfstab ファイルにエントリを追加する方法

レガシーのマウント動作が必要な ZFS ファイルシステムが存在しない場合は、ここに示す手順に従ってブート時に非 ZFS ファイルシステムをマウントします。ZFS ファイルシステムをマウントする方法の詳細は、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』を参照してください。

1. **管理者になります。**

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. **必要に応じて、マウントするファイルシステムのマウントポイントを作成します。**

```
# mkdir /mount-point
```

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にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. **/etc/vfstab ファイルを編集し、エントリを追加します。次のことを確認してください。**

- a. 各フィールドを空白 (空白文字またはタブ) で区切ります。
- b. フィールドで値を指定しない場合はダッシュ (-) を入力します。
- c. 変更を保存します。

注記 - ルート (/) ファイルシステムは、ブートプロセスの過程でカーネルによって読み取り専用としてマウントされます。そのため、remount オプション (および、remount と一緒に使用できるオプション) だけが /etc/vfstab ファイルのルート (/) エントリでは有効です。

例 2-7 /etc/vfstab ファイルにエントリを追加する

次の例は、ディスクスライス /dev/dsk/c0t3d0s7 をレガシー UFS ファイルシステムとして、マウントポイント /files1 にマウントする方法を示しています。「device to fsck」として raw キャラクタ型デバイス /dev/rdisk/c0t3d0s7 を指定します。「fsck pass」の値が 2 なので、ファイルシステムは順不同で検査されます。

```
#device          device          mount  FS      fsck  mount  mount
#to mount        to fsck         point  type   pass  at boot options
#
/dev/dsk/c0t3d0s7 /dev/rdisk/c0t3d0s7 /files1 ufs     2     yes    -
```

次の例は、システム pluto 上のディレクトリ /export/man を、NFS ファイルシステムとしてマウントポイント /usr/man にマウントする方法を示しています。ファイルシステムが NFS であるため、「device to fsck」や「fsck pass」は指定されません。この例では、「mount options」は ro (読み取り専用) と soft になっています。

```
#device          device          mount  FS      fsck  mount  mount
#to mount        to fsck         point  type   pass  at boot options
pluto:/export/man -             /usr/man nfs    -     yes    ro,soft
```

/etc/vfstab ファイルに、リモートシステムおよびリソースを追加した後に、次のサービスが起動していることを確認してください。

```
# svcs -a | grep nfs/client
disabled      May_14      svc:/network/nfs/client:default
# svcadm enable svc:/network/nfs/client:default
```

そうでない場合、リモートファイルシステムは、システムのリブート後にマウントされません。

▼ 1 つのファイルシステム (/etc/vfstab ファイル) をマウントする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. /etc/vfstab ファイル内に列挙されているファイルシステムをマウントします。

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

/mount-point は、/etc/vfstab ファイル内の「mount point」または「device to mount」フィールドのエントリを指定します。通常は、マウントポイントを指定するほうが簡単です。

例 2-8 1 つのファイルシステムをマウントする (/etc/vfstab ファイル)

次の例は、/etc/vfstab ファイル内に列挙されているローカル /legacy ファイルシステムをマウントする方法を示しています。

```
# mount /legacy
```

例 2-9 すべてのファイルシステムをマウントする (/etc/vfstab ファイル)

次に、ファイルシステムがすでにマウントされている状態で `mountall` コマンドを使用したとき表示されるメッセージの例を示します。

```
# mountall
mount: /tmp is already mounted or swap is busy
```

次の例は、`/etc/vfstab` ファイル内に列挙されているすべてのローカルシステムをマウントする方法を示しています。

```
# mountall -l
```

次の例は、使用可能なすべての ZFS ファイルシステムをマウントする方法を示します。

```
# zfs mount -a
```

次の例は、`/etc/vfstab` ファイル内に列挙されているすべてのリモートファイルシステムをマウントする方法を示しています。

```
# mountall -r
```

▼ NFS ファイルシステムのマウント方法 (mount コマンド)

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

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にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. リソース (ファイルまたはディレクトリ) がサーバーから使用可能かどうかを確認します。

NFS ファイルシステムをマウントするには、`share` コマンドを使用し、サーバー上のリソースを使用可能にしておかなければなりません。リソースを共有する方法の詳細は、『Oracle Solaris 11.2 のネットワークファイルシステム管理に関する説明』を参照してください。

4. NFS ファイルシステムをマウントします。

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

例 2-10 NFS ファイルシステムをマウントする (mount コマンド)

次の例は、サーバー pluto の /export/packages ディレクトリを /mnt にマウントする方法を示しています。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする方法 (mount コマンド)

次の手順で、PCFS (DOS) ファイルシステムをハードディスクからマウントします。

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 必要に応じて、マウントするファイルシステムのマウントポイントを作成します。

```
# mkdir /mount-point
```

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にマウントポイントを作成する必要があります。「マウントポイント」とは、マウントされるファイルシステムが接続されるディレクトリのことです。

3. PCFS ファイルシステムをマウントします。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
```

`-o rw | ro` PCFS ファイルシステムを読み取り / 書き込み (rw) または読み取り専用 (ro) にマウントできることを指定します。このオプションを指定しない場合のデフォルトは rw です。

`/dev/dsk/device-name` ディスク全体のデバイス名を指定します (/dev/dsk/c0t0d0p0 など)。

`logical-drive` DOS の論理ドライブ名 (c から z) またはドライブ番号 (1 から 24) を指定します。ドライブ c はドライブ 1 に相当し、ディスク上のプライマリ

DOS スライスを表します。ほかのすべてのドライブ名やドライブ番号は、拡張 DOS スライス内の DOS 論理ドライブを表します。

`/mount-point` ファイルシステムをマウントするディレクトリを指定します。

「device-name」と「logical-drive」は、コロンで区切る必要があります。

例 2-11 x86: ハードディスクから PCFS (DOS) ファイルシステムをマウントする (mount コマンド)

次の例は、プライマリ DOS スライス内の論理ドライブを `/pcfs/c` ディレクトリにマウントする方法を示しています。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

次の例では、`/mnt` ディレクトリの拡張 DOS スライスに含まれる最初の論理ドライブを読み取り専用でマウントする方法を示します。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

▼ ファイルシステムを使用中のすべてのプロセスを終了させる方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 停止対象のプロセスを確認するために、ファイルシステムにアクセスしているすべてのプロセスを一覧表示します。

```
# fuser -c [ -u ] /mount-point
```

`-c` ファイルシステムのマウントポイントとなっているファイルと、マウントされているファイルシステム内のファイルがすべて表示されます。

`-u` プロセス ID ごとにユーザーのログイン名が表示されます。

`/mount-point` プロセスを終了させるファイルシステムの名前を指定します。

3. ファイルシステムを使用しているすべてのプロセスを終了させます。

```
# fuser -c -k /mount-point
```

ファイルシステムを使用している各プロセスに SIGKILL が送信されます。

注記 - ユーザーのプロセスを終了させるときには、必ず事前に警告してください。

4. ファイルシステムを使用しているプロセスがないことを確認します。

```
# fuser -c /mount-point
```

例 2-12 ファイルシステムを使用中のすべてのプロセスを終了させる

次の例は、/export/home ファイルシステムを使用中のプロセス 4006c を終了させる方法を示しています。

```
# fuser -c /export/home
/export/home: 4006c
# fuser -c -k /export/home
/export/home: 4006c
# fuser -c /export/home
/export/home:
```

▼ 1 つのファイルシステムをマウント解除する方法

次の手順を使って、ファイルシステムをマウント解除します。

1. [35 ページの「Oracle Solaris ファイルシステムをマウント解除するための前提条件」](#)の前提条件を満たしているかどうかを確認します。
2. ファイルシステムをマウント解除します。

```
# umount /mount-point
```

/mount-point は、マウント解除するファイルシステムの名前を示します。

この名前は、次のいずれかを指定できます。

- ファイルシステムがマウントされているディレクトリ名
- ファイルシステムのデバイス名パス
- NFS ファイルシステムのリソース
- LOFS ファイルシステムのループバックディレクトリ

例 2-13 ファイルシステムをアンマウントする

次の例は、レガシー UFS ファイルシステムをマウント解除する方法を示しています。

```
# umount /legacy
```

次の例は、UFS の /legacy ファイルシステムを強制的にアンマウントする方法を示しています。

```
# umount -f /legacy
```

次の例は、すべての ZFS ファイルシステムをマウント解除する方法を示しています。

```
# zfs umount -a
```

使用中のファイルシステムを除く、すべてのファイルシステムがマウント解除されます。

◆◆◆ 第 3 章

追加スワップ空間の構成

この章では、Oracle Solaris OS のインストール後に、ZFS ルートファイルシステムで追加スワップ空間を構成するためのガイドラインや手順について説明します。

この章の内容は次のとおりです。

- [43 ページの「スワップ空間について」](#)
- [47 ページの「スワップ空間の追加が必要かどうかを調べる方法」](#)
- [48 ページの「スワップ空間の割り当て方法」](#)
- [48 ページの「スワップ空間の計画」](#)
- [50 ページの「スワップリソースのモニタリング」](#)
- [51 ページの「Oracle Solaris ZFS ルート環境でのスワップ空間の追加または変更」](#)

スワップ空間について

次を決定するには、Oracle Solaris のスワップメカニズムの機能を理解してください。

- スワップ空間の要件
- スワップ空間と TMPFS ファイルシステムとの関係
- スワップ空間に関連するエラーメッセージから回復する方法

スワップ空間と仮想メモリー

Oracle Solaris OS ソフトウェアやアプリケーションソフトウェアは、一部のディスク領域を、ファイルシステムとしてではなく一時ストレージとして使用できます。ディスク上に予約される領域は、スワップ空間と呼ばれます。スワップ空間は、現在のプロセスを処理するだけの十分な物理メモリーがシステムにない場合に、仮想メモリー記憶域として使用されます。ZFS ルートファイルシステムの場合、スワップ用に予約されるディスク領域は、ZFS ボリュームになります。

仮想メモリーシステムは、ディスク上のファイルの物理コピーをメモリー内の仮想アドレスに対応付けます。これらのマッピングに関するデータが入った物理メモリーページは、ファイルシステム内の通常ファイルまたはスワップ空間から読み直されます。メモリーをバックアップしているディスク空間に割り当てられる ID はわからないため、スワップ空間から読み直されたメモリーは匿名メモリーと呼ばれます。

Oracle Solaris OS には、*仮想スワップ空間*という概念が導入されています。これは、匿名メモリーページとこれらのページを実際にバックアップする物理ストレージ (またはディスク上にとられたバックアップ用のスワップ空間) の間に位置する層です。システムの仮想スワップ空間は、すべての物理 (ディスク上にとられたバックアップ用のスワップ空間) スワップ空間と現在使用可能な物理メモリーの一部の合計に等しくなります。

仮想スワップ空間の長所は次のとおりです。

- 仮想スワップ空間が物理 (ディスク) 記憶域に対応していなくてもかまわないので、大きな物理スワップ空間を確保する必要がなくなります。
- SWAPFS という疑似ファイルシステムが、匿名メモリーページのアドレスを提供します。SWAPFS はメモリーページの割り当てを制御するので、ページに対する処理を柔軟に決定できます。たとえば、ディスク上にとられたバックアップ用のスワップ記憶域のページ要件を変更できます。

スワップ空間と TMPFS ファイルシステム

Oracle Solaris 環境では、TMPFS ファイルシステムは `/etc/vfstab` ファイル内のエントリによって自動的に稼働されます。TMPFS ファイルシステムは、ファイルとそれに関連付けられた情報をディスクではなくメモリー (`/tmp` ディレクトリ) に格納するので、これらのファイルへのアクセスが高速になります。この機能によって、コンパイラや DBMS 製品のように `/tmp` の使用量の大きいアプリケーションの場合は、パフォーマンスが大幅に改善されます。

TMPFS ファイルシステムは、システムのスワップリソースから `/tmp` ディレクトリ内の領域を割り当てます。つまり、`/tmp` ディレクトリ内の領域を使い果たすと、スワップ空間も使い果たしたことになります。したがって、`/tmp` ディレクトリの使用量が大きいアプリケーションの場合は、スワップ空間の使用状況をモニターしなければ、システムがスワップ空間を使い果たす可能性があります。

スワップリソースが限られているときに TMPFS を使用する場合は、次の方法を使用してください。

- サイズオプション (-o size) を指定して TMPFS ファイルシステムをマウントし、TMPFS が使用できるスワップリソースを制御します。
- より大きな別のディレクトリをポイントするように、コンパイラの TMPDIR 環境変数の設定を変更します。

コンパイラの TMPDIR 変数を使用すると、コンパイラが /tmp ディレクトリを使用するかどうかだけが制御されます。この変数は、ほかのプログラムによる /tmp ディレクトリの使用には影響しません。

スワップ空間とダンプデバイスの構成

通常、ダンプデバイスとは、システムクラッシュダンプ情報を格納するために予約されているディスク領域のことです。システムのインストール時に、ZFS スワップボリュームおよびダンプボリュームが自動的に作成されます。システムのダンプボリュームは、dumpadm コマンドを使って変更できます。詳細は、『Oracle Solaris 11.2 でのシステム管理のトラブルシューティング』を参照してください。

ZFS ルート環境では、スワップとダンプは別々の ZFS ボリュームとして構成されます。このモデルの利点は次のとおりです。

- スワップ領域とダンプ領域を含めるためにディスクをパーティションに分割する必要がない。
- スワップおよびダンプデバイスが背後の ZFS 入出力パイプラインアーキテクチャーを利用できる。
- スワップおよびダンプデバイス上で圧縮などの特性を設定できる。
- スワップおよびダンプデバイスの両方のサイズをリセットできる。例:

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME          PROPERTY  VALUE      SOURCE
rpool/dump    volsize   2G         -
```

大規模ダンプデバイスを割り当てる処理には時間がかかることに注意してください。

- アクティブなシステムに 2 番目のスワップボリュームを追加して、全体のスワップサイズを増やすことを考慮できる。

ZFS スワップおよびダンプデバイスの使用方法の詳細については、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の「ZFS スワップデバイスおよびダンプデバイスを管理する」を参照してください。

スワップ空間と動的再構成

動的再構成時に CPU やシステムボードで障害が発生した場合に対応できるよう、十分なスワップ空間を確保することをお勧めします。スワップ空間が不足していると、CPU やシステムボードで障害が発生した際に、ホストまたはドメインはより少ないメモリーでリポートしなければなりません。

この追加スワップ空間を使用できない場合、メモリー不足のためアプリケーションの起動に失敗する可能性があります。この問題が発生した場合は、ユーザーが介入して、スワップ空間を追加するか、起動に失敗したアプリケーションのメモリー使用の構成を変更する必要があります。

リポート時のメモリー破損に備えて十分な追加スワップ空間が確保されていた場合、メモリーを大量に消費するアプリケーションのすべてが、通常どおりに起動します。したがって、スワップの多発によりシステムの動作は多少遅くなるにしても、ユーザーは通常どおりにシステムを利用できます。

詳細は、使用するハードウェアの動的構成ガイドを参照してください。

SAN 環境でのスワップ空間の構成

SAN 環境などで、ネットワークで接続されたディスクにスワップ空間を構成するかどうかを判断するには、次の点を確認してください。

- ローカルに接続されたディスクのスワップ空間の問題を診断する方が、ネットワークで接続されたディスクのスワップ空間の問題を診断するより簡単です。
- SAN 経由のスワップ空間のパフォーマンスは、ローカルに接続されたディスクに構成されているスワップ空間と同等になります。
- システムのパフォーマンスに問題がある場合は、パフォーマンスデータを分析したあとでシステムにメモリーを追加する方が、ローカルに接続されたディスクにスワップを移動するよりも、SAN 経由のスワップのパフォーマンス問題をより適切に解決できる可能性があります。

スワップ空間の追加が必要かどうかを調べる方法

スワップ空間の追加が必要かどうかを調べるには、`swap -l` コマンドを使用します。

たとえば、次の `swap -l` コマンドの出力は、このシステムのスワップ空間がほぼ完全に使い果たされたか、割り当て率が 100% に達していることを示しています。

```
$ swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 182,2    16  67108848  92
```

システムのスワップ空間の割り当て率が 100% に達すると、アプリケーションのメモリーページが一時的にロックしてしまいます。アプリケーションエラーが発生しない場合でも、システムパフォーマンスが低下する可能性があります。

スワップ関連のエラーメッセージ

次のメッセージは、アプリケーションに匿名メモリーを追加する必要がある場合に表示されます。ただし、バックアップ用のスワップ空間はもう余っていません。

```
application is out of memory

malloc error 0

messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

TMPFS 関連のエラーメッセージ

次のメッセージは、ファイルに書き込むときにページを割り当てることができない場合に表示されます。この問題は、TMPFS が許容限度を超えて書き込もうとしたときや、現在実行されているプログラムが大量のメモリーを使用している場合に発生することがあります。

```
directory: File system full, swap space limit exceeded
```

次のメッセージは、TMPFS が新しいファイルやディレクトリの作成中に物理メモリーを使い果たしたことを意味します。

```
directory: File system full, memory allocation failed
```

TMPFS 関連のエラーメッセージから回復する方法については、[tmpfs\(7FS\)](#) のマニュアルページを参照してください。

スワップ空間の割り当て方法

最初に、スワップ空間は Oracle Solaris インストールプロセスの一部として割り当てられますが、スワップ空間サイズはシステムメモリーによって異なります。

スワップ空間の割り当て方法については、[48 ページの「スワップ空間の計画」](#)を参照してください。

スワップ空間と `/etc/vfstab` ファイル

システムのインストールが完了すると、`/etc/vfstab` ファイルにスワップ領域とスワップファイルが一覧表示されます。スワップスライスとスワップファイルは、システムのブート時に `/sbin/swapadd` スクリプトによって有効になります。

`/etc/vfstab` ファイル内のスワップデバイスエントリには、次の情報が入っています。

- ZFS ルートファイルシステムを持つシステム上のスワップボリュームパス名の完全パス名
- スワップスライスまたはスワップファイルのファイルシステムタイプ

スワップファイルが入っているファイルシステムは、スワップファイルが有効になる前にマウントしておかなければなりません。このため、`/etc/vfstab` ファイル内で、ファイルシステムをマウントするエントリが、スワップファイルを有効にするエントリより前に入っていることを確認してください。

スワップ空間の計画

スワップ空間のサイズを決定する上でもっとも重要な要素は、システムのソフトウェアアプリケーションの要件です。たとえば、コンピュータ支援設計シミュレータ、データベース管理製品、トランザクションモニター、地質分析システムなどの大型アプリケーションは、200 - 1000M バイトのスワップ空間を消費することがあります。

アプリケーションのスワップ空間の要件については、アプリケーションベンダーに問い合わせてください。

アプリケーションベンダーに問い合わせてもスワップ空間の要件を判断できない場合は、次のシステムタイプ別のガイドラインに従ってスワップ空間を割り当ててください。

注記 - クラッシュダンプの内容は圧縮されるため、ダンプデバイスを物理メモリと同じサイズにする必要はありません。デフォルトでは、ダンプ内容の値はカーネルページに設定されます。ただし、すべてのメモリーページをダンプするようにダンプ内容の値が設定されている場合は、ダンプサイズを物理メモリーの半分以上に増やすことを検討してください。

表 3-1 ZFS ファイルシステムのスワップボリュームとダンプボリュームのサイズ

システムタイプ	スワップボリュームのサイズ	ダンプボリュームのサイズ
約 4G バイトの物理メモリーを備えたシステム	1G バイト	2G バイト
約 8G バイトの物理メモリーを備えた中型のサーバー	2G バイト	4G バイト
約 16 - 128G バイトの物理メモリーを備えたハイエンドサーバー	4G バイト	8 - 64G バイト
128G バイトを超える物理メモリーを備えたハイエンドサーバー	物理メモリーサイズの 1/4	物理メモリーサイズの 1/2

注記 - アクティブな ZFS ファイルシステムを多数備えたビジー状態のシステムでは、物理メモリーの 1/2 から 3/4 のサイズがダンプデバイスのサイズとして使用される可能性があります。

ZFS ルートファイルシステムを含むシステムで、サイズが小さすぎてシステムクラッシュを保持できないようなダンプデバイスを `dumpadm -d` コマンドで指定しようとすると、次のようなメッセージが表示されます。

```
dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to hold a
system dump dump size 43467329536 bytes, device size 42949672960 bytes
```

ZFS ベースのシステムでのスワップ空間の割り当て

ZFS ルートファイルシステムの初期インストール時に、スワップ領域が ZFS ルートプール内の ZFS ボリュームに自動的に作成されます。

- スワップデバイスが固定サイズのスライスに事前に割り当てられることはないため、あとでスワップサイズを変更することは非常に簡単です。
- アプリケーションのスワップ要件を評価してから、初期インストール時またはインストール後に必要に応じて、デフォルトのスワップサイズを使用することも、スワップボリュームのサイズを調整することもできます。

- ZFS 環境では、ファイルシステムはプールの領域を消費するため、`/var/crash` ディレクトリは、保存するクラッシュダンプの数に応じて必要な容量を消費します。

スワップボリュームを作成する場合、次の問題を考慮してください。

- スワップ領域とダンプデバイスには別個の ZFS ボリュームを使用する必要があります。
- 現時点では、ZFS ファイルシステムでスワップファイルを使用することはできません。
- スワップデバイスとダンプデバイスを非ルートプールに作成する場合は、RAID-Z プールにスワップボリュームとダンプボリュームを作成しないでください。プールにスワップボリュームとダンプボリュームが含まれている場合、これは 1 ディスクプールまたはミラー化プールでなければなりません。そうでない場合は、次のようなメッセージが表示されます。

```
/dev/zvol/dsk/rzpool/swap: Operation not supported
```

スワップリソースのモニタリング

`/usr/sbin/swap` コマンドを使用してスワップ領域を管理します。2 つのオプション `-l` と `-s` は、スワップリソースに関する情報を表示します。

`swap -l` コマンドを使用すると、システムのスワップ領域を確認できます。有効になっているスワップデバイスやファイルは、`swapfile` カラムの下に表示されます。例:

```
# swap -l
swapfile          dev swaplo blocks  free
/dev/dsk/c0t0d0s1 136,1          16 1638608 1600528
```

ZFS ルートファイルシステムを備えるシステムでは、`swap -l` コマンドにより、ZFS ボリュームパス名が出力されることを除いて、類似の出力が得られます。例:

```
# swap -l
swapfile          dev swaplo blocks  free
/dev/zvol/dsk/rpool/swap 256,1          16 1058800 1058800
```

`swap -s` コマンドを使用すると、スワップリソースをモニターできます。

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

`used` 値と `available` 値の合計は、システム上の合計スワップ空間に等しくなります。これには、物理メモリーの一部とスワップデバイス (またはファイル) が含まれます。

使用可能なスワップ空間と使用済みスワップ空間の容量 (`swap -s` の出力内) を使用して、時間経過に伴うスワップ空間の使用状況をモニターできます。システムパフォーマンスが適正であれば、`swap -s` を使用するとどの程度のスワップ空間が使用可能であるかがわかります。システムパフォーマンスが低下したときは、使用可能なスワップ空間の容量を検査して減少していないかどうかを調べてください。これによって、システムに対するどのような変更が原因でスワップ空間の使用量が増大したかを識別できます。

このコマンドを使用するときには、カーネルとユーザープロセスが物理メモリーをロックして解除するたびに、スワップに使用できる物理メモリーの容量が動的に変化するので注意してください。

注記 - `swap -l` コマンドでは、スワップ空間が 512 バイトのブロック数として表示されます。`swap -s` コマンドでは、スワップ空間が 1024 バイトのブロック数として表示されます。`swap -l` で表示されたブロック数を合計して K バイトに変換すると、結果は「used + available」(`swap -s` の出力) より小さくなります。これは、`swap -l` では、スワップ空間の計算に物理メモリーが算入されないからです。

次の表に、`swap -s` コマンドの出力とその説明を示します。

表 3-2 `swap -s` コマンドの出力

キーワード	説明
bytes allocated	現在バックアップ用の記憶域 (ディスク上にとられたバックアップ用のスワップ空間) として使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数。
reserved	現在は割り当てられていないが、あとから使用できるようにメモリーによって回収されるスワップ空間の合計容量を表す 1024 バイトのブロック数。
used	割り当て済みまたは予約済みのスワップ空間の合計容量を表す 1024 バイトのブロック数。
available	あとから予約や割り当てに使用可能なスワップ空間の合計容量を表す 1024 バイトのブロック数。

Oracle Solaris ZFS ルート環境でのスワップ空間の追加または変更

次のセクションでは、ZFS ルート環境でスワップ空間を追加または変更する方法を説明します。システムまたはアプリケーションで追加のスワップ空間が必要かどうかを確認する方法については、前述のセクションを参照してください。

ZFS ルート環境でのスワップおよびダンプボリュームの変更の詳細は、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の「ZFS スワップデバイスおよびダンプデバイスを管理する」を参照してください。

▼ Oracle Solaris ZFS ルート環境でスワップ空間を追加する方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 現在のスワップボリュームを確認します。

使用中のスワップボリュームは削除できません。現在のスワップボリュームが使用中かどうかは、blocks 列に示されるブロックと free 列に示されるブロックを比較して確認できます。2 つの列のブロックが等しい場合、スワップ領域は使用中ではありません。例:

```
# swap -l
swapfile          dev swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 256,1      16 1058800 1058800
```

3. 次のいずれかを選択して、スワップボリュームのサイズを変更します。

- a. 現在のスワップ領域が使用中でなければ、現在のスワップボリュームのサイズを変更できますが、増加したスワップ空間を確認するには、システムをリポートする必要があります。

例:

```
# zfs get volsize rpool/swap
NAME      PROPERTY  VALUE  SOURCE
rpool/swap volsize   517M   -
# zfs set volsize=2g rpool/swap
# zfs get volsize rpool/swap
NAME      PROPERTY  VALUE  SOURCE
rpool/swap volsize   2G     -
# init 6
```

- b. システムをリポートできない場合、別のスワップボリュームを追加して、総スワップ空間を増やします。

例:

```
# zfs create -V 2G rpool/swap2
```

2 番目のスワップボリュームをアクティブにします。

```
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile          dev swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 256,1    16 1058800 1058800
/dev/zvol/dsk/rpool/swap2 256,3    16 4194288 4194288
```

4. 必要に応じて、`/etc/vfstab` ファイルに 2 つ目のスワップボリュームのエントリを追加します。

例:

```
/dev/zvol/dsk/rpool/swap2 - - swap - no -
```


◆◆◆ 第 4 章

ファイルとファイルシステムのコピー

この章では、各種のバックアップコマンドを使用して、ファイルとファイルシステムをディスク、テープ、フロッピーディスクにコピーする方法について説明します。

この章で説明する手順は次のとおりです。

- 57 ページの「ファイルシステム間でディレクトリをコピーする方法 (cpio)」
- 59 ページの「ファイルをテープにコピーする方法 (tar)」
- 60 ページの「テープ上のファイルのリストを表示する方法 (tar)」
- 61 ページの「テープからファイルを取り出す方法 (tar)」
- 62 ページの「pax コマンドを使用してファイルをテープにコピーする」
- 63 ページの「ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)」
- 64 ページの「テープ上のファイルのリストを表示する方法 (cpio)」
- 65 ページの「テープからすべてのファイルを取り出す方法 (cpio)」
- 66 ページの「テープから特定のファイルを取り出す方法 (cpio)」
- 67 ページの「ファイルをリモートテープデバイスにコピーする方法 (tar と dd)」
- 68 ページの「ファイルをリモートテープデバイスから抽出する方法」

ファイルシステムをコピーするためのコマンド

個々のファイル、ファイルシステムの一部、またはファイルシステム全体をコピーまたは移動する場合は、この章で説明する手順を使用できます。

次の表に、Oracle Solaris リリースで使用可能な各種のバックアップおよび復元コマンドを示します。エンタープライズ環境では、エンタープライズレベルのバックアップ製品の使用を検討してください。エンタープライズレベルのバックアップ製品に関する情報は、Oracle Technical Network から入手できます。

表 4-1 バックアップコマンドのサマリー

コマンド名	ファイルシステム境界の認識	複数ボリュームバックアップのサポート	物理コピー / 論理コピー
volcopy	はい	はい	物理
tar	いいえ	いいえ	論理
cpio	いいえ	はい	論理
pax	はい	はい	論理
dd	はい	いいえ	物理
zfs send および zfs receive	はい	なし	論理
zfs snapshot	はい	なし	論理

次の表では、これらのコマンドの一部の長所と短所を説明します。

表 4-2 tar、pax、cpio コマンドの長所と短所

コマンド	機能	長所	短所
tar	ファイルやディレクトリサブツリーを 1 本のテープにコピーする場合に使用します。	<ul style="list-style-type: none"> ■ ほとんどの UNIX オペレーティングシステムで利用できます ■ パブリックドメインバージョンもすぐに利用できます 	<ul style="list-style-type: none"> ■ ファイルシステムの境界を認識しません ■ 完全パス名は最大 255 文字です ■ 複数のテープボリュームを作成する場合は使用できません
pax	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用します。または、POSIX 準拠システムとの間でファイルをコピーする場合に使用します。	<ul style="list-style-type: none"> ■ POSIX 準拠システムに対する、tar コマンドや cpio コマンドよりも高い互換性 ■ 複数のベンダーサポート 	tar コマンドの短所を参照してください。ただし、pax は、複数のテープボリュームを作成できます。
cpio	複数のテープボリュームを必要とするファイル、特殊ファイル、またはファイルシステムをコピーする場合に使用します。または、Oracle Solaris の最新リリースを実行しているシステムから古い Solaris リリースを実行しているシステムへファイルをコピーする場合に使用します。	<ul style="list-style-type: none"> ■ tar コマンドよりも効率的に、データをテープに書き込みます ■ 復元時、テープ中の不良箇所をスキップします ■ 異なるシステムタイプ間の互換性のために、異なるヘッダー形式 (tar、ustar、crc、odc、bar など) でファイルを書き込むオプションを提供します 	コマンド構文が tar コマンドや pax コマンドよりも難解です。

コマンド	機能	長所	短所
		■ 複数のテープボリュームを作成します	

次のセクションでは、これらのコマンドの使用例を紹介します。

cpio を使用してファイルシステム間でディレクトリをコピーする

cpio (コピーインとコピーアウト) コマンドを使用して、個々のファイル、ファイルグループ、またはファイルシステム全体をコピーできます。このセクションでは、cpio コマンドを使ってファイルシステム全体をコピーする方法について説明します。

cpio コマンドは、ファイルのリストを 1 つの大型出力ファイルにコピーするアーカイブプログラムです。また、復元しやすいように、個々のファイルの間にヘッダーを挿入します。cpio コマンドを使用すると、ファイルシステム全体を別のスライス、別のシステム、またはテープや USB フロッピーディスクなどのメディアデバイスにコピーできます。

cpio コマンドは、メディアの終わりを認識し、別のボリュームを挿入するように促すプロンプトを表示するので、複数のテープや USB フロッピーディスクが必要なアーカイブを作成するにはもっとも効率のよいコマンドです。

cpio コマンドの使用時には、しばしば `ls` や `find` のコマンドを使用してコピーするファイルを選択し、その出力を cpio コマンドにパイプします。

▼ ファイルシステム間でディレクトリをコピーする方法 (cpio)

1. 管理者になります。
詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。
2. 目的のディレクトリに移動します。

```
# cd filesystem1
```
3. `find` コマンドと `cpio` コマンドを組み合わせ使用して、`filesystem1` から `filesystem2` へディレクトリツリーをコピーします。

```
# find . -print -depth | cpio -pdm filesystem2
```

.	現在の作業ディレクトリで処理を始めます。
-print	ファイル名を出力します。
-depth	ディレクトリ階層を下降し、すべてのファイル名を出力します。
-p	ファイルのリストを作成します。
-d	必要に応じてディレクトリを作成します。
-m	ディレクトリ上で正しい変更時間を設定します。

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

指定したディレクトリ名からファイルがコピーされます。シンボリックリンクは保持されます。

また、`-u` オプションも指定できます。このオプションは、無条件にコピーを実行します。`u` オプションを指定しない場合、古いファイルが、新しいファイルで置換されません。このオプションは、ディレクトリごとコピーするときに、コピーするファイルの一部がすでにターゲットのディレクトリ中に存在する場合に便利です。

4. コピー先ディレクトリの内容を表示して、コピーに成功したかどうかを確認します。

```
# cd filesystem2
# ls
```

5. ソースディレクトリが不要な場合は削除します。

```
# rm -rf filesystem1
```

例 4-1 ファイルシステム間でディレクトリをコピーする (cpio)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

ファイルとファイルシステムをテープにコピーする

`tar`、`pax`、および `cpio` コマンドを使用すると、ファイルとファイルシステムをテープにコピーできます。どのコマンドを選択するかは、コピーする目的に応じて異なります。3 つのコマンドはすべて

raw デバイスを使用するので、使用する前にテープ上でファイルシステムをフォーマットまたは作成する必要はありません。

使用するテープドライブとデバイス名は、各システムのハードウェアの構成によって異なります。テープデバイス名の詳細については、『[Oracle Solaris 11.2 でのデバイスの管理](#)』の「[使用するメディアの選択](#)」を参照してください。

tar を使用してファイルをテープにコピーする

tar コマンドを使用してファイルをテープにコピーする前に、次の情報について知っておかなければなりません。

- -tar コマンドに c オプションを指定してファイルをテープにコピーすると、テープに入っているすべての既存のファイルまたはテープの現存の位置以降にある既存のファイルすべてが破壊 (上書き) されます。
- ファイルをコピーするとき、ファイル名の一部にファイル名置換ワイルドカード文字 (? と *) を使用できます。たとえば、接尾辞 .doc が付いたすべてのドキュメントをコピーするには、ファイル名引数として *.doc と入力します。
- tar アーカイブからファイルを抽出するときには、ファイル名置換ワイルドカードは使用できません。

▼ ファイルをテープにコピーする方法 (tar)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ tar cvf /dev/rmt/n filenames
```

c アーカイブの作成を指定します。

v 各ファイルがアーカイブされるたびに、その名前を表示します。

f /dev/rmt/n アーカイブを指定したデバイスまたはファイルに書き込むように指定します。

filenames コピーするファイルとディレクトリを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

指定した名前のファイルがテープにコピーされ、テープ上の既存のファイルが上書きされます。

4. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。
5. コピーしたファイルがテープ上に存在することを確認します。

```
$ tar tvf /dev/rmt/n
```

tar テープ上のファイルを表示する方法については、[60 ページの「テープ上のファイルのリストを表示する方法 \(tar\)」](#)を参照してください。

例 4-2 ファイルをテープにコピーする (tar)

次の例では、3 つのファイルをテープドライブ 0 のテープにコピーします。

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/0
```

▼ テープ上のファイルのリストを表示する方法 (tar)

1. テープをテープドライブに挿入します。
2. テープの内容を表示します。

```
$ tar tvf /dev/rmt/n
```

t テープ上のファイルのリストが表示されます。

v t オプションと一緒に使用すると、テープ上のファイルに関する詳細情報が表示されます。

f /dev/rmt/n テープデバイスを示します。

例 4-3 テープ上のファイルのリストを表示する (tar)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ tar tvf /dev/rmt/0
drwxr-xr-x  0/0      0 Jul 14 13:50 2010 reports/
-r--r--r--  0/0    206663 Jul 14 13:50 2010 reports/reportC
-r--r--r--  0/0    206663 Jul 14 13:50 2010 reports/reportB
-r--r--r--  0/0    206663 Jul 14 13:50 2010 reports/reportA
```

▼ テープからファイルを取り出す方法 (tar)

1. ファイルを置きたいディレクトリへ移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルを取り出します。

```
$ tar xvf /dev/rmt/n [filenames]
```

x	指定したアーカイブファイルからのファイルの抽出を指定します。指定したドライブのテープに含まれるすべてのファイルが現在のディレクトリにコピーされます。
v	各ファイルを取り出すたびに、その名前を表示します。
f /dev/rmt/n	アーカイブを含むテープデバイスを示します。
filenames	取り出すファイルを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

詳細は、[tar\(1\)](#) を参照してください。

4. ファイルがコピーされたことを確認します。

```
$ ls -l
```

例 4-4 テープ上のファイルを取り出す (tar)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
```

```
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

注意事項 テープから抽出されるファイル名は、アーカイブに格納されているファイル名と同一でなければなりません。ファイルの名前やパス名が不明な場合は、まずテープ上のファイルのリストを表示します。テープ上のファイルをリスト表示する方法については、[60 ページの「テープ上のファイルのリストを表示する方法 \(tar\)」](#)を参照してください。

pax コマンドを使用してファイルをテープにコピーする

このセクションでは、pax コマンドでファイルをテープにコピーする方法について説明します。

▼ ファイルをテープにコピーする方法 (pax)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ pax -w -f /dev/rmt/n filenames
```

`-w` 書き込みモードを有効にします。

`-f /dev/rmt/n` テープドライブを識別します。

`filenames` コピーするファイルとディレクトリを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

詳細は、[pax\(1\)](#) を参照してください。

4. ファイルがテープにコピーされたことを確認します。

```
$ pax -f /dev/rmt/n
```

5. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 4-5 ファイルをテープにコピーする (pax)

次の例は、pax コマンドを使用して、現在のディレクトリ内のファイルをすべてコピーする方法を示します。

```
$ pax -w -f /dev/rmt/0 .
$ pax -f /dev/rmt/0
filea fileb filec
```

cpio コマンドを使用してファイルをテープにコピーする

このセクションでは、cpio コマンドでファイルをテープにコピーする方法について説明します。

▼ ディレクトリ内のすべてのファイルをテープにコピーする方法 (cpio)

1. コピーするファイルの存在するディレクトリに移動します。
2. 書き込み可能なテープをテープドライブに挿入します。
3. ファイルをテープにコピーします。

```
$ ls | cpio -oc > /dev/rmt/n
```

ls cpio コマンドにファイル名のリストを渡します。

cpio -oc cpio コマンドがコピーアウトモード (-o) で動作し、ASCII 文字形式 (-c) でヘッダー情報を書き込むように指定します。これらのオプションによりほかのベンダーのシステムとの互換性を保ちます。

> /dev/rmt/n 出力ファイルを指定します。

ディレクトリ内のすべてのファイルは、指定したドライブ内のテープにコピーされ、テープ上の既存のファイルが上書きされます。コピーされた合計ブロック数が表示されます。

4. ファイルがテープにコピーされたことを確認します。

```
$ cpio -civt < /dev/rmt/n
```

-c cpio コマンドがファイルを ASCII 文字形式で読み込むように指定します。

- i cpio コマンドがコピーインモードで動作することを指定します。この時点ではファイルをリストするだけです。
- v ls -l コマンドの出力と同様の形式で出力を表示します。
- t 指定したテープドライブ内のテープ上にあるファイルの内容リストを表示します。
- < /dev/rmt/n 既存の cpio アーカイブの入力ファイルを指定します。

5. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 4-6 ディレクトリ内のすべてのファイルをテープにコピーする (cpio)

次の例では、/export/home/kryten ディレクトリ内のすべてのファイルをテープドライブ 0 のテープにコピーする方法を示します。

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
1280 blocks
$ cpio -civt < /dev/rmt/0
-r--r--r--  1 kryten  staff    206663 Jul 14 13:52 2010, filea
-r--r--r--  1 kryten  staff    206663 Jul 14 13:52 2010, fileb
-r--r--r--  1 kryten  staff    206663 Jul 14 13:52 2010, filec
drwxr-xr-x  2 kryten  staff         0 Jul 14 13:52 2010, letters
drwxr-xr-x  2 kryten  staff         0 Jul 14 13:52 2010, reports
1280 blocks
```

▼ テープ上のファイルのリストを表示する方法 (cpio)

注記 - テープの内容リストを表示するには、cpio コマンドがアーカイブ全体を処理する必要があるため、かなりの時間がかかります。

1. テープをテープドライブに挿入します。
2. テープ上のファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/n
```

例 4-7 テープ上のファイルのリストを表示する (cpio)

次の例では、ドライブ 0 のテープに含まれるファイルのリストを表示します。

```
$ cpio -civt < /dev/rmt/0
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filea
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, fileb
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filec
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, letters
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, reports
1280 blocks
```

▼ テープからすべてのファイルを取り出す方法 (cpio)

相対パス名を使用してアーカイブを作成した場合、入力ファイルはそれを取り出すときに現在のディレクトリ内のディレクトリとして作成されます。ただし、絶対パス名を指定してアーカイブを作成した場合は、それと同じ絶対パス名を使用してシステム上でファイルが再作成されます。



注意 - 絶対パス名を使用すると、自分のシステム上にある元のファイルを上書きすることになるので危険です。

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからすべてのファイルを抽出します。

```
$ cpio -icvd < /dev/rmt/n

-i          標準入力からファイルを取り出します。

-c          cpio コマンドがファイルを ASCII 文字形式で読み込むように指定します。

-v          取り出されたファイルを ls コマンドの出力と同様の形式で表示します。

-d          必要に応じてディレクトリを作成します。

< /dev/rmt/n 出力ファイルを指定します。
```

4. ファイルがコピーされたことを確認します。

```
$ ls -l
```

例 4-8 テープからすべてのファイルを取り出す (cpio)

次の例では、ドライブ 0 のテープからすべてのファイルを取り出す方法を示します。

```
$ cd /var/tmp
cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

▼ テープから特定のファイルを取り出す方法 (cpio)

1. ファイルを置きたいディレクトリに移動します。
2. テープをテープドライブに挿入します。
3. テープからファイルのサブセットを取り出します。

```
$ cpio -icv "file" < /dev/rmt/n
```

-i	標準入力からファイルを取り出します。
-c	cpio コマンドを使用してヘッダーを ASCII 文字形式で読み込むように指定します。
-v	取り出されたファイルを ls コマンドの出力と同様の形式で表示します。
" <i>file</i> "	パターンに一致するすべてのファイルを現在のディレクトリにコピーするように指定します。複数のパターンを指定できますが、個々のパターンを二重引用符で囲まなければなりません。
< /dev/rmt/ <i>n</i>	入力ファイルを指定します。

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

4. ファイルがコピーされたことを確認します。

```
$ ls -l
```

例 4-9 指定したファイルをテープから取り出す (cpio)

次の例では、末尾に接尾辞 `chapter` が付いているすべてのファイルをドライブ 0 のテープから取り出す方法を示します。

```
$ cd /home/smith/Book
$ cpio -icv "**chapter" < /dev/rmt/0
Boot.chapter
```

```
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

ファイルをリモートテープデバイスにコピーする

このセクションでは、tar および dd コマンドでファイルをリモートテープデバイスにコピーする方法について説明します。

▼ ファイルをリモートテープデバイスにコピーする方法 (tar と dd)

1. テープドライブにアクセスできるように、リモートシステムで ssh を構成します。『[Oracle Solaris 11.2 での Secure Shell アクセスの管理](#)』の「[Secure Shell の構成 \(タスク\)](#)」を参照してください。
2. ファイルを置きたいディレクトリに移動します。
3. テープをテープドライブに挿入します。
4. ファイルをリモートテープドライブへコピーします。

```
$ tar cvf - filenames | ssh remote-host dd of=/dev/rmt/n obs=block-size
```

tar cf テープアーカイブを作成し、アーカイブに含まれるファイルをリスト表示し、テープデバイスを指定します。

v tar ファイルのエントリに関する追加情報を表示します。

- (ハイフン) 可変部としてテープデバイスの代わりに指定します。

filenames コピーするファイルを指定します。ファイルが複数の場合は、各ファイルをスペースで区切ります。

ssh / remote-host tar コマンドの出力をパイプを通してリモートシステムに渡します。

dd of= /dev/rmt/
n 出力デバイスを指定します。

`obs=block-size` ブロック係数を指定します。

5. テープをドライブから取り出します。ファイル名をテープのラベルに記入します。

例 4-10 ファイルをリモートテープドライブにコピーする (tar と dd)

```
# tar cvf - * | ssh mercury dd of=/dev/rmt/0 obs=126b
password:
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

▼ ファイルをリモートテープデバイスから抽出する方法

1. テープをテープドライブに挿入します。
2. 一時ディレクトリに移動します。

```
$ cd /var/tmp
```

3. リモートテープデバイスからファイルを抽出します。

```
$ ssh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

`ssh remote-host` dd コマンドを使用してテープデバイスからファイルを取り出すために起動する Secure Shell です。

`dd if=/dev/rmt/n` 入力デバイスを指定します。

`| tar xvBpf -` dd コマンドの出力を tar コマンドにパイプして、ファイルを復元します。

4. ファイルが抽出されたことを確認します。

```
$ ls -l
```

例 4-11 ファイルをリモートのテープドライブから抽出する

```
$ cd /var/tmp
$ ssh mercury dd if=/dev/rmt/0 | tar xvBpf -
password:
```

```
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```


索引

数字・記号

ルート (/) ファイルシステムまたはディレクトリ, 17
/export/home ディレクトリ, 17
/home (自動マウント), 22
/opt ディレクトリ, 17
/proc ディレクトリ, 14, 17
/tmp ディレクトリ, 13, 17
/var ディレクトリ, 17
4.3 Tahoe ファイルシステム, 10

あ

一時ファイルシステム (TMPFS)
概要, 13

か

仮想メモリーストレージ, 定義, 43
擬似ファイルシステム
(概要), 13
強制終了
ファイルシステムにアクセスしているすべてのプロセス (方法), 40
共有
ファイル, 20
共有ファイルシステム, 21
検索
ファイルシステムのタイプ, 23
コピー
cpio コマンドを使って個々のファイルを (概要), 57
cpio コマンドを使ってファイルシステム間の (概要), 57
cpio コマンドを使ってファイルのグループを (概要), 57

さ

自動マウント
/home, 22
調べる
ファイルシステムタイプ, 23
ストレージ (仮想メモリー), 定義, 43
スワップパーティション, 定義, 43
スワップファイル
vfstab へ追加, 48
表示, 50

た

調査
マウント済みのファイルシステム, 32
追加
エントリ/etc/vfstab ファイル (方法), 36
スワップを vfstab へ, 48
停止
ファイルシステムに対するすべてのプロセス (方法), 40
ディスクベースのファイルシステム, 10
ディレクトリ
/proc, 14
/tmp, 13
cpio コマンドを使ってファイルシステム間のコピー (概要), 57
テープ
tar コマンドによるファイルの取り出し (方法), 61
デフォルト
/tmp 用のファイルシステム (TMPFS), 13
SunOS ファイルシステム, 17
取り出し
tar コマンドによるテープからのファイルの (方法), 61

な

ネットワークベースのファイルシステム, 12

は

パーティション (スワップ), 定義, 43

表示

スワップ空間, 50

ファイル

/proc ディレクトリ内, 14

tar コマンドによるテープからの取り出し (方法), 61

共有, 20

ファイルシステム

/export/home, 17

/opt, 17

/proc, 17

/var, 17

4.3 Tahoe, 10

BSD Fat Fast, 10

CTFS, 14

DOS, 11

FDFS, 15

FIFOFS, 15

High Sierra, 11

ISO 9660, 11

MNTFS, 17

NAMEFS, 15

OBJFS, 15

PCFS, 11

PROCFS, (概要), 14

SHAREFS, 15

SPECFS, 15

SWAPFS, 15

TMPFS, 13

UFS, 10

UNIX, 10

ZFS, 10, 10

アクセスしているすべてのプロセスの停止 (方法), 40

擬似, (概要), 13

共有, 20

タイプ, 10

タイプの検索, 23

ディスクベース, 10

デフォルト SunOS, 17

ネットワークベース, 12

プロセス, (概要), 14

マウントテーブル, 18

利用可能にする (概要), 32

ファイルシステムのタイプ, 10

プロセスファイルシステム (PROCFS), 14

ま

マウント

/etc/vfstab によるファイルシステムの, 37

NFS ファイルシステム, 37

UFS ファイルシステム, 36

ファイルシステムの自動マウント, 21

マウントテーブル, 18

マウントポイント定義, 18

メディアの終わりの検出

cpio コマンド, 57, 57

メモリーストレージ (仮想), 定義, 43

A

AutoFS, 21

B

BSD Fat Fast ファイルシステム, 10

C

cpio コマンド

(概要), 57

テープからすべてのファイルを抽出 (方法), 65

テープ上のファイルのリスト (方法), 64

ファイルシステム間でのディレクトリのコピー (方法), 57

CTFS ファイルシステム, 14

D

DOS

ファイルシステム, 11

DVD-ROM, 12

F

FDFS ファイルシステム, 15
FIFOFS ファイルシステム, 15
fsstat コマンド
 説明, 24
fsstat コマンド (例), 24

H

High Sierra ファイルシステム, 11
HSFS 参照 High Sierra ファイルシステム

I

ISO 9660 ファイルシステム, 11

M

MNTFS ファイルシステム, 17
mnttab ファイル, 18

N

NAMEFS ファイルシステム, 15
NFS
 vfstab エントリ, 37
 サーバーの説明, 20
 説明, 20

O

OBJFS ファイルシステム, 15

P

PCFS ファイルシステム, 11
PROCFS ファイルシステム
 (概要), 14

R

Rock Ridge 拡張 (HSFS ファイルシステム), 11

S

share コマンド, 20
shareall コマンド, 20
SHAREFS ファイルシステム, 15
SPECFS ファイルシステム, 15
SunOS のデフォルトファイルシステム, 17
swapadd コマンド, 48
SWAPFS ファイルシステム, 15

T

tar コマンド
 dd コマンドによるファイルのリモートテープへのコ
 ピー (方法), 67
 dd コマンドによるリモートテープからのファイルの取
 り出し (方法), 68
 (概要), 59
 テープからのファイルの取り出し (方法), 61
 テープ上のファイルのリスト (方法), 60
TMPFS ファイルシステム
 概要, 13

U

UDF ファイルシステム, 11
UFS ファイルシステム, 10
 /etc/vfstab によるマウント, 37
 マウント, 36
UNIX ファイルシステム, 10

V

vfstab ファイル, 48
 LOFS のエントリ, 31
 エントリの追加 (方法), 36
 スワップの追加, 48

Z

ZFS ファイルシステム, 10, 10

