



Solaris 移行ガイド

Sun Microsystems, Inc.
901 San Antonio Blvd.
Palo Alto, CA 94303
U.S.A.650-960-1300

Part No: 805-5828-10
1998 年 11 月

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。日本サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

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。

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービイ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イスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、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, SunSoft, SunDocs, SunExpress, OpenWindows は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

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、日本サン・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。

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1998 All Rights Reserved.)

ATOK は、株式会社ジャストシステムの登録商標です。

ATOK7 は株式会社ジャストシステムの著作物であり、ATOK7 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

ATOK8 は株式会社ジャストシステムの著作物であり、ATOK8 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

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、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DiComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(Copyright (c) 1993 Interleaf, Inc.)

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、日本サン・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 Transition Guide

Part No: 805-3864-10

Revision A

© 1998 by Sun Microsystems, Inc.



目次

- はじめに xv
- パートI ユーザとシステム管理者のための移行情報
- 1. 概要 3
 - Solaris オペレーティング環境に移行するメリット 3
 - 移植性、スケーラビリティ、相互運用性、互換性 5
 - 大規模な組織にとってのメリット 6
 - SVR4 と Solaris オペレーティング環境の比較 6
 - Solaris オペレーティング環境で追加された機能 7
 - Solaris オペレーティング環境に含まれない SVR4 の機能 11
- 2. 主な変更について 13
 - ソフトウェアパッケージとクラスタ 14
 - パッケージの管理 14
 - パッチの管理 15
 - ディスクスライス (またはディスクパーティション) 16
 - シリンダグループ 17
 - デバイスの命名 17
 - ファイルシステム 18
 - ファイルシステムの位置と名前の変更 18
 - 疑似ファイルシステム 19

	追加されたファイルシステム	19
	除去されたファイルシステム	20
	カーネルの構成	20
	カーネルのレイアウト	21
	自動マウント	21
	メールの管理	23
	Admintool	24
	ネットワーク情報サービスプラス (NIS+)	25
	印刷サブシステム	25
	印刷ツール	26
	コマンドの変更	26
	サービスアクセス機能	27
	ボリュームマネージャ	28
3.	SunOS 4.x システムから Solaris 7 オペレーティング環境への変換	29
	インストールに関する変更点	29
	Solaris をインストールする前に行うこと	30
	ディスクパーティション情報の保存	31
	ファイルシステム情報の保存	32
	メタデバイス構成情報の保存	32
	バックアップ内容の決定	33
	ディスク空間の所要量の決定	35
	ネットワークのインストール順序の決定	35
	インストール前のファイルとファイルシステムのバックアップ	35
	Solaris ソフトウェアのインストール	36
	保存オプション	36
	インストール後のファイルとファイルシステムの復元	36
	SunOS 4.x ファイルシステムとユーザファイルの復元	36
	SunOS 4.x システム構成ファイルの復元	37

4.	互換パッケージの使用方法	41
	アプリケーションを移植する理由	41
	SunOS/BSD ソース互換パッケージ	42
	バイナリ互換パッケージ	42
	SunOS 4.x のアプリケーションを実行するためのバイナリ互換パッケージの使用方法	43
5.	セキュリティ	45
	Solaris 7 のセキュリティ機能	45
	/etc/passwd と /etc/shadow ファイル	46
	/etc/default ファイル	46
	制限付きシェル	47
	パスワード有効期限の変更	47
	アクセス制御リスト (ACL)	48
	自動セキュリティ拡張ツール (ASET)	49
	セキュリティオプション	49
	Kerberos 4.0 セキュリティ	49
	SunSHIELD パッケージ	50
	PAM	50
6.	ユーザ環境の管理	51
	デフォルトシェルの選択	51
	ユーザ環境のカスタマイズ	53
	Solaris ソフトウェアでの SunOS 4.x 作業環境の使用	54
	ウィンドウシステム	54
	ユーザとグループの管理	55
	ユーザとグループ管理の選択	55
	ユーザアカウントの追加	56
	メールの使用	56
	文書ツールの使用	57

- マニュアルページの構成の変更 58
 - man コマンド検索パスのカスタマイズ 59
 - whatis データベースと windex データベース 60
 - man コマンドの使用 61
- 7. デバイスの管理 63
 - デバイス命名規則 63
 - ディスクに関する規則 64
 - テープドライブに関する規則 65
 - ディスク情報の表示 65
 - df コマンド 66
 - du コマンド 66
 - dckinfo コマンド 66
 - devinfo コマンド 67
 - デバイスをシステムへ追加する 67
 - 動的再構成 68
 - ボリュームマネージャの使用 68
- 8. 起動とシャットダウン 71
 - ブート 71
 - boot コマンドの変更 72
 - PROM からのブート 73
 - ブート処理の相違 73
 - init コマンドの使用 74
 - init コマンドの変更 75
 - システム実行レベルの変更 75
 - シャットダウン 76
 - shutdown コマンドの変更 77
 - fasthalt コマンドと fastboot コマンドの使用 78
 - halt コマンドと reboot コマンドの使用 78

9.	ファイルシステムの管理	79
	ファイルシステムの変更	80
	疑似ファイルシステム	80
	ファイルシステムの追加	81
	デフォルトのファイルシステムとディレクトリ	81
	仮想ファイルシステムアーキテクチャ	83
	サポートされるファイルシステムの形式	84
	サポートされない SVR4 ファイルシステムの形式	86
	汎用ファイルシステムコマンド	86
	ディレクトリとファイルの変更	89
	/dev ディレクトリ	90
	/etc ディレクトリ	90
	/sbin ディレクトリ	95
	/usr ディレクトリ	95
	/var ディレクトリ	96
	/kernel ディレクトリ	97
	/opt ディレクトリ	98
	/sys ディレクトリ	98
	ファイルシステム管理コマンドの使用	98
	ファイルシステムのマウントと autofs	99
	ファイルシステムの監視	101
	ファイルシステムの共用	102
	新しいファイルシステムの作成	103
	ファイルシステムのチェック	104
	ファイルのバックアップと復元	104
	UFS ログ	109
10.	SunOS 4.x クライアントをサポートする Solaris 7 サーバの設定	111
	SunOS 4.x をサポートする機能を Solaris 7 サーバへ追加する	111

- discover4x の実行 113
- CD-ROM ドライブを install4x 用に設定する 113
- install4x の実行 115
- convert4x の実行 118
- 11. プリンタ、端末、モデムの管理 121**
 - 印刷 121
 - 印刷の変更点の要約 121
 - 印刷コマンドと互換パッケージ 122
 - プリンタコマンドの使用 122
 - SunOS release 5.7 プリンタ管理コマンドの使用 123
 - シリアルポート管理 125
 - 端末とモデムの管理 125
 - サービスアクセス機能 (SAF) 125
- 12. ネットワークサービスの管理 129**
 - TCP/IP の変更 129
 - TCP と SACK 130
 - NFS の変更 130
 - PPP 131
 - LDAP 131
 - IIIMP 131
 - UUCP 131
 - チェックポイントからの再起動 133
 - ユーザジョブの処理順序 133
 - Limits ファイル 134
 - Config ファイル 134
 - ログファイル 134
- 13. ネームサービスの使用方法 135**
 - ネームサービススイッチ 136

	NIS+	136
	DNS	136
	DNS と NIS+ の比較	136
	NIS と NIS+ の比較	137
	NIS+ アップグレードの計画	139
14.	Solaris 共通デスクトップ環境	141
	Solaris 共通デスクトップ環境について	141
	開発者、エンドユーザ、および CDE	142
	デスクトップの概要	142
	フロントパネル	143
	スタイル・マネージャ	144
	ファイル・マネージャ	145
	OpenWindows 環境から CDE への移行	145
	デスクトップサービス	146
	CDE でのウィンドウ、メニュー、ボタンおよびマウスの使用	146
	ワークスペースアプリケーションメニューへのアクセス	147
	スタイル・マネージャおよびワークスペースのカスタマイズ	147
	CDE での OpenWindows アプリケーションの実行	147
	アプリケーションの設定とプロパティ	147
	キーボードのデフォルト値の変更	148
	マウスのデフォルト値の変更	148
	パートII 開発者用移行情報	
15.	コンパイラ、リンカ、デバッガ	151
	コンパイラ	151
	リンカ	152
	リンクエディタオプションの相違	152
	共用ライブラリの作成	155
	実行可能ファイルの作成	155

- ライブラリ検索パスの指定 156
- 検索パスの規則 156
- バージョン番号 157
- 例 158
- デバッグ 160
 - dbx と dbxtool 160
 - adb と kadb 160
 - kadb マクロ 161
 - 動作中のカーネルのデバッグ 161
 - truss コマンド 162
- 16. ツールと資源 163**
 - ioctl() 要求 163
 - ptrace() 要求値 166
 - ライブラリ 168
 - 再編成ライブラリ 168
 - 共用ライブラリ 168
 - 資源の制限 168
 - make の使用 171
 - SCCS の使用 172
 - アプリケーション互換性の判断 172
 - アプリケーションパッケージ作成 173
 - パッケージ作成ユーティリティ 174
 - ツールキット 175
 - OLIT 175
 - XView 175
 - SunOS 4.x ツールの検索 176
- 17. ネットワークと国際化機能 181**
 - ネットワーク 181

	NIS と NIS+	181
	nsswitch.conf ファイル	182
	Network Interface Tap	182
	ソケット	182
	国際化	182
	文字サポート	183
	メッセージカタログ	183
	ロケールデータベース	184
	コマンド	184
	ライブラリ	184
18.	システムとデバイスの構成	187
	システム構成	187
	動的にロードされるカーネル	187
	カーネルの配置	188
	config コマンド	188
	/etc/system ファイル	189
	boot コマンド	189
	ブートの相違点の要約	190
	再構成ブート	191
	開発者に関するデバイスの命名規則	191
	/devices	192
	/dev	192
	デバイスドライバの命名規則	192
A.	コマンドリファレンス	195
	リファレンスの使い方	195
	例	196
	コマンドリファレンス	198
	コマンドリファレンス (A, B)	198

	コマンドリファレンス (C, D)	202
	コマンドリファレンス (E, F, G, H, I, J)	211
	コマンドリファレンス (K, L, M, N, O)	220
	コマンドリファレンス (P, Q, R)	228
	コマンドリファレンス (S, T)	236
	コマンドリファレンス (U, V, W, X, Y, Z)	243
B.	システムコールリファレンス	251
	リファレンスの使い方	251
	例	252
	システムコールのリファレンス (A, B, C, D, E, F, G)	253
	システムコールのリファレンス (I, K, L, M, O, P, Q)	261
	システムコールのリファレンス (R, S, T, U, V)	268
	システムコールのリファレンス (W)	275
C.	ライブラリルーチンリファレンス	281
	リファレンスの使い方	281
	例	282
	ライブラリルーチンのリファレンス (A, B)	283
	ライブラリルーチンのリファレンス (C, D)	287
	ライブラリルーチンのリファレンス (E, F)	296
	ライブラリルーチンのリファレンス (G, H)	303
	ライブラリルーチンのリファレンス (I, J, K)	309
	ライブラリルーチンのリファレンス (L)	314
	ライブラリルーチンのリファレンス (M, N)	319
	ライブラリルーチンのリファレンス (O, P, Q, R)	327
	ライブラリルーチンのリファレンス (S)	336
	ライブラリルーチンのリファレンス (T)	350
	ライブラリルーチンのリファレンス (U, V)	356
	ライブラリルーチンのリファレンス (W)	359

	ライブラリルーチンのリファレンス (X, Y)	366
D.	システムファイルリファレンス	371
	リファレンスの使い方	371
	システムファイル	371
E.	/ と /usr ファイルシステムの変更	377
	/ ファイルシステムの配置	377
	/usr ファイルシステムの配置	381
F.	基本的な変更についてのクイックリファレンス	385
	変更一覧表	385
	用語集	393
	索引	397

はじめに

Solaris™ 7 環境は、米国 Sun Microsystems, Inc. の分散コンピューティングソリューションであり、SunOS™ 5.7 と ONC™、OpenWindows™、ToolTalk™、DeskSet™、さらに他のユーティリティから構成されています。このマニュアルは、すでに SunOS 4.x に慣れているユーザを対象に、SunOS 4.x と SunOS 5.7 の違いを説明することに焦点を当てています。さらに、Solaris 7 環境への移行に際して、他の重要な項目についても取り上げています。

Solaris 7 環境で利用できるようになった機能についての詳細は、『*OpenWindows ユーザーズガイド*』を参照してください。

このマニュアルに記されているシステム管理ツールは、Solaris 7 の一部でありローカルシステム管理でのみ使用可能です。システム管理ツールは、Solstice™ ファミリにより提供されるシステムネットワークを管理するために使用されます。

注 - 「x86」という用語は、マイクロプロセッサチップのインテル 8086 ファミリを指します。これには、Pentium、Pentium Pro の各プロセッサと AMD 社製および Cyrix 社製の互換マイクロプロセッサチップが含まれます。このマニュアルでは、このプラットフォームのアーキテクチャ全体を指すときに「x86」という用語を使用し、製品名では「Intel 版」という表記で統一しています。

対象読者

このマニュアルは、SunOS リリース 4.x 環境から Solaris 7 環境へ移行する、ユーザ、システム管理者、ソフトウェア開発者を対象にしています。

このマニュアルの目的

このマニュアルは、Solaris 7 環境への移行を容易にするため、SunOS 4.x 環境と SunOS 5.6 環境の主な違いをユーザが把握することを目的としています。したがってこのマニュアルは、広い範囲の項目を網羅しています。作業手順の詳細については、このマニュアルで必要に応じて紹介している Solaris 7 マニュアルセットの各マニュアルを参照してください。

このマニュアルの構成

このマニュアルは2つのパートに分けられ、全部で19の章と6つの付録から構成されています。概要は次のとおりです。

Part 1— ユーザとシステム管理者のための移行情報

Part 1 は、Solaris 7 ソフトウェアインストール時の注意点、ローカルコンピューティング環境の変化や、日常の作業の変更点を理解するのに役立ちます。

この Part は、次の章から構成されています。

- 第1章では、Solaris オペレーティング環境へ移行することの利点について説明し、SVR4 (System V リリース 4) と Solaris オペレーティング環境の主な違いを要約します。
- 第2章では、SunOS 4.x と SunOS 5.6 の主な違いを説明します。これ以降の章で取り上げられている項目の背景を説明し、これらのリリース間で変更された、手順、ツール、概念に焦点を当てています。
- 第3章では、ソフトウェアのインストールとその後の作業を通じて、移行を容易に行うために考慮すべき点を取り上げます。その結果、SunOS 4.x のデータを Solaris 7 環境に容易に移行することができます。
- 第4章では、SunOS/BSD ソース互換パッケージとバイナリ互換パッケージについて説明します。これらのパッケージにより、Solaris 7 環境への移行の際に、SunOS 4.x のコマンドとアプリケーションを使用することができるので、移行は容易になります。

- 第 5 章では、SunOS 4.x と Solaris 7 のセキュリティの主な違いと、それらの違いがシステム管理作業に及ぼす影響について説明します。
- 第 6 章では、Solaris ソフトウェアをインストールした後で、ローカルのユーザ環境を設定する作業の変更点について説明します。デフォルトシェルの設定、ユーザ環境のカスタマイズ、ウィンドウシステム、ユーザとグループの管理についても説明します。さらに、マニュアルページの変更点についても取り上げます。
- 第 7 章では、SunOS 5.6 のデバイス命名規則について説明します。また、ディスク情報の入手といったデバイスに関連する作業、システムへのデバイス追加、ボリュームマネージャの使用方法などに対する変更点も説明します。
- 第 8 章では、システムのブートとシャットダウンの手順の変更について説明します。
- 第 9 章では、ファイルシステムのレイアウトの変更、またファイルシステム、仮想ファイルシステム、ディレクトリ、およびファイルの変更点について説明します。また、ファイルシステムの管理に関する変更についても説明します。
- 第 10 章では、クライアント用にサーバを設定する方法を説明します。また、Solaris 7 のサーバが SunOS 4.x のクライアントにサービスを提供するための準備に必要な `discover4x`、`install4x`、`convert4x` という 3 つのプログラムについても説明します。
- 第 11 章では、Solaris 7 ソフトウェアをインストールした後でプリンタの設定と管理を行う方法について説明します。また、プリンタコマンドの変更点についても説明します。システム管理ツール (Admintool) と サービスアクセス機能 (SAF) を使って端末とモデムを管理する方法についても説明します。
- 第 12 章では、ネットワーク機能である TCP/IP と UUCP の変更点について概要を説明します。
- 第 13 章では、NIS+ とドメインネームシステム (DNS) について説明し、NIS+ を NIS と DNS のそれぞれと比較します。
- 第 14 章では、共通デスクトップ環境 (以降「CDE」とします) について説明し、OpenWindows 環境から CDE への移行方法を解説します。

Part 2— 開発者のための移行情報

Part 2 は、主に開発者に関する変更点について説明します。それらの違いと類似点を取り上げ、プログラミング環境に関する内容についても説明します。

この Part は、次の章から構成されています。

- 第 15 章では、コンパイラ、リンカ、デバッガに追加された機能や削除された機能について説明します。
- 第 16 章では、開発環境用のツールと資源の変更について説明します。変更点としては、`ioctl()` リクエスト、`ptrace()` リクエストの値、ライブラリ、`make` と `sccs` が挙げられます。この章では、アプリケーションの互換性を調べる方法、Solaris 7 のパッケージ作成機能、SunOS 4.x のツールの検索方法についても説明します。
- 第 17 章では、プログラミング環境に関する Solaris 7 のネットワーク機能について説明します。また、国際化機能の拡張についても説明します。
- 第 18 章では、システムとデバイスの構成の変更点について説明します。動的にロードされるカーネル、カーネルのレイアウト、`config` コマンドと `boot` コマンド、`/etc/system` ファイルなどを取り上げます。

リファレンスとなる付録

次の付録は、SunOS 4.1 のインタフェースと、他のオペレーティングシステムでの対応機能を示しているため、リファレンスとして活用することができます。この情報は、ユーザ、システム管理者、開発者にとって有用です。付録の構成は次のとおりです。

- 付録 A では、SunOS 4.x と SunOS 5.6 のコマンドを比較します。
- 付録 B では、SunOS 4.x と SunOS 5.6 のシステムコールを比較します。
- 付録 C では、SunOS 4.x と SunOS 5.6 のライブラリルーチンを比較します。
- 付録 D では、SunOS 4.x と SunOS 5.6 のシステムファイルを比較します。
- 付録 E では、`/` と `/usr` ファイルシステムのディレクトリの配置を示します。
- 付録 F は、一般的なコマンド、ファイル、ディレクトリ、またデーモンと標準的なプロセスのクイックリファレンスです。

マニュアルの注文方法

SunDocs™ プログラムでは、米国 Sun Microsystems™, Inc. (以下、「サン・マイクロシステムズ社」とします) の 250 冊以上のマニュアルを扱っています。このプログラムを利用して、マニュアルのセットまたは個々のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、米国 SunExpress™, Inc. のインターネットホームページ <http://www.sun.com/sunexpress> にあるカタログセクションを参照してください。

表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、およびディレクトリ名を示します。または、画面上のコンピュータ出力を示します。	.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	ユーザが入力する文字を、画面上のコンピュータ出力とは区別して示します。	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。
「 」	参照する章や節を示します。また、ボタンやメニューなど、強調する単語を囲む場合にも使用します。	第 5 章「衝突の回避」を参照してください。

コード例は次のように表示されます。

■ C シェルプロンプト

```
system% command [filename]
```

- Bourne シェルおよび Korn シェルのプロンプト

```
system$ command [filename]
```

- スーパーユーザのプロンプト

```
system# command [filename]
```

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

ただし AnswerBook2 では、ユーザが入力する文字と画面上のコンピュータ出力は区別して表示されません。

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」キーを押す）。ただし、キーボードによっては「Enter」キーが「Return」キーの動作をします。

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

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が適宜、併記されています。

マニュアルページの参照

コマンド、システムファイル、ライブラリルーチンの名前が本文中で最初に言及されるときは、詳細な説明が記載させているマニュアルページのセクション番号が追加されています。たとえば「mv(1)」となっているときは、このマニュアルページは、セクション1「*man Pages(1): User Commands*」の中に記載されています

関連マニュアル

Solaris 7 環境に関する詳細は、次のマニュアルを参照してください。

- 『OpenWindows ユーザーズガイド』
- 『OpenWindows ユーザーズガイド (上級編)』
- 『Solaris 2.6 インストールの手引き (SPARC 版)』
- 『Solaris のシステム管理 (第 2 巻)』
- 『NIS+ への移行』
- 『NFS の管理』
- 『Solaris ネーミングの管理』
- 『Solaris ネーミングの設定と構成』
- 『TCP/IP とデータ通信』
- 『バイナリ互換性ガイド』
- 『ソース互換性ガイド』
- 『プログラミングの国際化』
- 『マルチスレッドのプログラミング』
- 『リンカとライブラリのガイド』
- 『プログラミングユーティリティガイド』

サン・マイクロシステムズ社の WWW サイトからヘルプ情報を入手するには

以下の URL にアクセスすると、Solaris の移行に関する追加情報が得られます。

<http://www.sun.co.jp/smcc/solaris-migration/index.html>

Solaris 移行イニシアチブホームページは各種ツール、ドキュメンテーション、および情報を配布するための中心点であり、ユーザが Solaris 2.x に移行する際の助けとなります。ここでは、つねに最新の資源とポイントが表示されています。

パート I ユーザとシステム管理者のための移行情報

パート I は、Solaris 7 ソフトウェアのインストールを容易にし、また、ローカルコンピューティング環境と日常的な作業の変更点について理解するのに役立ちます。



概要

Solaris オペレーティング環境は、強力なツールと機能を追加することにより、ユーザのシステムを拡張します。この概要では、Solaris オペレーティング環境へ移行するメリットについて説明するとともに、SVR4 (System V リリース 4) と Solaris オペレーティング環境の主な違いについても説明します。

- 3ページの「Solaris オペレーティング環境に移行するメリット」
- 6ページの「SVR4 と Solaris オペレーティング環境の比較」

Solaris オペレーティング環境に移行するメリット

UNIX[®]の標準である SVR4 は、主要な UNIX システム (System V、BSD、SunOS、Xenix) を取り込み、UNIX ユーザが実際に使用しているシステムのほとんどを統合しています。Solaris オペレーティング環境は SVR4 に基づくもので、多くのシステムとの互換性、スケーラビリティ、市場へ製品を送り出すまでの時間の短縮といった標準的なオペレーティングシステムの特長を、ソフトウェア開発者、システム管理者、エンドユーザに提供します。また、長年にわたって洗練された、機能的で強力な製品も提供しています。この多くのメリットの中には、Solaris オペレーティング環境の移植性、スケーラビリティ、相互運用性、互換性などが含まれます。

Solaris オペレーティング環境は、SVR4 に基づくものですが、マルチスレッド対応のシンメトリック (対称型) のマルチプロセッシング、リアルタイム機能、セキュリティの向上、システム管理の改善などの面で、機能を拡張しています。

Solaris オペレーティング環境には次のような特長があります。

- SunOS release 5.7 は、UltraSPARC システム対応の SVR4 に基づく 64 ビット Solaris オペレーティングシステム環境と SPARC および x86 システム対応の 64 ビット Solaris アプリケーション環境です。
- SunOS 5.6 は、SVR4 に基づく、32 ビット完全対応のオペレーティングシステムです。
- 相互互換性により、SunOS 5.6 は SPARC™ 上だけでなく、Intel 386、486、Pentium、その他の DOS 互換の CPU でも動作します。
- SVR4 や、ONC ファミリのネットワーキングプロトコルなどの業界標準を採用しています。
- OPEN LOOK ウィンドウマネージャは、グラフィカルユーザインタフェース (GUI) を採用しています。
- CDE は、デスクトップグラフィカルインタフェースとして機能します。このウィンドウ環境を使うと、作業の編成と管理が容易に行えます。デスクトップはウィンドウ、作業スペース、コントロール、メニュー、およびメール、ファイルマネージャ、プリンタ、イメージツール、カレンダーマネージャなどに簡単にアクセスできるフロントパネルで構成されています。
- カレンダーマネージャは、スケジュール管理のアプリケーションで、アポイントや作業予定を一覧表示します。また、複数のカレンダーを表示する機能により、グループ内のスケジュールの調整が容易になります。
- ファイルマネージャは、グラフィックによる使いやすいツールで、ローカルファイルシステムとリモートファイルシステムを提供します。
- イメージツールを使用すると、40 種類以上のフォーマットのイメージをロードしたり表示したり、保存したりすることができます。
- オーディオは、Motif ベースの新しいアプリケーションであり、AU、WAV、および AIFF の各ファイルの再生と録音を行います。
- Motif Admintool は、ローカルシステム管理を行う基礎となります。
- GUI インストールにより、インストールと更新が容易になります。
- ログベースのファイルシステムをサーバ上に格納しています。
- 対称型マルチプロセッシングや洗練されたマルチスレッドのような、最新のアーキテクチャを採用しています。
- リアルタイムのスケジューリングと、完全にプリエンティブなカーネルにより、オープンシステムのメリットを提供すると同時に、制御系アプリケーションの必要性を満たしています。

- ネットワーク情報サービスプラス (NIS+) は、NIS ネームサービスの上位互換バージョンで、より単純な階層管理、改善されたセキュリティ、迅速な更新を特長としています。
- アプリケーションの移植性によるメリットを必要とするアプリケーション開発者のために、業界標準に準拠しています。
- マルチメディアメールは、オーディオ、グラフィック、ファイルの取り込みをサポートし、メッセージの送信を簡単にします。
- Java Virtual Machine は、Solaris 動作環境に対応した Java™ プラットフォームへのアクセスを可能にします。
- WebNFS™ は、Web ブラウザを通してファイルシステムにアクセスできるようにします。
- AnswerBook2™ ビューアは、ウェブブラウザベースのインタフェースを採用し、サン・マイクロシステムズ社の主要なオンラインドキュメンテーションシステムです。

移植性、スケーラビリティ、相互運用性、互換性

Solaris オペレーティング環境は、移植性、スケーラビリティ、相互運用性、互換性を備えています。

移植性

SunOS は、複数のベンダのプラットフォームに移植されています。アプリケーションバイナリインタフェース (ABI) に準拠しているソフトウェアは、共通のアーキテクチャのマイクロプロセッサを採用したシステムであれば、どのベンダのシステムでも、シュリンクラップ形式のソフトウェアとしてそのまま実行できます。この結果、アプリケーション開発者はソフトウェア開発コストを削減し、製品を迅速に市場へ送り出すことができます。また、ユーザは、ハードウェアをアップグレードしてもソフトウェアアプリケーションを変更する必要がなく、変換に要するコストを最小限に抑えることができます。

スケーラビリティ

アプリケーションはより広く使われるようになり、より強力なシステムを必要とするようになります。拡張を続ける環境で効率の良い動作を実現するには、ソフ

トウェアは幅広い範囲のシステムで動作すると同時に、強力なシステムの処理能力を十分に生かせるものでなければなりません。Solaris オペレーティング環境は、ラップトップからスーパーコンピュータまで、あらゆる大きさのマシンで動作します。

相互運用性

今日、異機種コンピューティング環境が現実のものとなっています。ユーザは、必要とするソリューションを実現するために、多くのベンダからシステムを購入します。異機種環境では、ユーザがネットワーク全体で情報をやりとりしながら、戦略を実行に移すことができるように、標準化と明確なインタフェースが必要とされます。サンのシステムは、今日市場に出回っている一般的なすべてのシステムと相互運用することが可能であり、UNIX 上で動作しているアプリケーションと、容易に情報の交換を行うことができます。

互換性

コンピュータ技術は急速な成長を続けていますが、その一方でベンダは競争力を保つために、コストを最小限に抑え、最大の投資効果を生み出さなければなりません。サンは、新しい技術が採用されても、既存のソフトウェアへの投資が無駄にならないことを保証します。ユーザは、今日のソリューションのメリットを与えられると同時に、明日の技術との互換性も保証されます。

大規模な組織にとってのメリット

Solaris オペレーティング環境は、UNIX オペレーティングシステムをベースとする業界標準へ移行することによるメリットを提供します。アプリケーションの開発と保守のコストは削減され、アプリケーションの移植性は拡張されます。

SVR4 と Solaris オペレーティング環境の比較

この節では、SVR4 と Solaris オペレーティング環境の主な違いについて説明します。Solaris オペレーティング環境で利用できて SVR4 で利用できない機能と、SVR4 で利用できて Solaris オペレーティング環境で利用できない機能を取り上げます。

Solaris オペレーティング環境で追加された機能

Solaris オペレーティング環境は、SVR4 ベースのオペレーティングシステムに、付加価値の高いコンポーネントを追加しています。これらはコンピュータ処理を容易にするとともに、ユーザ、システム管理者、開発者に新しい可能性を提供するものとなります。

一般的に、SVR4 に代表される既存のシステムや Solaris オペレーティング環境は、既存のアプリケーションとの互換性を維持しながら、機能を強化しています。この結果、削除する機能をわずかにとどめ、多くの機能やコマンドを追加しています。

ユーザ用の機能

Solaris オペレーティング環境に含まれる強力な DeskSet アプリケーションのセットは、ユーザ個人の生産性を高めます。DeskSet のアプリケーションすべては、ドラッグ&ドロップのインタフェースを採用しているため、ユーザはマウスを使用するだけで、UNIX の複雑なコマンドを実行することができます。特に重要な機能を次に示します。

- ワークスペースマネージャ - 基本的なウィンドウ管理サービス (オープン、クローズ、移動) と、ユーザがワークスペースをカスタマイズするために必要なツールを提供します。
- デスクトップ統合サービス - ToolTalk、ドラッグ&ドロップ、カット&ペーストで構成されますが、これらはアプリケーション同士を円滑に統合するための基礎となります。
- グラフィックライブラリ - XGL™、Xlib™、PEX™、XIL™、で構成され、2D や 3D のグラフィックアプリケーションをサポートします。
- カレンダーマネージャ - スケジュール管理のアプリケーションです。会議や作業の予定を一覧表示します。期間として、1 日、1 週間、1 カ月のいずれかを選択できます。また、複数のカレンダーを表示する機能により、グループ内のスケジュールの調整が容易になります。複数のカレンダーを重ね合わせることで、会議に適した時間帯が一目でわかります。
- イメージツール - 40 種類以上のフォーマットのイメージをロードしたり表示したり保存したりすることができます。フォーマットとして、PICT、PostScript™、TIFF、GIF、JFIF などがサポートされています。
- この他には、印刷ツール、オーディオツール、シェルツール、時計、テキストエディタなどのツールが含まれます。

システム管理者用の機能

システム管理者は Solaris オペレーティング環境のさまざまな新しいツールにより、分散コンピューティング環境を容易に管理することができます。システム管理者用の機能として次のものが挙げられます。

- 64 ビットアプリケーション開発用の 64 ビット Solaris オペレーティングシステム環境およびアプリケーション環境 (SPARC プラットフォームのみ)。新しい 64 ビットアプリケーションが大きなアドレス空間を処理できるほか、既存の 32 ビットアプリケーションの大半が動作します。
- デバイス情報 - システム管理者は、オプションのユーティリティを使うことにより、インストールされているデバイスの名前、属性、アクセス状況などの情報を得ることができます。従来の UNIX システムにはなかった機能としてデバイス割り当てプールを作成することにより、管理は容易になります。
- ファイルシステムの管理 - これらのユーティリティを使用することにより、システム管理者はファイルシステムの作成、コピー、マウント、デバッグ、修復、マウント解除が行えます。また、ハードファイルリンクと名前付きパイプの作成や削除、ボリュームの管理も行えます。
- プロセス間通信 - 2つのプロセス間通信ユーティリティは、システムのプロセス間通信機能 (メッセージ待ち行列、セマフォ、共有メモリの ID) の作成、削除、状態のレポートを行います。これらのユーティリティは、システムのチューニングに役立つ情報を提供します。
- プロセス管理 - プロセス管理ユーティリティは、システムのスケジューリングを制御します。これらのユーティリティを使用すると、管理者は性能、ログイン、ディスクアクセスの位置に関するレポートを作成し、システムのパフォーマンスをより効率良くチューニングする方法を検討することができます。さらに、システムの実行レベルの変更、アクティブなプロセスの終了、コマンドの実行時間の指定、カーネルによるデフォルトのスケジューリング優先順位の変更、タイムシェアリング、リアルタイム処理などの変更も可能です。
- システムアカウント - アカウントユーティリティを使用すると、システム管理者は CPU、ユーザ、プロセスによるシステムの使用状況を追跡し、より効率の良い優れたリソースの割り当てを検討することができます。
- システム情報 - これらのユーティリティは、システムメモリとシステムの構成についてレポートします。システム管理者は、これらのユーティリティを使用して、システム名やネットワークのノードを変更することができます。
- ユーザとグループの管理 - システム管理者は、これらのユーティリティを使用して、グループとパスワードのデータベース内のエントリの作成と削除、デフォルト

トのホームディレクトリと環境の指定、ユーザログインとシステムログインの管理、グループ ID とユーザ ID の割り当てなどを行うことができます。これらのユーティリティは、一次グループと二次グループの両方をサポートしています。

- **Admintool - Admintool** は OpenWindows 環境下で動作するもので、システム管理機能を提供します。これらの機能は、ローカルシステム上での、ホストの追加、ネットワークの管理、他の多くの日常作業を支援します。
- **自動構成 - Solaris** オペレーティング環境は動的なカーネルを持っており、デバイスがアクセスされたときに、ドライバや他のモジュールをメモリにロードします。インストールの後でカーネルを再構築する必要や、システム管理者がドライバの追加と削除を行う必要はありません。
- **ネットワーク情報サービスプラス (NIS+) - NIS** ネームサービスの上位互換バージョンで、より簡単な階層管理、改善されたセキュリティ、迅速な更新を特長としています。
- **インストール - Solaris** オペレーティング環境はインストール用の GUI を備えているため、インストール作業やアップグレード作業が容易になります。ネットワーク全体にわたる自動インストールや自動アップグレードも可能です。
- **セキュリティ - 自動セキュリティ拡張ツール (ASET)** は、セキュリティを改善するユーティリティです。システム管理者は、このユーティリティを使用することにより、パーミッション、所有権、ファイル内容などを含む、システムファイルの設定をチェックすることができます。ASET は、ユーザにセキュリティ上の潜在的な問題を警告するほか、適切な状況で、指定されたセキュリティレベルに従って、システムファイルのパーミッションを自動的に設定します。
- **AnswerBook2** マニュアルページ形式 - AnswerBook 形式ではなく AnswerBook2(SGML) 形式でマニュアルページを使用できます。これによって、他の AnswerBook2 ドキュメントからマニュアルページに対する直接の操作とリンクが可能になりました。

開発者用の機能

アプリケーション開発者は、Solaris オペレーティング環境に含まれるさまざまなツールキットと機能によって、グラフィカルユーザインタフェースを使用する複雑なアプリケーションを容易に開発することができます。

- **マルチスレッド (MT) カーネル - MT** は、対称型マルチプロセッシングカーネルを提供します。複数のプロセッサが、このカーネルを同時に実行することができます。1つのスレッドによる制御ではなく、複数の独立したプロセッサによる処

理 (マルチスレッド) を想定して、アプリケーションの構造を決めることができます。オペレーティングシステムは、独立した操作のインタリーブを行うことができるため、マルチスレッドの効率は高くなります。マルチスレッドのメリットは、「アプリケーションの多重度」として知られています。

- **STREAMS - STREAMS** とは、柔軟なキャラクタ入出力 (I/O) の基本的なしくみで、SVR4 で実装されています。アプリケーションごとに容易にカスタマイズすることが可能です。
- **拡張された基本データ型 - ID のデータ型 (uid、pid、デバイス ID など) と、その他のデータ型のうち特定のものは、32 ビットに拡張されました。** この結果、大規模なシステムや組織での仕様に適した、オペレーティングシステムのスケラビリティが改善されました。
- **デバイスドライバのインタフェース - Solaris のデバイスドライバには、3 種類のインタフェースがあります。** デバイскарネルインタフェース (DKI)、デバイスドライバインタフェース / デバイскарネルインタフェース (DDI/DKI)、Sun デバイスドライバインタフェース (Sun DDI) の 3 つです。DDI/DKI に準拠しているデバイスドライバは、SPARC プラットフォームにおけるソース互換性やバイナリ互換性が優れているため、開発者は、ある周辺機器用のドライバを 1 つ作成するだけで、すべての SPARC プラットフォーム上の周辺機器をサポートすることができます。
- **デバイスドライバの自動ロード -** ドライバのインストールや、デバイスのアクセスを容易にします。
- **デバイス構成ライブラリ -** デバイス構成情報を取得するための libdevinfo ライブラリが、Solaris 7 ソフトウェアにおいてさらに強固で包括的になりました。詳細については、libdevinfo(3) のマニュアルページを参照してください。
- **動的リンク - Solaris** オペレーティング環境は、ライブラリの動的リンクと静的リンクをサポートしています。リンクはライブラリと実行可能ファイルのバージョン番号を参照して、アプリケーションに適したライブラリ、ルーチン、インタフェースをリンクします。
- **オペレーティングシステム環境 - 64 ビットアプリケーション開発用の 32 ビットの Solaris アプリケーションおよびオペレーティングシステム環境をサポートします。** また、64 ビットアプリケーション開発用の 64 ビットの Solaris アプリケーションおよびオペレーティングシステム環境をサポートすることで、新しい 64 ビットアプリケーションが大きなアドレス空間を処理できるほか、既存の 32 ビットアプリケーションの大半が動作します。
- **WebNFS ソフトウェア開発キット - WebNFS ソフトウェア開発キット (SDK) には、WebNFS による Java アプリケーションのリモートファイルアクセス機能が**

用意されています。NFS プロトコルを直接実装するので、ホストシステムで NFS をサポートする必要はありません。

Solaris オペレーティング環境に含まれない SVR4 の機能

SVR4 の機能の一部が、Solaris オペレーティング環境ではサポートされていません。これらは、AT&T のハードウェアに固有の機能、または、主に SVR3 の機能との上位互換性を保つために用意されている機能であるため、SunOS ユーザにとってはあまり意味のないものです。

Solaris オペレーティング環境は、System V のファイルシステムと、それに関連するユーティリティを含んでいません。これらは、UNIX のファイルシステムに比べて、制約が大きいからです。SVR4 のブートファイルシステムも、従来の SunOS のブートモデルに比べると管理負担が大きいので、採用されていません。

デバイスの自動構成とカーネルの再構築を行う AT&T の SVR4 の一般的なモデルは、完全な動的構成が可能なカーネルに置き換えられています。このカーネルは、現在と将来の SPARC システムのユーザのニーズに応えることができます。

SPARC XENIX はインストールベースでは存在しないため、Solaris オペレーティング環境の SPARC 用リリースは、XENIX アプリケーションとの互換性を持っていません。

Solaris オペレーティング環境には、AT&T の `sysadm` ユーティリティは含まれません。`sysadm` メニューユーティリティは、主に独立したシステムの端末デバイスを対象として設計されているため、ネットワーク全体にわたる分散システムの処理を容易にするという視点からグラフィカルユーザインタフェースが採用されました。Solaris オペレーティング環境は、SVR4 の `sysadm` ユーティリティの基本的な部分を持つユーティリティや構成ディレクトリを提供していますが、`sysadm` ユーティリティ自体は提供していません。

主な変更について

Solaris 7 環境には、SunOS 4.x 環境と似ている点があり、また、異なる点もいくつかあります。このマニュアルのこれ以降の章では、これらのリリースの間で変更された作業手順、ツール、コマンド、また、概念について説明します。

この章では、主な変更点について紹介します。また、第 3 章以降で取り上げられる項目の概念を説明します。この章で取り上げる項目の中には、記述されている説明だけで十分なものもあれば、より詳細な技術的背景の説明を必要とするものもあります。後者の場合は、詳細に説明されている章を示してあります。

- 14ページの「ソフトウェアパッケージとクラスタ」
- 16ページの「ディスクスライス (またはディスクパーティション)」
- 17ページの「デバイスの命名」
- 18ページの「ファイルシステム」
- 20ページの「カーネルの構成」
- 21ページの「自動マウント」
- 24ページの「Admintool」
- 25ページの「ネットワーク情報サービスプラス (NIS+)」
- 25ページの「印刷サブシステム」
- 27ページの「サービスアクセス機能」
- 28ページの「ボリュームマネージャ」

ソフトウェアパッケージとクラスタ

Solaris 7 システムソフトウェアは、パッケージという単位で出荷されます。パッケージとは、ソフトウェア製品に必要なファイルとディレクトリの集合体のことです。クラスタは、パッケージの集合体のことです。

ここで4つのクラスタについて説明します。2番目以降の各クラスタは、その上にあるクラスタのソフトウェアに、別のソフトウェアを追加したものとなっています。

- コアシステムサポート - 必要最小限のソフトウェア構成です。Solaris 7 環境のブートと動作に必要なソフトウェアだけから構成されています。
- エンドユーザシステムサポート - コアシステムサポートに、エンドユーザをサポートする OpenWindows ウィンドウシステムや、関連する DeskSet アプリケーションファイルを追加したものです。このクラスタの中には、エンドユーザに推奨されるソフトウェアが含まれています。
- 開発者システムサポート - エンドユーザシステムサポートに、Solaris 7 環境でソフトウェア開発を行うために必要なライブラリ、インクルードファイル、ツールを追加したものです。ここでいう Solaris 7 環境には、コンパイラとデバッガは含まれていません。
- 全体ディストリビューション - Solaris 7 環境全体が含まれます。

詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

パッケージの管理

パッケージを管理するプログラムにより、ソフトウェアのインストールとアップデートが容易になります。管理システムのソフトウェアと、サードパーティのアプリケーションの管理方法が一貫しているため、管理が簡単になっています。ソフトウェアパッケージを作成するツールは、アプリケーションパッケージ作成ツールのライブラリの中に含まれています。

パッケージのインストールと削除に使用できるツールは2つあります。

- グラフィカルユーザインタフェースプログラム (admintool (1M) のマニュアルページを参照)
- コマンド行ユーティリティ (pkgadd (1M) と pkgrm (1M) のマニュアルページを参照)

グラフィカルユーザインタフェース (admintool)

admintool (admintool コマンドで起動します) を使用して、ローカルシステムやリモートシステムにソフトウェアをインストールすることができます。デフォルトでは、ローカルシステムへインストールされます。

Admintool を使用すると、次のことが行えます。

- ローカルシステムにインストールされているソフトウェアの表示
- ローカルシステムでのソフトウェアのインストールと削除

ソフトウェアのインストールと削除を行うには、スーパーユーザまたはシステム管理グループ (グループ 14) のユーザとして **Admintool** を実行しなければなりません。システム上にすでにインストールされているソフトウェアパッケージを表示する場合は、スーパーユーザでなくてもかまいません。

コマンド行ユーティリティ

コマンド行ユーティリティにより、ソフトウェアパッケージをインストールしたり、削除したり、また、インストールの状況をチェックしたりすることができます。コマンドは次のとおりです。

- `pkgadd(1M)` は、パッケージのインストールを行います。
- `pkgrm(1M)` は、パッケージの削除を行います。
- `pkgchk(1M)` は、パッケージのインストール状況をチェックします。
- `pkginfo(1M)` は、システム内にインストールされているパッケージを一覧表示します。

パッチの管理

`patchadd(1M)` コマンドと `patchrm(1M)` コマンドは、Solaris 2.x システムにパッチをインストールするためと、パッチを除去するために使われます。システム、クライアント、サービス、またはネットインストールイメージに 1 つ、または複数のパッチを追加できます。

詳細は、`patchadd(1M)` と `patchrm(1M)` を参照してください。

ディスクスライス (またはディスクパーティション)

1つの連続ブロック、またはディスクの物理的なサブセットを SunOS 4.x ではディスクパーティションといいます。SunOS 5.xでは、ディスクの物理的なサブセットをディスクスライスといいます。ディスク上でファイルシステムを作成する前に、ディスクをフォーマット化し、いくつかのスライスに分割する必要があります。これは通常、Solaris 2.x インストールプログラムを使って Solaris をインストールする時に行われます。インストール後にディスクをインストールし、フォーマット化しなければならない場合は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

注 - 一部の Solaris のマニュアルでは、Solaris スライスをまだ「パーティション」と呼んでいる場合があります。Solaris 2.x のマニュアルでは、fdisk パーティション (Intel システムの場合) と、fdisk パーティション内部の区分を区別しています。後者の場合、各区分はスライスともパーティションとも呼ばれています。

Solaris fdisk パーティションの説明については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

スライスはスワップ空間用の raw デバイスとして、または1つの UFS ファイルシステムだけを持つデバイスとして使用できます。ただし、Solstice DiskSuite™ のような製品を使っている場合は除きます。表 2-1 に、それぞれの Solaris 2.x プラットフォーム上でディスクスライスをセットアップする方法を説明します。

表 2-1 プラットフォームによるスライスの相違

SPARC	Intel ベース
ディスク全体が Solaris 動作環境専用に使われます。	ディスクは 4 つの fdisk パーティションに分割され、1 つの動作環境に 1 つずつ使われます。
ディスクは 8 つのスライスに分割されており、0 ~ 7 の番号が付いています。	Solaris fdisk パーティションは 10 個のスライスに分割されており、0 ~ 9 の番号が付いています。そのうち、ユーザのデータを格納できるのは 0 ~ 7 だけです。

プラットフォーム別のディスクスライスの慣例的な割り当ての説明については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

シリンダグループ

あるディスクスライス上に UFS ファイルシステムを作成した場合、このディスクスライスは、1 つまたは複数の領域に分割されますが、この領域のことをシリンダグループと呼びます。シリンダグループは、1 つまたは複数の連続したディスクシリンダによって構成されています (ディスク装置は複数のディスクによって構成され、各ディスクの円周方向に沿っていくつものトラックが刻まれています。シリンダとは、いくつかのディスクで、ディスクの中心から同じ距離にあるトラックのことを指します)。ディスクの構造に関する詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

シリンダグループごとに、シリンダグループマップが作成されます。シリンダグループマップには、ブロックの使用状況と利用可能なブロックが記録されます。

図 2-1 は、ディスクスライスとシリンダグループの関係を示しています。

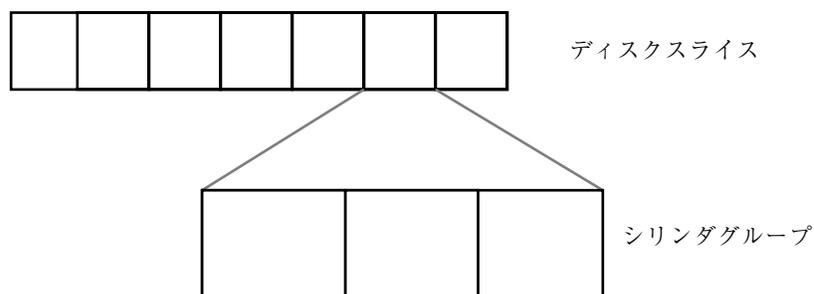


図 2-1 ディスクスライスとシリンダグループ

デバイスの命名

SunOS release 5.7 のデバイス名は、そのデバイスの特徴を容易に推定できるようになっています。SunOS 4.x では、デバイスの属性よりもタイプを表す名前を採用していたため、プログラムやスクリプトがデバイスに関する情報を得ることが困難でした。SunOS release 5.7 では、ディスクに 8 つのパーティションしか設けられないので、AT&T SVR4 とはわずかに異なるデバイスの命名規則を採用しています。

さらに、特殊なデバイスファイルは現在は階層構造の `/devices` ディレクトリに格納されており、これには管理者とユーザがデバイスにアクセスするために使用する階層構造の `/dev` ディレクトリへのシンボリックリンクが付いています。`/dev` ディレクトリには、ディスクデバイスにアクセスするための `/dev/dsk/*`、`raw` ディスクデバイスにアクセスするための `/dev/dsk/*`、などのサブディレクトリが含まれています。詳しくは、63ページの「デバイス命名規則」を参照してください。デバイスの命名規則については、191ページの「開発者に関するデバイスの命名規則」を参照してください。

ファイルシステム

SunOS release 5.7 と SunOS 4.x のファイルシステムは類似していますが、システムディレクトリとシステムファイルの位置と名前には変更が加えられています。SunOS release 5.7 には、新しいファイルシステムと新しい疑似ファイルシステムが追加される一方で、1つのディレクトリが使用されなくなっています。

80ページの「ファイルシステムの変更」では、ファイルシステムの変更について説明しています。『Solaris のシステム管理 (第 1 巻)』では、ファイルシステムの概念と管理について詳細に説明しています。

ファイルシステムの位置と名前の変更

ファイルシステムの位置と名前についての変更は、次のとおりです。

- `/dev` ディレクトリは、平坦な構造から階層構造に変更されました。
- `/etc` ディレクトリには、システム構成情報が含まれるようになりました。ファイルとサブディレクトリの追加、削除、変更が行われています。
- `/etc/vfstab` ファイルは、`/etc/fstab` に変更されました。
- `/etc/lp` ディレクトリは、`/etc/printcap` に変更されました。
- SunOS release 5.7 では、`/sbin` ディレクトリに、システムの実行レベルを変更する `rc` スクリプトと、ファイルシステムをマウントする前にシステムを初期設定する `rfs` スクリプトが含まれています。
- SunOS release 5.7 では、`/usr` ディレクトリに、システムの提供する共有可能ファイルと実行可能ファイルが含まれています。

- /var ディレクトリには、通常の使用の際に大きさが変化するファイルが含まれています。ファイルとサブディレクトリの追加、削除、変更が行われています。
- /var/mail ディレクトリは、/var/spool/mail に変更されました。
- カーネルが動的にロードされるため、/sys ディレクトリは使用されなくなりました。
- terminfo データベースは、termcap に変更されました。
- SunOS release 5.7 のコアカーネルは genunix と呼ばれます。カーネルモジュール (64 ビットバージョンも含む) は /kernel、/usr/kernel、/platform および /usr/platform の各ディレクトリに格納されています。

疑似ファイルシステム

疑似ファイルシステムとは、ディスクベースのシステムに存在する、論理的なファイルグループのことです。SunOS release 5.7 には、TFS 疑似ファイルシステムは含まれていません。

SunOS release 5.7 で使用されている疑似ファイルシステムは、次のとおりです。

- CACHEFS 疑似ファイルシステム - CD-ROM ドライブのような低速デバイスの性能を上げるために使用します。
- PROCFS 疑似ファイルシステム - メモリに常駐し、アクティブなプロセスのプロセス番号を /proc ディレクトリに保持しています。proc(4) のマニュアルページを参照してください。
- FDFS 疑似ファイルシステム - 明示的なファイル名を提供し、ファイル記述子を使用してファイルをオープンします。
- FIFOFS 疑似ファイルシステム - パイプファイルを格納して、複数のプロセスが共通のデータをアクセスをできるようにします。
- NAMEFS 疑似ファイルシステム - 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のが目的で、ほとんどの場合 STREAMS が使用します。
- SWAPFS 疑似ファイルシステム - デフォルトのスワップデバイスで、システムのブート時やシステム管理者が他のスワップ空間を作成するときに使用します。

追加されたファイルシステム

SunOS release 5.7 のディレクトリ構造には、次のファイルシステムも含まれています。

- オプションの /opt ファイルシステム - サードパーティのソフトウェアや、アンバンドルのソフトウェアを格納するために使用します。/opt が単独のファイルシステムではない場合は、/usr/opt へのシンボリックリンクになっています。
- /vol ファイルシステム - ボリューム管理デーモン vold(1M) が使用するデフォルトのファイルシステムを提供します。volfs(7) のマニュアルページを参照してください。

除去されたファイルシステム

RFS ファイルシステムタイプのサポートは除去されました。

カーネルの構成

SunOS 4.x とは異なり、SunOS release 5.7 のカーネルは、動的に構成されます。したがって、システム構成に変更を加えても、システム管理者が手作業でシステムを再構築する必要はありません。

SunOS 5.5 以降、カーネルとそのモジュールはプラットフォーム非依存オブジェクトとプラットフォーム依存オブジェクトに分割されました。プラットフォーム非依存カーネルは /kernel/genunix と呼ばれる小さい静的コアと動的にロード可能なカーネルモジュールで構成されています。動的にロード可能なカーネルモジュールは、プラットフォーム非依存の場合は /kernel ディレクトリと /usr/kernel ディレクトリに格納され、プラットフォーム非依存の場合は /platform ディレクトリと /usr/platform ディレクトリに格納されます。プラットフォーム依存ディレクトリとその内容については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

ドライバ、ファイルシステム、STREAMS モジュール、その他のモジュールは、ブート時か実行時に、必要に応じて自動的にロードされます。modinfo(1M) コマンドは、現在システムにロードされているモジュールに関する情報を表示します。

modload(1M) と modunload(1M) の各コマンドは、SunOS release 5.7 でも利用できますが、動作が変更されています。これらのコマンドの用途は限られたものとなっており、ロード可能ドライバをシステムにインストールするには、もはや十分ではありません。modunload(1M) コマンドは、SunOS 4.x のコマンドに似ていますが、現在は、次の例のように、アンロード可能な (そして使用されていない) すべてのモジュールをアンロードする機能を備えています。

```
# modunload -i 0
```

第 18 章ではこれらの項目について詳細に説明します。

カーネルのレイアウト

カーネルの内容は、以前は 1 つのファイル `/vmunix` に入っていましたが、現在、プラットフォーム非依存とプラットフォーム依存のディレクトリ階層内の複数のモジュールになっています。デフォルトでは、ディレクトリ階層は以下のようになっています。

- `/kernel`
- `/usr/kernel`
- `/platform`
- `/usr/platform`

`/etc/system` ファイル内の `moddir` 変数を使用して、モジュールのディレクトリ検索パスを設定することも可能です。`system(4)` のマニュアルページを参照してください。通常は、`/kernel/genunix` はロードされるカーネルの先頭部分になります。詳しくは、`system(4)` と `kernel(1M)` を参照してください。

自動マウント

AutoFS と呼ばれる新規バージョンの自動マウント機能が組み込まれました。SunOS 4.x では、自動マウント機能は `/tmp_mnt` にすべてマウントして、シンボリックリンクを使ってルックアップをリダイレクトしました。AutoFS では、ファイルシステムを所定の場所 (たとえば、`/home`) にマウントできます。

SunOS 4.x では、自動マウント機能のマップには `auto.master` および `auto.home` という名前が付けられていました。Solaris 7 では、これらのマップの名前は `auto_master`、`auto_home` などに変更されています。この変更は、このリリースに組み込まれている NIS+ ネームサービスが必要とするものです。このリリースにはこれらのマップのデフォルトコピーが組み込まれているため、システムをブートした時に AutoFS サービスが起動します。SunOS 4.x にはマップは組み込まれていなかったため、余分なインストール手順が必要でした。

Solaris 7 には、`/etc/nsswitch.conf` を通して使われるネームサービスを選択できる機能が備わっています。自動マウントのエントリを変更して、ローカルファイル、NIS+、NIS、またはこれらの組合せを選択するようになります。

旧リリースは、`/home/server/login` のようなホームディレクトリの命名規則をサポートしていました。AutoFS マップを使うと、`/home/login` をエントリ別に簡単に使用できるようになります。この新しい命名規則には、位置に依存しない機能も備わっています。古い規則もまだ使用できますが、いったん AutoFS マップの使用に移行すると、短いパスの管理が簡単になります。

AutoFS で使用できるパスには、次のものがあります。

- `/net` - 認識されているホストからファイルシステムをマウントする場合
- `/home` - 認識されているユーザのホームディレクトリをマウントする場合
- `/xfn` - X/Open XFN 標準対応のファイルシステムをマウントする場合

ホームディレクトリサーバでは、実際のホームディレクトリを `/home` ではなく `/export/home` に移動して、自動マウント機能のディレクトリ構造との干渉を防ぐ必要があります。つまり、自動マウント機能が実行している間は `/home` にファイルシステムをマウントすることはできません。

AutoFS ソフトウェアは現在、2つのプログラムを持っています。1つは `automount` であり、ブート時に実行して AutoFS マウントポイントを設定します。このコマンドは、マウントポイントを変更するためにスーパーユーザがいつでも実行できます。もう1つのコマンドは状態なしデーモンである `automount` で、これは AutoFS ファイルシステムのマウントおよびアンマウントの要求に応答します。この2つのプログラムは、4.1.x `automount` デーモンに代わるものです。

自動マウントデーモンは、現在は完全にマルチスレッドに対応しています。複数の自動マウント要求に同時に対応できるため、AutoFS の信頼性が高まります。要するに、1つのマウント要求が遅いサーバへの接続をブロックしている間に、2番目の要求を待たせずに処理することができます。

Solaris 7 では、間接的な AutoFS マップのブラウズ機能をサポートします。AutoFS マウントポイント (例: `/home`) に属するマウント可能なエントリはすべて、最初にそれをマウントしなくても表示できるようになりました。

また、階層的に関連しているファイルシステムの要求時自動マウント機能も改善されました。旧リリースでは、ファイルシステムが階層的に関連している場合は、ファイルシステムの中の1つだけを参照するときでも、そのセット全体 (たとえば、`/net/server`) を自動的にマウントしていました。参照されるファイルシステムは動的にマウントされるようになり、階層内のその他のファイルシステムをマウ

ントする必要がなくなりました。その他のファイルシステムは、個別に参照されたときにマウントされます。

詳しくは、99ページの「ファイルシステムのマウントと autofs」を参照してください。また、AutoFS の使用法については『NFS の管理』を参照してください。

メールの管理

このリリースに組み込まれている sendmail のバージョンは、バージョン 8 と互換性があります。新しいバージョンではセキュリティ上の不備をいくつか修正しており、バージョン 5 を改善した内容も組み込まれています。ネームサービススイッチや NIS+ のサポートなど、標準 BSD リリースの拡張機能もいくつか追加されています。

NIS+ をさらにサポートするために、aliasadm という新しいコマンドが組み込まれています。このコマンドは、NIS+ エイリアスリストの管理に役立ちます。

メールボックスプールディレクトリは、/var/spool/mail から /var/mail に移動しました。新しいディレクトリ /var/mail/:saved が、ロックおよび一時ファイルの作成のために mailx プログラムによって使われます。また、メール構成ファイルはすべて /etc/mail に配置されるようになりました。この新規ディレクトリには aliases ファイルと sendmail.cf ファイルが入っています。

メールボックスロック機構が機能拡張されたため、Solaris 7 のクライアントは Solaris 2.x および SunOS 4.x のメールサーバの両方から安全にメールボックスをマウントできます。この拡張機能によって、とくに大規模なサイトでのメールの管理が容易になります。

Solaris 7 では、/usr/ucb/mail に代わって /usr/bin/mailx が使われています。mailx プログラムは、/usr/ucb/mail の SunOS 4.x バージョンと同じように動作するよう機能拡張されました。/usr/ucb/mail ファイルは、現在は /usr/bin/mailx へのシンボリックリンクになっています。

SunOS 4.x では、sendmail.mx と呼ばれるプログラムを DNS サイトで使ってメール交換レコードにアクセスしていました。新規バージョンの sendmail には必要な機能が組み込まれており、/etc/nsswitch.conf を通して構成できます。

『メールシステムの管理』に sendmail の管理方法を説明しています。

Admintool

システム管理に関する SunOS 4.x と SunOS release 5.7 の主な変更点の 1 つは、基本的な管理タスクを実行する Admintool (システム管理ツール) を利用できるようになったことです。このツールは、グラフィカルユーザインタフェースを採用して、ユーザ、ホスト、プリンタ、シリアルデバイスの管理作業を容易にします。適切なアクセス権が割り当てられていれば、ローカルシステムでこれらの作業を管理することができます。

Admintool のアプリケーションを使用すると、ローカルシステムで次の管理を行うことができます。

- `aliases` や `netmasks` のようなシステムデータベースファイルの管理
- ユーザとグループの追加、機能追加に伴うパスワードの変更、ユーザアカウント情報の削除などを含む、ユーザアカウント情報の管理
- ローカルとリモートのプリンタとリモートプリンタの設定
- 端末とモデムの設定
- パッケージの管理

Admintool のようなグラフィカルユーザインタフェースを使用してシステム管理作業を行うと、次のようなメリットがあります。

- SunOS の多くのコマンドを使用して同じ作業を行う場合より高速です。
- 手作業で編集する場合は、ミスをする危険がありますが、システムファイルは自動的に更新されるので、そのような危険がありません。
- これらのアプリケーションプログラムは適切なシステムデーモンと相互に情報を交換し、これらの同期がとれなくなると、そのことをシステム管理者に通知します。

注 - Admintool を起動するために `root` としてログインする必要はありません。ただし、`sysadmin` グループ (GID= 14) のメンバでなければなりません。

Admintool ウィンドウを表示するには、任意のウィンドウで次のコマンドを入力します。

```
$ admintool &
```

ネットワーク情報サービスプラス (NIS+)

NIS+ とは、Solaris ネットワーク用のネットワーク情報サービスのことで、Solaris ネットワークは NIS を NIS+ の代替機能としても、NIS+ の補足機能としても使用できます。

NIS+ とは、ONC トランスポート独立遠隔手続き呼び出し (transport-independent remote procedure call, RPC) インタフェースの最上位に構築されているネームサービスのことで、NIS+ は、セキュリティ、パフォーマンス、スケーラビリティ、管理の面で、NIS に比べて大きな利点を持っています。NIS+ を使用する利点は、次のとおりです。

- NIS+ は、NIS 環境とデータを共有するので、円滑に移行できます。
- ドメインは階層構造です。サブドメインを作成することもできます。
- ネームサービススイッチ (/etc/nsswitch.conf) を使用して、NIS+、NIS、/etc、DNS のうち、システムが最初にどのネームサービスを試みるか設定することができます。
- AdminSuite を使用して、NIS+ のテーブルを変更し、情報を追加、変更、削除、検索することができます。
- NIS+ で企業全体のネームサービスを作成し、保守することができます。WAN リンクによって接続されている地理的に離れたサイトにも対応できます。
- NIS+ バックアップおよび復元機能を使って、名前空間データセットをすばやく、簡単に保存できます。また、この機能を使って追加のレプリカサーバをすぐにオンライン化することもできます。

詳細は、このマニュアルの第 13 章や、『NIS+ への移行』、『NFS の管理』を参照してください。

印刷サブシステム

Solaris 2.6 リリースの印刷用ソフトウェアから、ネットワーク上のプリンタに対するクライアントアクセスの設定と管理を行うための集中的な環境が実現しました。印刷用ソフトウェアは以下の機能を含んでいます。

- 印刷クライアントソフトウェア - Solstice AdminSuite 管理ツールセット専用だったこのソフトウェアによって、印刷クライアントがネームサービスを介してプリンタを利用できます。
- Admintool - ローカルシステム上の印刷を管理するためのグラフィカルユーザーインターフェース。
- LP 印刷サービスコマンド - 他の印刷管理ツールの内部機能または拡張機能を使用するプリンタの設定と管理を行うためのコマンド行インターフェース。
- Solstice AdminSuite Print マネージャ - ネームサービス環境でプリンタを管理するためのグラフィカルユーザーインターフェース。Solaris 2.6 リリースから利用可能になりました。

詳細は、第 11 章と、『Solaris のシステム管理 (第 2 巻)』を参照してください。

印刷ツールまたはコマンドをシェルで使用しても、上記の基本的なタスクを実行できます。

印刷ツール

印刷ツールとは、Solaris 7 ユーザ環境で利用できるソフトウェアツールのことです。このツールを使用すると OpenWindows や CDE でユーザはプリンタの監視や、印刷ジョブの監視と中止を行うことができます。

コマンドの変更

次に変更されたコマンドを示します。

- lpr は、lp(1) に変更されました。
- lpq は、lpstat(1) に変更されました。
- lprm は、cancel(1) に変更されました。
- troff(1) は、プリンタ名を必要とするようになりました。
- Solaris 7 では、TEX と pscat (C/A/T) とラストイメージフィルタは利用できません。

lp サービスは、複数のデーモン (プロセス) によって構成されて、これらのデーモンは、システムの動作と、/etc/lp ディレクトリ内の構成ファイルの階層を監視しています。また、lp サービスを管理コマンドのセットとみなすこともできます。

サービスアクセス機能

サービスアクセス機能 (SAF) とは、端末、モデム、他のネットワークデバイスの管理に使用するツールのことです。特に、SAF を使用すると次の作業が行えます。

- `ttymon` と `listen` ポートモニタの追加と管理 (`sacadm` コマンドを使用)
- `ttymon` ポートモニタサービスの追加と管理 (`pmadm` と `ttyadm` コマンドを使用)
- `listen` ポートモニタサービスの追加と管理 (`pmadm` と `nlsadmin` コマンドを使用)
- TTY デバイスの管理と問題の解決
- 印刷サービスを要求するネットワーク要求の管理と問題の解決
- サービスアクセス制御 (SAC) の管理と問題の解決 (`sacadm` コマンドを使用)

SAF はオープンシステムソリューションで、TTY デバイスとローカルエリアネットワーク (LAN) を通して、システムへのアクセスとネットワークの資源を制御します。SAF は、適切な定義の行われたインタフェースを提供し、新しい機能の追加と既存の機能の構成を容易にします。

SAF はプログラムではありません。バックグラウンドプロセスと管理コマンドからなる階層によって構成されています。最上位にある SAF プログラムは、SAC です。SAC は、システム管理者が `sacadm` コマンドにより管理するポートモニタを制御します。各ポートモニタは、1 つまたは複数のポートを管理することができます。

`pmadm` コマンドを使用して、ポートに関係するこれらのサービスを管理することができます。SAC 経由で提供されるサービスはネットワークごとに異なりますが、SAC と、管理プログラムの `sacadm` と `pmadm` は、ネットワークの種類に合わせてカスタマイズされてはいません。

表 2-2 は、SAF の制御階層を示しています。`sacadm` コマンドは SAC を管理し、SAC は `ttymon` と `listen` の各ポートモニタを制御します。

表 2-2 SAF の機能と関連プログラム

機能	プログラム	説明
全体的な管理	<code>sacadm</code>	ポートモニタの追加と削除を行うコマンド
サービスアクセス制御	<code>sac</code>	SAF のマスタプログラム

表 2-2 SAF の機能と関連プログラム 続く

機能	プログラム	説明
ポートモニタ	ttymon	シリアルポートのログイン要求を監視する
	listen	ネットワークサービスの要求を監視する
ポートモニタサービスの管理	pmadm	ポートモニタのサービスを制御する
サービス	ログイン、 遠隔手続き 呼び出しなど	SAF がアクセスを提供するサービス access

ttymon と listen のサービスは、pmadm によって順番に制御されます。ある ttymon のインスタンスは、複数のポートへサービスを提供することができます。また、ある listen のインスタンスは、1つのネットワークインタフェースを使用して複数のサービスを提供することができます。

詳細は、第 11 章を参照してください。

ボリュームマネージャ

Solaris 2.2 以降のバージョンでは、CD-ROM とフロッピーディスクデバイスを管理する新しいソフトウェア階層、ボリュームマネージャが採用されています。このソフトウェアは、ユーザが CD-ROM やフロッピーディスクに対して行う操作を自動化します。

CDE OpenWindows ファイルマネージャにボリュームマネージャが利用できるように変更を加え、ファイルシステムを持つ CD-ROM やフロッピーディスクに、すぐにアクセスすることができます。ファイルマネージャの新しい機能の詳細については、『OpenWindows ユーザーズガイド』を参照してください。

システム上でボリュームマネージャの管理を支援する新しいコマンドも追加されています。

詳細は第 7 章 68 ページの「ボリュームマネージャの使用」を参照してください。

SunOS 4.x システムから Solaris 7 オペレーティング環境への変換

SunOS 4.x システムから Solaris 7 環境への変換プロセスは 3 つの段階で構成されており、インストール前 (データのバックアップ)、Solaris オペレーティング環境のインストール、およびインストール後 (データの復元) があります。

この章では、1 つのシステム、またはネットワーク全体でのインストール前とインストール後の段階について説明します。Solaris 7 と SunOS 4.x クライアントの両方に対応できる環境の設定方法については、第 10 章を参照してください。

- 29ページの「インストールに関する変更点」
- 30ページの「Solaris をインストールする前に行うこと」
- 35ページの「インストール前のファイルとファイルシステムのバックアップ」
- 36ページの「Solaris ソフトウェアのインストール」
- 36ページの「インストール後のファイルとファイルシステムの復元」

インストールに関する変更点

Solaris 7 には、システムへのソフトウェアのインストール方法に多くの変更が行われています。したがって、SunOS 4.x ソフトウェアのインストール方法とは異なります。変更点は、次のとおりです。

- Solaris 7 の配布媒体は、コンパクトディスク (CD) だけです。したがって、システム管理者はソフトウェアをインストールするのに、CD-ROM ドライブにアクセ

スできなければなりません。しかし、CD-ROM ドライブが接続されているシステムをインストールサーバとして利用することにより、ネットワークを通してインストールすることができます。ネットワークを経由したインストールの詳細は『Solaris のインストール (上級編)』を参照してください。

- Solaris 7 は、パッケージと呼ばれる複数のモジュールに入って出荷されます。システム管理者は、実際のシステムに合わせてパッケージを選択し、インストールの際に必要なディスク空間を制御し、また個別のシステムで利用できるアプリケーションを指定することができます。

Solaris 7 のパッケージは、クラスタという単位にグループ化されています。この結果、各パッケージを個別に選択するのではなく、関連するパッケージ一式を選択することができます。

- Solaris 7 のインストールでは、セットになったソフトウェアグループ、つまり典型的なユーザを想定した複数のパッケージとクラスタも提供します。個別のパッケージやクラスタを選択するのではなく、システムを動作させるために必要なソフトウェアグループを選択することができます。これは、テストの目的で、Solaris 7 を限定された環境で最初にインストールするときに役立ちます。そのシステムで経験を積んだ後で、パッケージ単位で追加や削除を行うことができます。
- Solaris 7 は、SunOS の従来リリースで提供されていた一般的なカーネル構成ではなく、アーキテクチャ固有のカーネルを提供します。カーネルは、/vmunix ではなく、/kernel にインストールされます。
- Solaris 7 のインストーションプログラムは、ステップごとにインストールプロセスを実行できるように構成されています。
- Solaris 7 環境には、インストール作業を自動化するためのカスタムの JumpStart™ 技術が用意されています。このため、多数のシステムにインストールしなければならない場合に、時間を節約できます。詳しくは、『Solaris のインストール (上級編)』を参照してください。

Solaris をインストールする前に行うこと

インストーションプログラムを実行してソフトウェアをロードすることだけが SunOS 4.x を Solaris 7 に移行するときに必要な作業ではありません。通常は、SunOS 4.x のシステム内にデータが存在するので、それらのデータを Solaris 7 システムに転送しなければなりません。これらのデータとしては、/home のようにファイルシステム

全体的場合や、`/etc/hosts` や `/etc/passwd` のように、ローカルごとにカスタマイズされたシステムファイルの場合があります。

データ転送についてどのような方法を計画する場合でも、インストールを始める前に、フルダンプを行なって全ディスクパーティションをバックアップしてください。Solaris 7 環境ではデバイス名の規則が変更されているため、誤認が発生して、Solaris 7 をインストールしたディスクを間違える可能性もあります。インストールを開始する前にファイルシステムをバックアップすることにより、このようなミスを未然に防ぐことができます。デバイス命名規則の詳細については、63ページの「デバイス命名規則」を参照してください。

ファイルシステムのフォーマットについて、次のことに注意してください。

- Solaris 7 の Extended Fundamental Types (EFT) を使用していない場合、SunOS 4.x で使用されているファイルシステムフォーマットは、ソフトウェアで上位互換、または、場合によっては同じフォーマットになります。
- QuickCheck や Backup CoPilot™ をインストールして SunOS 4.1.1 を実行している場合や、SunOS 4.1.2 を実行している場合は、ファイルシステムのフォーマットは同じです。
- QuickCheck や Backup Copilot™ をインストールせずに SunOS 4.1.1 を使用している場合、または SunOS 4.0.x や SunOS 4.1 を使用している場合は、フォーマットはすべて同じとは限りませんが、上位互換と下位互換のフォーマットとなります。

ディスクパーティション情報の保存

インストールを開始する前に、システム内の既存のディスクパーティションのハードコピーを出力してください。このような情報をディスク上に保存すると、インストールの際に上書きされる可能性があります。Solaris 7 システムを構成するときに多くの決定を下す必要がありますが、既存のディスクパーティション情報を保存しておく、決定を下す際の参考資料となります。ディスクパーティション情報を出力する方法の1つを説明します。

1. システムに接続しているディスクの名前を調べます。
システムに接続しているディスクの名前を調べるには、`format(8)` コマンドを使用します。
2. ディスクパーティション情報を保存します。

各ディスクにコード化された形で記録されているパーティション情報を調べるために、`dckinfo(8)` コマンドを使用します。この出力をプリンタへパイプにより接続します。または、他のシステムに保存するために、リダイレクト先のファイルを指定します。

注 - このコマンドを使用した場合、構成が行われているパーティションの情報だけが得られます。構成が行われていないすべてのパーティションは、「No such device or address.」というメッセージとともに表示されません。

ファイルシステム情報の保存

ファイルシステム名 (例: `/usr`、`/home`) とデバイス名 (例: `/dev/sd0g`) とのマッピングは、構成ファイル `/etc/fstab` に常駐しています。先に進む前に、`/etc/fstab` ファイルの印刷コピーを1部作成しておき、Solaris 7 ファイルの作成に役立ててください。

メタデバイス構成情報の保存

アンバンドル製品である SPARCserver™ Manager や Solstice DiskSuite を実行しているシステムをアップグレードする場合は、この節を読んでください (これらの製品は、複数のディスクのミラーリング、連結、ストライピングを行うために使用します)。

これらの製品を使用しないでシステムをアップグレードする場合、多重パーティションの構成を単一パーティションに変更しなければなりません。特に、連結やストライピングが行われていたファイルシステムは1つのディスクに再構成しなければなりません。また、多重パーティションで使用していたパーティションやミラーリング用のパーティションは使用できなくなります。

アップグレードを行いたいシステムで、SPARCserver™ Manager や Solstice DiskSuite を動作させている場合は、Solaris 7 ソフトウェアをインストールする前に、メタデバイスの構成情報を保存してください。こうすることにより、Solaris 7 ソフトウェアをインストールした後で、メタデバイスの状態を復元できるほか、システムに接続されるディスクのリストを作成する際に、参照することができます。

1. 次の例のように `metastat(8)` コマンドを使用して、情報をプリンタに出力します。

```
# /etc/metastat -p | lpr
```

2. metadb(8) コマンドの出力を印刷します。
たとえば、次のように入力します。

```
# /etc/metadb -i | lpr
```

metadb の出力からは、状態データベースの構成情報がわかります。Solstice DiskSuite を再度インストールする場合は、状態データベースを再構築するために、この情報が必要になります。

バックアップ内容の決定

バックアップして、Solaris 7 ソフトウェアのインストール後に復元したい SunOS 4.x ファイルとファイルシステムのリストを作成する必要があります。

バックアップするシステムコンポーネントのリスト作成

既存の SunOS 4.x 環境にあるすべてのシステムコンポーネントのリストを作成して、ユーザのシステムにとって重要なものを決定します。次のことを考慮してください。

- ローカルに開発されたアプリケーション
- バンドルされていないすべてのソフトウェア製品
- サードパーティのアプリケーション
- サードパーティの周辺装置とドライバ(たとえば、8mm テープドライバおよび SBus カード)

バックアップするファイルとファイルシステムのリスト作成

次のガイドラインに従って、保存するファイルシステムのリストを作成してください。

- 一般規則として、全体として「システム」ファイル(例: /usr または / ファイルシステム)を持っているファイルシステムは転送しないでください。

- /tmp のような一時ファイルシステムは保存しないでください。
- ローカルに変更されたデータファイル、および管理データ用としてサーバが依存するファイルは、抽出して保存してください。後者の例としては、/etc/hosts などいくつかの /etc ファイル、エクスポートされたファイルシステム (exportfs コマンドでリスト表示可能)、/trtpboot ディレクトリなどが挙げられます。これらは安全のために保存すべきファイルです。
- spool やユーザのホームディレクトリのような、ローカルに生成されたデータだけを持つファイルシステムは完全に保存してください。
- SunOS 4.x クライアント用にサーバを移行している場合は、クライアントに関する情報を持つファイルシステムを保存してください。このようなファイルの代表例は /export です。

バックアップする SunOS システム構成ファイルのリスト作成

Solaris プラットフォーム用にマージしたり変換できる SunOS 4.x システム構成ファイルはたくさんあります。以下の例のリストを使って、バックアップしたいシステム構成ファイルの選択に役立ててください。

注 - このリストには推奨されるファイルが含まれています。各項目をよく調べた上で、サイトの構成に応じて項目をリストに追加したり、リストから削除してください。たとえば、サードパーティのソフトウェアベンダのディレクトリに特殊なファイルが含まれている場合は、それを保存する必要があります。

システムが NIS マスタサーバの場合は、NIS マスタディレクトリ (例: /etc) に入っているファイルをすべて保存する必要があります。さらに、NIS に追加したその他のマスタファイルもすべて保存してください。バックアップするファイルとして、次のものを推奨します。

- ./cshrc
- ./profile
- ./login
- ./logout
- ./rhosts
- ./etc (システムが NIS クライアントであるか、またはネームサービスを使っていない場合)
- ./var/spool/calendar

- `./var/spool/cron`
- `./var/spool/uucp`
- `./var/nis` (システムが NIS マスタサーバである場合)
- `./tftpboot` 内のブートプログラム

ディスク空間の所要量の決定

Solaris 7 アップグレードに移したいファイルシステムのディスク空間の所要量リストを作成します。Solaris 7 インストールプログラムの実行時に SunOS 4.x ファイルシステム用にディスク空間を区分できるため、Solaris 7 ソフトウェアをインストールする時にこのリストを参照してください。

ネットワークのインストール順序の決定

ネットワークを移行するときは、ユーザにとって最も便利な状況を想定して、SunOS 4.x から Solaris 7 ソフトウェアをインストールするシステムの順序を決定します。たとえば、サーバのアップグレードを始める前に、すべてのクライアントをアップグレードしたいと考えることもあります。最初にアップグレードするシステムとして、CD-ROM ドライブがローカル接続されている、スタンドアロンのシステムを選択するようにしてください。

しばらくの間は、SunOS 4.x と Solaris 7 システムの両方が混在するネットワークを管理することになるでしょう。そのため、計画の一部として、どちらを優先するか決定しておくべきです。たとえば、1つのドメインをアップグレードし、システム管理のテストや内部で開発されたアプリケーションの移植を行なった後、ネットワーク環境全体をアップグレードすることも考えられます。

インストール前のファイルとファイルシステムのバックアップ

SunOS 4.x システムからバックアップしなければならないファイルまたはファイルシステムが決まったら、SunOS 4.x マニュアルに記載されている標準のコマンドとプロシージャを使ってバックアップを実行できます。使用するコマンドは、テープドライブがローカルかリモートかによって異なります。データの転送をどのように

処理するかには関係なく、インストールプロセスを開始する前に完全なダンプを行って、すべてのディスクパーティションをバックアップするようお勧めします。

Solaris ソフトウェアのインストール

Solaris 7 ソフトウェアをサーバやスタンドアロンシステムにインストールするときは、『Solaris 2.7 インストールの手引き (SPARC 版)』または『Solaris 2.7 インストールの手引き (Intel 版)』に記載されているインストールの手順に従います。

保存オプション

Solaris 7 の対話型インストールプログラムには、インストール中に既存のファイルを保存するための、保存画面があります。このため、SunOS 4.x ファイルをそのまま保存できるので、復元する必要がありません。

SunOS 4.x ファイルシステムを保存できない、または保存したくない (システムディスクのパーティションを変更するため) 場合、復元する SunOS 4.x ファイルシステム用に十分なディスク容量を準備して新しいファイルを作成します。Solaris をインストールした後、新しいファイルシステムに SunOS 4.x ファイルシステムを復元します。

インストール後のファイルとファイルシステムの復元

この節では、Solaris 7 ソフトウェアをインストールする前にバックアップしたユーザデータとシステムデータの復元に関する項目を説明します。

SunOS 4.x ファイルシステムとユーザファイルの復元

Solaris 7 のインストール時に作成した新しいファイルシステムに保存できなかったり、保存しないように選択した SunOS 4.x ファイルシステムは、復元することができます。バックアップおよび復元手順の詳細については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

注 - 先に進む前に、目的のスライスが復元しようとしているファイルシステムを収容するのに十分な大きさであることを確認してください。

バックアップした SunOS 4.x ユーザファイルをどれでも復元し、それを新しいシステムにコピーします。

SunOS 4.x システム構成ファイルの復元

最初に、SunOS 4.x システム構成ファイルを Solaris 7 システムの一時ディレクトリに復元する必要があります。この情報をシステムの一時ディレクトリに復元したあとで、それが Solaris 7 の動作環境で使用できるようにしなければなりません。データの中にはファイルにマージするだけでよいものもありますが、データのタイプによっては新しいフォーマットに変換しなければなりません。

システムの構成により、どちらのタイプを使用するかが決まります。次の指示に従って、ファイルの結合か変換を行い、復元を完了させてください。

- ネームサービスを利用しないシステム: システムがネームサービスを利用しない場合、/etc や /var に配置されていた、該当するすべてのファイルの結合や変換を行います。
- NIS クライアントのシステム: システムが NIS クライアントの場合は、/etc と /var に配置されていた、ローカルのシステム構成ファイルだけの結合や変換を行います。これらのファイルは、NIS ネームサービス経由では提供されません。
- NIS マスタサーバのシステム: システムが NIS マスタサーバの場合は、NIS マスタディレクトリ (たとえば、/etc) に配置されていた、すべてのファイルの結合や変換を行います。さらに、/etc や /var にある他のローカル構成ファイルも更新します。

結合すべきファイル

次のファイル内のデータを利用できる状態にするため、Solaris 7 の同じ名前のファイルに、変更内容を結合してください。ただし、これらのファイルすべてが、SunOS 4.x のシステムで変更されたとは限りません。SunOS 4.x のシステムで変更されたファイルだけを見つけ、それらのファイルだけを結合してください。このリストにおいて、一部のファイル名にわずかですが変更があることに注意してください。たとえば、/etc/auto.* ファイルは、/etc/auto.* に変更されています。

次に、この章の前半の指示に従ってバックアップされた、SunOS 4.x のファイルリストの例を示します。これらのファイルは、Solaris 7 環境で結合の対象となるファイルです。付録 D を参照して、SunOS 4.x ファイルに変更がないか調べてください。

- /etc/auto.master と他のファイルを含む、すべてのオートマウントマップ
- /etc/aliases
- /etc/bootparams
- /etc/ethers
- /etc/hosts
- /etc/format.dat
- /etc/inetd.conf
- /etc/netmasks
- /etc/networks
- /etc/protocols
- /etc/publickey
- /etc/rpc
- /etc/services
- /etc/hosts.equiv
- /etc/remote
- /.cshrc
- /.profile
- /.login
- /.logout
- /.rhosts
- /var/spool/cron
- /var/spool/mail
- /var/spool/calendar
- /var/spool/uucp

変換すべきファイル

/etc/fstab を含む多くのシステムファイルは、Solaris 7 環境では別のファイルに置き換えられ、削除されています。このようなファイルに記録されていた情報を取り出し、Solaris 7 環境に手作業で変換しなければなりません。SunOS 4.x のファイルが変更されているかどうか調べるには、付録 D を参照してください。



注意 - Solaris 7 ソフトウェアをインストールした後は、SunOS 4.x システムのオペレーティングシステム実行可能ファイル (/usr/bin にあるシステムコマンドなど) を復元することは避けてください。

Solaris 7 のシステムにデータを結合する前に、次のファイルを変更しなければなりません。

- /etc/uucp - UUCP システムには、いくつかの変更点があります。Config、Grades、Limits の各ファイルは、Solaris 7 環境で更新されています。Devconfig、Devices、Dialcodes、Dialers、Permissions、Poll、Sysfiles、systems の各ファイルは、SunOS 4.x でもソフトウェア Solaris 7 でも共通です。これらのファイルは、結合できます。このほかに、SunOS 4.x のファイルのうち、Solaris 7 では使用されていないものもいくつかあります。
- /etc/group - このファイルの基本的なフォーマットは、SunOS 4.1 と SunOS 4.1.x で共通しています。しかし、以前のリリースでは、NIS マップから選択して取り出したエントリをグループに取り込むために、プラス (+) かマイナス (-) どちらかの記号を使用していました。Solaris 7 環境での互換性が必要な場合は、group(4) のマニュアルページを参照してください。
- /etc/netgroup - SunOS 5.6 には、/etc/netgroup ファイルはありません。
- /etc/exports - Solaris 7 環境のネットワーク上で共有されているファイルシステムは、/etc/exports ではなく、/etc/dfs/dfstab ファイルを使用します。このファイル内のエントリは、次のようになっています。

```
share -F fstype -o options -d "text" pathname resource
```

詳細は、dfstab(4) のマニュアルページを参照してください。

- /etc/fstab - Solaris 7 環境でマウントされるファイルシステムは、/etc/fstab ではなく、/etc/vfstab ファイルを使用します。/etc/fstab ファイル内のエントリは、次のようになっています。

```
dev raw_dev mnt_pt fs_type
fsck_pass auto_mnt mnt_option
```

詳細は、`vfstab(4)` のマニュアルページを参照してください。

- `/etc/passwd-passwd` ファイルのフォーマットは、SunOS 4.x ソフトウェアのフォーマットと同じです。ただし、ユーザのパスワードは、`/etc/shadow` ファイルに保存されるように変更されました。詳細は、`passwd(4)` と `shadow(4)` のマニュアルページを参照してください。
- `/etc/sendmail.cf-sendmail.cf` のフォーマットは、SunOS 4.x のフォーマットと同じです。ただし、ファイルの位置は `/etc/mail/sendmail.cf` に変更されました。
- `/etc/ttytab` – SunOS 4.x システムでは、`ttytab`、シリアルポートと、シリアル回線上の端末の特性を制御するために使用されていました。Solaris 7 環境では、これらの特性を構成するために、サービスアクセス機能 (SAF) が使用されています。
- `/etc/printcap` – Solaris 7 環境では、SVR 4 の印刷サービスを使用してプリンタの構成を行なっています。詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

互換パッケージの使用方法

SunOS 5.6 は、SunOS 4.x ソフトウェアとはソースレベルの互換性も、バイナリレベルの互換性もありません。つまり、リリース 4.x に基づくプログラムやユーザアプリケーションは、Solaris 7 環境では正しく動作しません。しかし互換パッケージを使うことによって、これらのプログラムを Solaris 7 システムで動作させることができます。

この章では、SunOS/BSD ソース互換パッケージと、バイナリ互換パッケージという、2つの互換パッケージについて簡単に説明します。これらのパッケージは、環境とアプリケーションを Solaris 7 環境に移行する間、SunOS リリース 4.x のコマンドやアプリケーションを使用できるようにして、移行を容易にするものです。

- 41ページの「アプリケーションを移植する理由」
- 42ページの「SunOS/BSD ソース互換パッケージ」
- 42ページの「バイナリ互換パッケージ」

SunOS 4.x のコマンドの一部は、Solaris 7 では利用できません。Solaris 7 にはあっても、変更されているコマンドもあります。Solaris 7 環境で変更されている SunOS 4.x のコマンドに関する詳細は、付録 A を参照してください。

アプリケーションを移植する理由

SunOS のバイナリ互換パッケージと SunOS/BSD ソース互換パッケージを使用すると、そのままアプリケーションを利用することができます。ただし、なるべくア

アプリケーションを移植するようにしてください。次の理由から、互換パッケージを長期的に使用することは推奨できません。

- アプリケーションのパフォーマンスが低下します。
- Solaris 7 環境で拡張された操作や移植性のメリットを生かすことができません。
- 互換パッケージは、サイトの移行を一時的に支援するためのものです。

SunOS/BSD ソース互換パッケージ

SunOS/BSD ソース互換パッケージは、Solaris 7 環境で利用できる、オプションのパッケージです。このパッケージは、SunOS 4.x と BSD のコマンド、ライブラリルーチン、ヘッダファイルのうち、互換パッケージがないと Solaris 7 環境で利用できないものから構成されています。SunOS/BSD ソース互換パッケージを使用するときは、バイナリ互換パッケージもインストールしなければなりません。

SunOS/BSD ソース互換パッケージのインタフェースは、`/usr/ucb` ディレクトリにインストールされるので、SunOS 5.6 の既存のインタフェースとの衝突を避けることができます。これらのインタフェースは、環境とアプリケーションを SunOS 5.6 ソフトウェアに移行する間、今まで慣れてきた SunOS の環境を提供するものとなります。これらのインタフェースを使用するには、フルパス名を指定するか、`PATH` 環境変数を変更するか、どちらかが必要です。`PATH` 環境変数を変更する場合は、`/usr/bin` より先に `/usr/ucb` を記述すべきことに注意してください。

ソース互換パッケージに関する詳細は、『*Solaris Source Compatibility Guide*』を参照してください。

バイナリ互換パッケージ

バイナリ互換パッケージは、Solaris 7 環境で利用できるオプションのパッケージです。このパッケージを使用すると、SunOS 4.x の既存のアプリケーションが実行時に静的または動的なリンクを行い、変更や再コンパイルを行うことなく Solaris 7 環境で動作することができます。2つのリリースの間には、バイナリインタフェースの違いが存在しますが、このパッケージはユーザにそれらを意識させることなく、ほとんどの違いを処理します。この結果、Solaris 7 環境で、SunOS 4.x のアプリケーションが正しく動作します。

このパッケージにアクセスする環境を設定する方法は、『バイナリ互換性ガイド』を参照してください。このマニュアルは、バイナリ互換パッケージの制約についても詳細に説明しています。

SunOS 4.x のアプリケーションを実行するためのバイナリ互換パッケージの使用法

バイナリ互換パッケージを使用すると、Solaris 7 環境でほとんどのアプリケーションを動作させることができるので、SunOS 5.6 に移植するまでの間、それらのアプリケーションを利用することができます。このパッケージを使うと、「お行儀の良い」アプリケーションは、変更や再コンパイルを行うことなく、SunOS 5.6 の下で動作します。

バイナリ互換パッケージは、開発環境ではなく、エンドユーザの環境で使用することを想定しています。SunOS 5.6 ソフトウェアのアプリケーションの開発は、SunOS 5.6 の環境で行うようにしてください。

セキュリティ

Solaris 7 環境のセキュリティは、SunOS 4.x の機能と AT&T SVR4 の機能を組み合わせたもので、また、Solaris 7 Soralis 2.6 に特別に追加された機能も含まれています。SunOS 4.x のセキュリティプログラムのパッケージも変更されています。

この章では、SunOS 4.x と Solaris 7 環境のセキュリティの主な違いについて説明し、この変更がどのようにシステム管理手順に影響するかを示します。『Solaris のシステム管理 (第 2 巻)』では、管理やこれらの機能の使用方法をより詳細に説明しています。

- 45ページの「Solaris 7 のセキュリティ機能」
- 47ページの「パスワード有効期限の変更」
- 49ページの「自動セキュリティ拡張ツール (ASET)」
- 49ページの「セキュリティオプション」

Solaris 7 のセキュリティ機能

SunOS 4.x システムのセキュリティ機能のうち、ほとんどが Solaris 7 の環境でも使用できます。これらの機能には次のものがあります。

- インターネットセキュリティ
- .rhosts ファイルと .rhosts.equiv ファイル
- Secure RPC と Secure NFS™

GSS-API に基づいて RPC が変更されました。その結果、セキュリティの完全性と信頼性が向上し、NFS サービスが特定または単独のセキュリティ機構に拘束されなくなります。また、認証キーの長さを 192 ビットから 640 ビットに拡張することで、NIS+ セキュリティを強化しています。

『NFS の管理』では、Secure NFS と `.rhosts` ファイルについて説明しています。『TCP/IP とデータ通信』では、インターネットセキュリティの管理について説明しています。

ローカルの SunOS 5.6 システムのセキュリティには、別ファイルへの暗号化されたパスワードの格納や、ログインデフォルトの制御、制限付きシェルなどが含まれます。『NIS+ への移行』と『NFS の管理』で説明されている、同じ機能の NIS+ セキュリティは、システムへのアクセスをネットワーク全体で制御します。

次の項では、ローカルでのシステム制御におけるセキュリティ機能について要約しています。

`/etc/passwd` と `/etc/shadow` ファイル

SunOS 5.6 の `passwd` コマンドは暗号化したパスワードを別のファイル `/etc/shadow` に格納し、シャドウファイルへの `root` のアクセスのみを許可します。これにより、以前の `/etc/passwd` ファイルにあったような暗号化されたパスワードへのアクセスを防止します。

また、`/etc/shadow` ファイルは個々のユーザログインアカウントのパスワード有効期限の設定を強制的に行うエントリを含んでいます。`passwd` ファイルと `shadow` ファイルへのエントリを変更するメカニズムについては、『Solaris のシステム管理 (第 2 巻)』に説明があります。

`/etc/default` ファイル

デフォルトのシステムアクセスを制御するいくつかのファイルは、`/etc/default` ディレクトリに格納されます。これらのファイルはネットワークの特定のシステムへアクセスすることを制限します。表 5-1 では、`/etc/default` ディレクトリにあるファイルについて要約します。

表 5-1 /etc/default ディレクトリにあるファイル

/etc/default/login	root のアクセスを含むシステムログイン条件を制御する。デフォルトでは、root のアクセスをコンソールに制限する。
/etc/default/passwd	パスワード有効期限のデフォルト条件を制御する。
/etc/default/su	システムにアクセスする root (su) のログをどこにとるか、またその root のアクセスをどこに表示させるかを制御する。

制限付きシェル

システム管理者は、制限付きの Korn シェル (rksh) および Bourne シェル (rsh) を使用して特定のユーザアカウントに対する操作を制限できます。

制限付きシェルは次のような操作を許可しません。

- ディレクトリの変更
- \$PATH 変数の設定
- 「/」で始まるパスまたはコマンド名の指定
- 出力のリダイレクト

これらのシェルの説明については、ksh(1) と sh(1) のマニュアルページを参照してください。

制限付きシェルとリモートシェルには異なるパス名を持つ同じコマンド名 (rsh) があることに注意してください。

- /usr/lib/rsh は制限付きシェル
- /usr/bin/rsh はリモートシェル

パスワード有効期限の変更

SunOS 5.6 システムには、パスワードに有効期限を設定する機能があります。この機能は、各ユーザのパスワードに限られた有効期間を割り当てて、パスワードの機密を保持します。パスワードの有効期間が終了すると、パスワードの所有者に通知し、新しいパスワードを選ぶよう要求します。

パスワード有効期限の設定は、次のどちらかの方法で実装することができます。

- 方法 1 - X Window を実行している場合は、Admintool を使ってユーザを管理します。この方法については、『OpenWindows ユーザーズガイド (上級編)』を参照してください。
- 方法 2 - 新しい passwd コマンドまたは nispasswd コマンドオプションを使います (どのネームサービスがアカウントを格納しているかによります)。

システム管理者もパスワードの有効期間を設定できます。

ユーザパスワードは次のどちらかの方法で変更できます。

- 方法 1 - どのネームサービスがアカウントを格納しているかにより、passwd または nispasswd のどちらかを使用します。
- 方法 2 - X Window が動作している場合、Admintool を使ってユーザを管理します。この方法については、『OpenWindows ユーザーズガイド (上級編)』を参照してください。

passwd や nispasswd についての詳細は、付録 D を参照してください。

アクセス制御リスト (ACL)

アクセス制御リスト (ACL) は UFS と NFS のどちらでもサポートされており、ファイル許可を管理する際に従来の UNIX ファイル保護よりも柔軟に対応できます。従来の UNIX ファイル保護では、所有者、グループ、その他という 3 つのユーザクラスに対して読み取り、書き込み、実行の許可が与えられます。

ACL を使用すると、ファイルの許可を所有者、所有者のグループ、その他、特定のユーザとグループに定義することができ、これらのカテゴリの 1 つずつにデフォルトの許可を定義できます。たとえば、あるユーザのグループに読み取り許可を定義し、そのグループ内の 1 人のユーザだけに書き込み許可を定義する ACL を設定することができます。標準の UNIX ファイル許可では、このような設定はできませんでした。

setfacl(1) コマンドは ACL エントリの設定、追加、変更、および削除を行い、getfacl(1) コマンドは ACL エントリを表示します。

ACL の使用法については、『Solaris のシステム管理 (第 2 巻)』を参照してください。

自動セキュリティ拡張ツール (ASET)

SunOS 4.x システムでは別売オプションとして利用できた自動セキュリティ拡張ツール (ASET) は、Solaris 7 の動作環境に組み込まれています。ASET では、全体のシステムセキュリティレベル (low、medium、high) を指定し、それらのレベルで自動的にシステムを管理できます。このツールは、サーバやそのすべてのクライアント、または個々のクライアントが実行するように設定できます。

ASET は次の作業を実行します。

- システムファイルのパーミッションの検証
- システムファイルの内容の検証
- グループファイルエントリの整合性の確認
- システム構成ファイルの確認
- 環境ファイルの確認 (.profile、.login、.cshrc)
- コンソールログインアクセスを制限するための、EEPROM 設定値の検証
- ファイアウォールまたはゲートウェイシステムの確立

『Solaris のシステム管理 (第 2 巻)』では、ASET の設定と監視について詳細に説明しています。

セキュリティオプション

現在使用できるアンバンドルのセキュリティオプションは、Kerberos と SunSHIELD™ と Pluggable Authentication Module (PAM) です。

Kerberos 4.0 セキュリティ

Solaris 7 の環境には、Secure RPC における Kerberos V4 認証のサポートが含まれます。Kerberos のソースコードや管理ユーティリティは MIT から入手できます。

このリリースに含まれるものは以下です。

- Kerberos を使用できるクライアントアプリケーションライブラリ
- Secure RPC における Kerberos オプション

- Kerberos を組み込んだ Sun の NFS 分散型コンピューティングファイルシステムアプリケーション
- クライアントのユーザチケットを管理するためのコマンド

『Solaris のシステム管理 (第 2 巻)』では、クライアント側のユーティリティを使用する方法について説明しています。また、『NFS の管理』では、NFS で Kerberos を使用する方法について説明しています。

SunSHIELD パッケージ

Solaris 7 には、SunSHIELD 基本セキュリティモジュール (BSM) パッケージが組み込まれています。この製品は、トラステッドコンピュータシステム評価基準 (TCSEC) の中に C2 として定義されているセキュリティ機能を備えています。BSM によって提供される機能は、セキュリティ監査サブシステムとデバイス割当て機構です。C2 の任意アクセス制御機能と識別および認証機能は、オペレーティングシステムに組み込まれています。

BSM の管理については、『SunSHIELD 基本セキュリティモジュール』で説明しています。

PAM

プラグ可能な認証モジュール (PAM) フレームワークは login、ftp、telnet などのコマンドを変更せずに新しい認証技術を「プラグイン」できるものです。このフレームワークを使うことによって、システム管理者はサービスを任意に組み合わせで認証を提供できます。また、アカウント、セッション、パスワードの管理機構も、このフレームワークを使って「プラグイン」できます。

PAM の管理については、『Solaris のシステム管理 (第 2 巻)』で説明しています。

ユーザ環境の管理

この章では、Solaris 7 ソフトウェアをインストールした後に、ローカルユーザの環境を設定する際の作業の違いについて説明します。

- 51ページの「デフォルトシェルの選択」
- 53ページの「ユーザ環境のカスタマイズ」
- 54ページの「ウィンドウシステム」
- 55ページの「ユーザとグループの管理」
- 56ページの「メールの使用」
- 57ページの「文書ツールの使用」
- 58ページの「マニュアルページの構成の変更」

デフォルトシェルの選択

ログインシェルは、ログイン時に実行されるコマンドインタプリタです。Solaris 7 環境には、次の3つのシェルがあります。

- Bourne シェル、デフォルトシェル (/bin/sh)
- C シェル (/bin/csh)
- Korn シェル (/bin/ksh)

シェルを頻繁に使用するのであれば、対話形式で行える C シェルか Korn シェルを使用する方がよいでしょう。表 6-1 は、3つのシェルの機能の一覧です。

表 6-1 Bourne シェル、C シェル、Korn シェルの基本機能

機能	Bourne	C	Korn
sh との構文の互換性	あり	なし	あり
ジョブ制御	あり	あり	あり
履歴リスト	なし	あり	あり
コマンド行編集	なし	あり	あり
別名	なし	あり	あり
ログインディレクトリの 1 文字省略形	なし	あり	あり
上書きに対するファイルの保護 (noclobber)	なし	あり	あり
CTRL-D の無視 (ignoreeof)	なし	あり	あり
拡張 cd	なし	あり	あり
.profile 以外の初期設定ファイル	なし	あり	あり
ログアウトファイル	なし	あり	なし

あるシェルから別のシェルへ変更したいときは、次のどちらかの方法を使ってください。

- 方法 1 - /etc/passwd ファイルの中の、ユーザのログイン名で始まる行の最後のフィールドにある情報を編集してください。この指定がブランクまたは sh の場合、ログインシェルは Bourne シェルです。csh の場合、ログインシェルは C シェルです。ksh の場合、ログインシェルは Korn シェルです。
- 方法 2 - ウィンドウ環境では、Admintool を使います。詳細は『OpenWindows ユーザーズガイド (上級編)』を参照してください。

新しいシェルに変更した後は、ログアウトし、再びログインを行い、シェルを起動します。

ユーザ環境のカスタマイズ

この節では、ユーザが選択したログインシェルに基づいてローカル環境をカスタマイズするために、どの初期設定ファイルを編集するか、また SunOS 5.6 ファイルシステムのどこにその初期設定ファイルがあるかを確認する方法を説明します。初期設定ファイルの変数を編集して、環境設定を行います。使用するデフォルトシェルにより、`.profile`、`.login`、`.cshrc` のうちどのファイルを編集するか決定します。表 6-2 に Bourne シェル、C シェル、Korn シェルの初期設定ファイルを示します。

表 6-2 Bourne シェル、C シェル、Korn シェルの初期設定ファイル

シェル	初期設定ファイル	使用される目的
Bourne	<code>/etc/profile</code>	ログイン時にシステムプロファイルを定義する
	<code>\$HOME/.profile</code>	ログイン時にユーザのプロファイルを定義する
C	<code>/etc/.login</code>	ログイン時にシステム環境を定義する
	<code>\$HOME/.cshrc</code>	ログイン時にユーザの環境を定義する
	<code>\$HOME/.login</code>	ログイン時にユーザのプロファイルを定義する
Korn	<code>/etc/profile</code>	ログイン時にシステムプロファイルを定義する
	<code>\$HOME/.profile</code>	ログイン時にユーザのプロファイルを定義する
	<code>\$HOME/ksh_env</code>	<code>ksh_env</code> 変数によって指定されるファイルを使用し、ログイン時にユーザの環境を定義する

このリリースでは、シェルの初期設定ファイルのテンプレートは、SunOS 4.x ソフトウェアの `/usr/lib` から `/etc/skel` ディレクトリに移りました。テンプレートファイルの位置を表 6-3 に示します。デフォルトシェルを変更するときは、対応す

るテンプレートファイル (1 つまたは 2 つ) を、ホームディレクトリへコピーしてから行なってください。

表 6-3 デフォルトホームディレクトリの起動ファイル

シェル	ファイル名
Bourne	/etc/skel/local.profile
C	/etc/skel/local.login /etc/skel/local.cshrc
Korn	/etc/skel/local.profile

初期設定ファイルの設定の詳細については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

Solaris ソフトウェアでの SunOS 4.x 作業環境の使用

SunOS 4.x 作業環境を引き続き使用したい場合、SunOS 5.6 では、古いシステムファイルと .login、.cshrc、.profile などの初期設定ファイルを使って SunOS 4.x の見た目と使い心地を作成し直すことができます。これらの SunOS 4.x ファイルの多くは、変換するか、またはそのまま使用することができ、簡単に実行できます。

第 3 章のインストール作業で、SunOS 4.x 環境を Solaris 7 環境で作成し直す方法について説明しています。

ウィンドウシステム

CDE は、Solaris 7 ウィンドウ環境のデフォルトで、簡単で視覚的に理解しやすいインターフェースです。CDE についての詳細は、第 14 章を参照してください。

OpenWindows 3.6 は、Solaris 7 ウィンドウ環境のデフォルトです。OpenWindows 2.0 環境を使用していた場合は、OpenWindows 3.6 のアイコンが変更されているこ

とと、OpenWindows 3.6 プラットフォームと互換性のないアプリケーションがあることに気が付きます。

OpenWindows Developer's Guide File Chooser (gfm) の正規表現ファイルパターンマッチングコード (`filter_pat`) は、XView™ File Chooser オブジェクトの正規表現ファイルパターンマッチングコードとはわずかに異なります。2つの異なる chooser にあるわずかに異なるファイルセットに対応し、同じ正規表現が得られます。XView File Chooser は、SunOS 5.6 では `/usr/include/reexp.h` を使用しており、使用方法は正しいものです。

SunView™ ソフトウェアは、Solaris 7 環境に含まれるものではありません。SunView アプリケーションは OpenWindows 環境には対応していないため、変換する必要があります。

次の情報については、『*OpenWindows Version 3.1 User's Guide*』を参照してください。

- OpenWindows 3.1 環境の機能
- OpenWindows Version 2.0 および 3.1 プラットフォーム間で互換性のないアプリケーション
- 互換性のないアプリケーションを変更するためのガイドライン

ユーザとグループの管理

この節ではユーザとグループを管理するためのオプションについて説明します。

ユーザとグループ管理の選択

`useradd`、`userdel`、`usermod` をコマンド行から入力し、ユーザとグループを追加、変更、削除することができます。これらのコマンドは、Admintool ほど強力ではありませんが、Admintool がサポートしている大部分の作業を OpenWindows や CDE ソフトウェアを実行しないで、コマンド行から実行できます。

`useradd`、`userdel`、`usermod` コマンドはローカルシステムにしか影響を与えない点で、`/etc` 内のファイルを編集するのに似ています。これらのコマンドは、ネットワークネームサービスの情報を変更するのに使用することはできません。しかし、`useradd` を使用して、ネットワークネームサービスでのユーザ名やユーザ ID の重複、グループ名の存在をチェックすることができます。

ユーザアカウントの追加

この節では、ユーザアカウントを追加する一般的な手順の変更について説明します。

SunOS 4.x では、システムに新しいユーザを追加するための一般的な手順は次のようになっていました。

1. /etc/passwd ファイルを編集し、新しいユーザのエントリを追加します。
2. ホームディレクトリを作成し、新しいユーザのパーミッションを設定します。
3. 新しいユーザのスケルトンファイルを設定します (.cshrc、.login、.profile など)。
4. ネームサービス (NIS) に新しいユーザを追加します。

Solaris 7 環境には、ユーザアカウントを追加 (および管理) する方法が3つあります。

- Admintool を使用する – システムが OpenWindows 環境を実行している場合は、これがもっとも簡単な方法です。
- コマンド行インタフェース (useradd、usermod、userdel) を使用する – Admintool を使いたくない場合は、この方法を使います。
- 手作業でファイルを編集する – SunOS 4.x の手順に似ています。ただし、いくつか例外があります。

注 - SunOS 5.6 ソフトウェアはシャドーパスワードファイルを使用しているため、/etc/passwd ファイルを編集するだけでは不十分です。この種の管理作業の経験が十分でない場合は、この方法を使用しないでください。

『Solaris のシステム管理 (第 1 巻)』では、アカウントの設定を始める前に考慮すべき点について詳しく説明しています。また、システムとネットワークのユーザアクセスを制御するための、セキュリティ上の留意点についても説明しています。

メールの使用

SunOS 4.x メールプログラムは、Solaris 7 環境では変更されています。ただし、設定手順は同じです。SunOS 4.x の mail は、SunOS/BSD ソース互換パッケージに含まれています。このインタフェースは Solaris 7 の mail とは異なります。さらに、互換性を確保するために便利なメール機能もいくつか含まれています。

Solaris 7 環境には、メールを送信したり受け取るためのプログラムが3つあります。これら3つのプログラムはすべて下方互換性があり、古い SunOS 4.x メールを読み取ることができます。これらのプログラムは次のとおりです。

- **mailtool** – メールプログラムのための OpenWindows インタフェース。新しい Solaris 7 の **mailtool** オプションによって、メッセージへのファイルの添付、メールへのサードパーティメッセージの組み込み、複数の受信者へのメールの配信、音声メッセージの送信を行うことができます。

mailtool の詳細な説明は、『Solaris ユーザーズガイド』を参照してください。

- **mailx** – /usr/bin/mailx. にインストールされています。これは Solaris 7 のメール読み取りプログラムです。SunOS 4.x の /usr/ucb/mail の機能を拡張させたものです。Solaris 7 環境では、/usr/ucb/mail は /usr/bin/mailx へのリンクです。**mailx** はメッセージヘッダを提供し、メッセージを読む前にそのメッセージの送信者とタイトルを見ることができます。またメールメッセージの読み取り、送信、編集を切り替えることも可能です。

mailx についての詳細は、**mailx(1)** のマニュアルページを参照してください。

- **mail** は /usr/bin/mail にあるメールプログラムを参照します。Solaris 7 インタフェースは SunOS 4.x の /usr/bin/mail バージョン (『SunOS 4.x リファレンスマニュアル』の **bin-mail(1)** のマニュアルページを参照) と似ています。

mail についての詳細は **mail(1)** のマニュアルページを参照してください。

Solaris 7 メールプログラムの詳細は、『メールシステムの管理』を参照してください。

文書ツールの使用

この節では SunOS 4.x と Solaris 7 環境で文書ツールを使用する際の主な違いを説明します。

- Solaris 7 環境は、PostScript フィルタセットとデバイスに依存しないフォントを備えています。しかし、SunOS 5.6 は SunOS 4.x TranScript フィルタの大部分と同等の機能を備えています。ただし、SunOS 5.6 システムでは、TEX フィルタや **pscat (C/A/T)** フィルタ、ラスタイメージフィルタはありません。
- Solaris 7 環境には、デバイスに依存しない **troff** があります。SunOS 4.x の **troff** 入力ファイルは、Solaris 7 の **troff** で使用できますが、**troff** のデフォ

ルト出力はプリンタではなく、標準出力に出力されます。したがって、`troff` 出力をプリンタへ送るときには、プリンタを指定しなければなりません

マニュアルページの構成の変更

マニュアルページの構成は、SVR4 構成と互換性を持つように変更されました。その結果、一部の節は名前が変更されています。たとえば、`man(8)` は `man(1M)` になっています。

表 6-4 は、SunOS 5.6 のマニュアルページのディレクトリを示します。

表 6-4 SunOS 5.6 マニュアルページディレクトリ

/man ディレ クトリ	内容	接尾辞
man1	ユーザコマンド	1B - SunOS/BSD 互換コマンド
		1C - 通信コマンド
		1F - FMLI コマンド
		1S - SunOS コマンド
man1M	システム管理コマンド	
man2	システムコール	
man3	ライブラリ関数	3B - SunOS/BSD 互換ライブラリ
		3C - C ライブラリ関数
		3E - ELF ライブラリ関数
		3G - C ライブラリ関数
		3I - ワイド文字関数
		3K - カーネル VM ライブラリ関数

表 6-4 SunOS 5.6 マニュアルページディレクトリ 続く

/man ディレ クトリ	内容	接尾辞
		3M - 数学関数
		3N - ネットワーク関数
		3R - RPC サービス関数
		3S - 標準入出力関数
		3T - スレッドライブラリ関数
		3X - その他のライブラリ関数
man4	ファイル形式	4B - SunOS/BSD 互換ファイル形式
man5	ヘッダ、テーブル、マクロ	
man7	特殊ファイル	
man9	DDI/DKI	
man9E	DDI/DKI エントリポイント	
man9F	DDI/DKI カーネル関数	
man9S	DDI/DKI データ構造	

man コマンド検索パスのカスタマイズ

SunOS 4.x ソフトウェアでは、個々の man ディレクトリを、あらかじめ決められた順序に従って検索していましたが、SunOS 5.6 ソフトウェアではユーザが検索パスを決めます。man コマンドは、man ページ構成ファイル man.cf に設定されたパスを使用します。

MANPATH 環境変数の各構成部分に、異なる man.cf ファイルを入れることができます。man.cf を変更して、検索の順序を変更することができます。たとえば、3c の前に 3b を検索するように指定できます。/usr/share/man ディレクトリ用の構成ファイルを次に示します。

```
#
# Default configuration file for the on-line manual pages.
#
MANSECTS=1,1m,1c,1f,1s,1b,2,3,3c,3s,3x,3i,3t,3r,3n,3m,3k,3g, \
3e,3b,9f,9s,9e,9,4,5,7,4b,6,1,n
```

MANSECTS に指定している引数は、利用できる man サブディレクトリの、man の次に付いている文字です。サブディレクトリのは、各サブセクションに独自のディレクトリがあるため、このリリースでは大幅に増えています。この新しい構造は、man コマンドの性能を向上させ、検索パスをより細かく制御します。次の 2 つの図は、2 つのリリースのマニュアルディレクトリを比較したものです。

```
sunos4.1% ls /usr/share/man
man1/ man2/ man3/ man4/ man5/ man6/ man7/ man8/
man1/ mann/
```

```
sunos5.6% ls /usr/share/man
man.cf man1f/ man3/ man3g/ man3n/ man3x/ man6/ man9f/
man1/ man1m/ man3b/ man3i/ man3r/ man4/ man7/ man9s/
man1b/ man1s/ man3c/ man3k/ man3s/ man4b/ man9/ man1/
man1c/ man2/ man3e/ man3m/ man3t/ man5/ man9e/ mann/
```

whatis データベースと windex データベース

SunOS 4.x では、マニュアルページの目次とキーワードのデータベースを `whatis` と呼んでいましたが、SunOS 5.6 ソフトウェアでは、この情報は `windex` ファイルにあります。どちらのリリースでもデータベースの作成には `catman` コマンドを使用し、データベースを利用するには `man`、`apropos` および `whatis` コマンドを使用します。

また次に示すように `windex` ファイルのフォーマットも `whatis` ファイルとは少し異なっています。

```
sunos4.1% man -k tset
tset, reset (1) - establish or restore terminal characteristics
```

```
sunos5.6% man -k tset
reset tset (1b) - establish or restore terminal characteristics
tset tset (1b) - establish or restore terminal characteristics
```

man コマンドの使用

表 6-5 は、SunOS 5.6 の man コマンドにさらに検索オプションがあることを示しています。

表 6-5 新しい man コマンドオプション

オプション	説明
-a	<i>file name</i> に一致するマニュアルページをすべて表示する。ページは見つかった順に続けて表示される。
-l	<i>file name</i> に一致するマニュアルページをすべて一覧表示する。このコマンドの出力で確認すれば、-s オプションでセクション番号を指定できる。
-s <i>section-number</i>	<i>file name</i> の <i>section-number</i> を検索する。SunOS 4.x ソフトウェアの man コマンドでは、オプションでセクション番号を指定することができたが、このリリースではセクション番号を -s に続けて指定しなければならない。
-F	<i>file name</i> で指定したファイルが見つかるまで man コマンドにすべてのディレクトリを検索させる。このオプションは windex データベースと man.cf ファイルの設定を無効にする。

SunOS 5.6 の man のコマンドの詳細については、man(1) のマニュアルページを参照してください。

デバイスの管理

この章では、SunOS 5.6 のデバイス命名規則について説明し、さらにディスクに関する情報の表示、システムへのデバイスの追加、ボリュームマネージャなど、デバイスに関連する作業の変更について説明します。

- 63ページの「デバイス命名規則」
- 65ページの「ディスク情報の表示」
- 67ページの「デバイスをシステムへ追加する」
- 68ページの「ボリュームマネージャの使用」

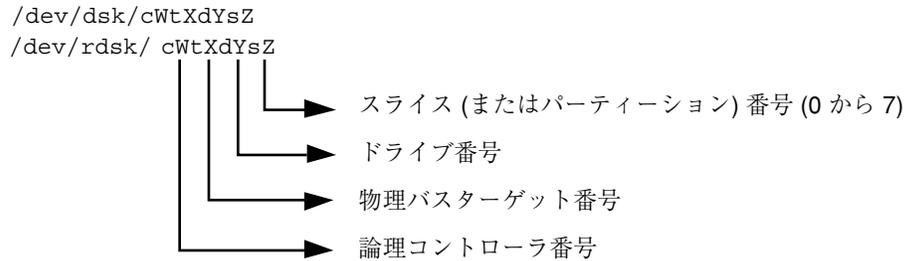
デバイス命名規則

SunOS 5.6 プラットフォームのデバイス命名規則は、SunOS 4.x から変更されました。さらに、特殊なデバイス名を含む `/dev` ディレクトリは、平坦なディレクトリからデバイスの種類ごとに個別のサブディレクトリを持つ階層型ディレクトリに変更されました。たとえば、ディスクデバイスファイルは `/dev/dsk` にあり、raw ディスクは `/dev/rdsk` にあります。

デバイス名を引数として使用する SunOS 5.6 コマンドは、SunOS 5.6 のデバイス命名規則に従う必要があります。ただし、SunOS/BSD ソース互換パッケージをインストールしている場合は、SunOS 4.x のデバイス名を使用し、認識することができます。詳細については、『*Source Compatibility Guide*』を参照してください。

ディスクに関する規則

ディスクパーティションスライス番号 (0 から 7) は、前の SunOS リリースのパーティションの a から h に相当します。



注 - ほとんどの SCSI ディスクにはコントローラが組み込まれています。これは、ドライブ番号は常に「0」で、ターゲット番号が変わることを意味します。たとえば、外部ディスクドライブの後部スイッチが「2」に設定されている場合、第 1 番目のスライスのデバイス名は `/dev/dsk/c0t0d2s0` ではなく `/dev/dsk/c0t2d0s0` です。

SCSI ターゲット 0 と 3 の名前は、一部の Sun4c システムでは予約されているため、デバイス名が混乱する可能性があります。SunOS 4.1.x ソフトウェアでは、SCSI ターゲット 3 は `sd0()` となっていました。現在では `c0t3d0` となっています。SCSI ターゲット 0 は `sd3()` となっていました。現在は `c0t0d0` です。他の SCSI ディスク名は、規則どおりに変換されます。たとえば、SunOS 5.6 ソフトウェアの `sd2a` は、`c0t2d0s0` となり、`sd2b` は `c0t2d0s1` となります。

テープドライブに関する規則

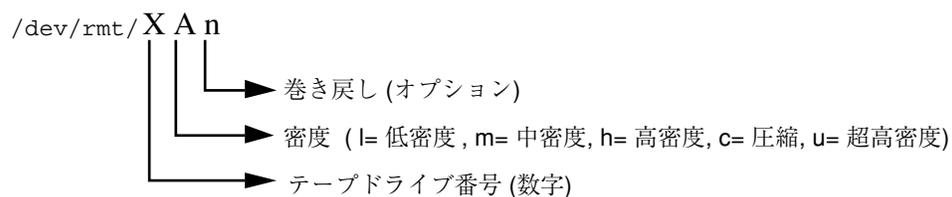


表 7-1 に、SunOS 4.x と SunOS 5.6 のデバイス命名規則を比較した例をいくつか示します。

表 7-1 SunOS 4.x と SunOS 5.6 のデバイス名

デバイスの種類	SunOS 4.x デバイス名	SunOS 5.6 デバイス名
ディスクデバイス	/dev/sd0g	/dev/dsk/c0t3d0s6
	/dev/rsd3b	/dev/rdsk/c0t0d0s1
	/dev/rsd3a	/dev/rdsk/c0t0d0s0
磁気テープデバイス	/dev/nrmt8	/dev/rmt/8hn
	/dev/rst0	/dev/rmt/0
CD-ROM デバイス	/dev/sr0	/dev/dsk/c0t6d0s2

ディスク情報の表示

SunOS 5.6 では、ディスク情報を表示するコマンドが変更されています。df (1M) と du (1M) はまだ使用できますが、変更されています。dinfo (8) と devinfo (1M)

は `prtvtoc` と `sysdef -d` に置き換えられました。この節ではこれらの変更について概要を説明します。

互換パッケージをインストールしている場合は、SunOS 4.x のコマンドが `/usr/ucb/df` と `/usr/ucb/du` の中にあります。

df コマンド

`df` コマンドは VFS アーキテクチャをサポートするために変更されました。他の VFS コマンドと同様に、これらのコマンドには汎用バージョンとファイルシステムバージョンがあります。SunOS 5.6 コマンドの構文は、SunOS 4.x で使用されていたものとは大幅に異なります (詳細は、付録 A を参照してください)。VFS の詳細については、83ページの「仮想ファイルシステムアーキテクチャ」を参照してください。

`df` コマンドはキロバイト単位ではなく、512 バイトのブロック単位でディスク空間の状況を表示しますが、`-k` オプションを使用するとキロバイト単位で表示することができます。また、`-t` オプションの機能が異なっています。以前このオプションは、指定した形式 (たとえば、“nfs” または “4.2”) のファイルシステムに出力を制限していました。SunOS 5.6 では、合計値を含めた完全なリストを表示します。

最後に、このコマンドに特殊デバイス名を指定するときは、SunOS 5.6 のデバイス命名規則を使用してください。詳細は、63ページの「デバイス命名規則」を参照してください。

du コマンド

`du` コマンドは、`df` コマンド同様に、キロバイト単位ではなく 512 バイトのブロック単位でディスクの使用状況を表示します。また、ディレクトリを読み込んだり、ファイルをオープンすることができない場合に、通常「何も出力しない」コマンドにメッセージを出力させる `-r` オプションもあります。

dkinfo コマンド

SunOS 4.x の `dkinfo` コマンドは削除されました。デバイス情報を印刷するには、`dkinfo` ではなく `prtvtoc(1M)` を使用します。

`prtvtoc` コマンドは、ディスクのパーミッション情報を含む、ディスクのラベルに記録されている重要な情報を表示します。`prtvtoc` の詳細については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

コード例 7-1 は、SunOS 5.6 prtvtoc コマンドの出力例を示します。

コード例 7-1 SunOS 5.6 prtvtoc コマンドの出力

```
# prtvtoc /dev/rdisk/c0t2d0s2
* /dev/rdisk/c0t2d0s2 partition map
*
* Dimensions:
*   512 bytes/sector
*   36 sectors/track
*   9 tracks/cylinder
*   324 sectors/cylinder
*   1272 cylinders
*   1254 accessible cylinders
*
* Flags:
*  1: unmountable
* 10: read-only
*
*
*           First   Sector Last
* Partition Tag  Flags Sector   Count  Sector Mount
Directory
  0      0  00      0     32724  32723  /
  1      0  00    32724    65448  98171
  2      0  00      0    406296  406295
  6      0  00    98172   308124  406295  /usr
```

devinfo コマンド

SunOS 4.x の devinfo は、SunOS 5.6 とは互換性がありません。SunOS 4.x に似た出力を作成するには、`-v` オプションを付けて `prtconf` を使用してください。

デバイスをシステムへ追加する

ブート時に、システムは接続されているすべてのデバイスのセルフテストとチェックを行います。システムに新しいデバイスを追加した後、`boot -r` を使用してカーネルの動的な再構成を行なってください。再構成スクリプトが実行されて、モジュールのディレクトリに登録されているすべてのデバイスドライバをロードし、対応するハードウェアノードを作成します。詳細は `kernel (1M)` のマニュアルページを参照してください。

また、`boot -a` を使用してシステムにドライバまたはモジュールを対話形式で追加することができますが、この場合は、何をブートするのか、ルートファイルシステムはどこにあるのかなど、他のブートパラメータを入力するように求められますので注意してください。

システムファイルとカーネルモジュールへのパスは、`/etc/system` に格納されています。システムをブートすると、`/etc/system` にある情報を読み出し、どのモジュールをロードするかを決定します。`system(4)` ファイルの `MODDIR` 構文、または `boot -a` を使用すれば、別のパスを指定できます。

`boot(1m)` またはデバイスとドライバの追加についての詳細は、『*Solaris のシステム管理 (第 1 巻)*』を参照してください。

動的再構成

Solaris 2.7 をインストールした特定の *SPARC* サーバで動的再構成を使用できます。この編成では、サービスプロバイダが実行中のシステムでホットプラグ対応のシステム I/O ボードを着脱できるので、リブートする必要はありません。また、交換したボードをただちに使用できない場合、システム管理者は、動的再構成でそのボードを無効にしたままシステムの動作を継続させることが可能です。

使用中のサーバが動的再構成をサポートしているかどうかについては、ハードウェア製造業者のマニュアルを参照してください。

ボリュームマネージャの使用

Solaris 2.2 ソフトウェア以降、ソフトウェアの新しいレイヤであるボリュームマネージャが *CD-ROM* やフロッピーディスク装置を管理しています。このソフトウェアは、ユーザと *CD* およびフロッピーディスク間のやりとりを自動化します。

OpenWindows と *CDE* のファイルマネージャは、ボリュームマネージャを使用できるように変更されており、ファイルシステムを持つ *CD* とフロッピーディスクにユーザがすばやくアクセスできるように、ファイルマネージャが変更されています。ファイルマネージャの新しい機能についての詳細は、『*OpenWindows ユーザーズガイド*』を参照してください。

また、システム上でボリュームマネージャの管理に役立つ新しいコマンドもいくつか追加されています。

ボリュームマネージャは、取り外し可能な媒体がデバイスに挿入されると、*CD* とフロッピーディスクのファイルシステムを自動的にマウントします。*CD* またはフロッピーディスクにファイルシステムがある場合は、表 7-2 に示す位置に自動的にマウントされます。

表 7-2 ファイルシステムをもつ CD-ROM とフロッピーディスクの位置

媒体	位置
CD	/cdrom/cdrom_name
フロッピーディスク	/floppy/floppy_name

CD またはフロッピーディスクにファイルシステムがない場合は、表 7-3 に示す位置でアクセスできます。

表 7-3 ファイルシステムのない CD-ROM とフロッピーディスクの位置

媒体	位置
CD	/vol/dev/aliases/cdrom0
フロッピーディスク	/vol/dev/aliases/floppy0

セキュリティ上の理由から、これらのファイルシステムは `setuid` を有効にしてマウントされます。このマウントオプションとその他のマウントオプションについての詳細は、`mount (1M)` のマニュアルページを参照してください。

ボリュームマネージャの構成およびフロッピーディスクと CD-ROM の使用方法についての詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

ボリュームマネージャの各コマンドのマニュアルページも用意されています。以下のマニュアルページを参照してください。

`rmmount (1)`、`rmmount.conf (4)`、`volcancel (1)`
`volcheck (1)`、`vold (1M)`、`volmgt (3)`
`vold.conf (4)`、`volfs (7)`、`volmissing (1)`

注 - 現在、ボリュームマネージャは、次の CD-ROM パスを制御します。

/dev/dsk/c0t6d0s0

/dev/rdisk/c0t6d0s0

次のフロッピーディスクパスを制御します。

/dev/diskette

/dev/rdiskette

これらのパスを使用して CD やフロッピーディスクをマウントまたはアクセスしようとする、エラーメッセージが表示されます。

表 7-4 に示すように、ボリュームマネージャの管理用に新しいコマンドが追加されています。

表 7-4 ボリュームマネージャのコマンド

コマンド	説明
rmmount(1)	取り外し可能な媒体のマウンタ。CD やフロッピーディスクがインストールされたときに vold によって使用され、/cdrom および /floppy を自動的にマウントする。
volcancel(1)	特定の CD やフロッピーディスクにアクセスするユーザの要求をキャンセルする。
volcheck(1)	インストールされたについてドライブをチェックする。デフォルトでは、/dev/diskette によって示されたドライブをチェックする。
volmissing(1)	ドライブにない CD やフロッピーディスクにアクセスを試みると、ユーザにそのことを通知する。
vold(1)	/etc/vold.conf によって制御されるボリュームマネージャのデーモン

また、ボリュームマネージャの動作を定義するため、2つの構成ファイル、/etc/vold.conf と /etc/rmmount.conf があります。これらのファイルの説明については rmmount.conf(4) と rmmount.conf(4) のマニュアルページを参照してください。また、CD-ROM とフロッピーディスクの管理については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

起動とシャットダウン

この章では、システムのブートとシャットダウン手順の変更について説明します。

- 71ページの「ブート」
- 74ページの「init コマンドの使用」
- 76ページの「シャットダウン」

ブート手順の詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。それぞれのコマンドのマニュアルページは『SunOS Reference Manual』のセクション 1M「Maintenance Commands」にオンラインで提供されています。

ブート

Solaris 7 のブートプロセスによってシステム管理が容易になりました。主な変更点は次のとおりです。

- カーネルは自動的に構成されるため、手作業で再構成する必要はありません。
- カーネルのメモリ消費量は、デバイスが最初にオープンされるときに自動的にロードすることによって削減されています。
- ファイルシステムは必要なときにだけチェックされ、ブートアップ時間が短縮されています。
- ブートブロックは UNIX ファイルシステムを読み込むことができ、ブートプログラムを移動したときのブートエラーをなくします。
- サードパーティの起動可能デバイスがサポートされています。

- 二次ブートプログラム、`ufsboot` と `inetboot` は CacheFS ファイルシステムを読み取るように修正されました。この新しいブート機能により、Solstice AutoClient™ システムはこれまでよりも速くブートでき、ネットワークリソースへの影響も少なくなります。
- SunOS 4.x の `fastboot` コマンドは、SunOS/BSD ソース互換パッケージがインストールされた Solaris 2.6 システムだけで使用できます。`fastboot` コマンドは、Solaris 7 システムでは使用されません。これは、ファイルシステムのチェックを行うのが、ファイルシステムの状態がクリーンでないと認識された場合だけに限られるからです。
- SunOS 4.x の `halt` と `reboot` コマンドは、SunOS 5.6 では `shutdown(1M)` と `init(1M)` に相当します。

Solaris 7 の環境では、システムを停止、シャットダウン、リブートするには `shutdown` と `init` コマンドが望ましい方法です。`reboot` コマンドは Solaris 7 環境で使用できますが、通常のシャットダウンサービスなしで、システムをすぐに停止します。表 8-1 に、SunOS 4.x から変更された SunOS 5.6 のコマンドを示します。

表 8-1 SunOS 5.6 でのリブート用コマンドの変更 `fastboot`

SunOS 4.x コマンド	SunOS 5.6 変更コマンド
<code>reboot</code>	<code>shutdown -i -6, init 6</code>
<code>fastboot</code>	<code>boot, init 6</code>

boot コマンドの変更

SunOS 5.6 ソフトウェアには次のような `boot` コマンドのオプションが追加されています。

- 新しいハードウェアを追加したり、ハードウェアの位置を変更するときには、`boot -r` を入力してください。このオプションは、物理デバイス名と論理デバイス名を作成します。論理デバイス名は物理デバイス名にリンクされます。
- すべてのシステムブートメッセージを見たいときは、`boot -v` と入力してください。デフォルトはメッセージを表示しないでブートします。メッセージは、常に `/var/adm/messages` ファイルに格納されます。

- ブートするカーネル、`/etc/system` ファイルやカーネルモジュールディレクトリ名を指定するプロンプトを表示させたいときは、`boot -a` と入力してください。

PROM からのブート

PROM からブートするときは、次の変更点に注意してください。

- PROM はディスクから `bootblk` をロードします。このファイルは、UFS ファイルシステムに固有のものであるということ以外は、以前の SunOS 4.x のブートブロックに似ています。

SunOS 4.x と同様に、`installboot (1M)` を使用してブートに使用されるパーティションにブートブロックをインストールする必要があります。

- `bootblk` はブートデバイスをオープンし、指定したファイルシステムを使用して `ufsboot` を検索しロードします。
- `ufsboot` がメモリにロードされた後で、ブート PROM はカーネル `/kernel/genunix` をロードします。SunOS 4.x システムは `/vmunix` を使いました。ただし、SunOS 5.6 では、`/kernel` ディレクトリにはシステムをブートするのに必要なプラットフォーム非依存カーネルモジュールがすべて (`unix` も含む) 入っています。
- カーネルは、`/kernel/drv` ディレクトリから `esp` など、他のドライバをロードします。これらのドライバは、SunOS 4.x カーネルの一部として構築しなければなりません。しかし、SunOS 5.6 のシステムでは、これらのドライバが必要なときに、動的にロードすることができます。
- `/sbin/init` コマンドは、`/etc/inittab` の指示に基づいてシステムを設定するためのプロセスを生成します。次の節では、`init` が使用する実行レベルについて説明します。

ブート処理の相違

表 8-2 にブート時の処理の相違をまとめます。

表 8-2 ブート時の相違

SunOS 4.x	SunOS 5.6	機能
bootsd	bootblk	ディスクから ufsboot をロードする。
boot program	ufsboot	ディスクから unix をロードする。
/vmunix	/kernel/genunix	ブート可能なカーネルイメージ
boot.sun4c.sunos.4.1	inetboot	ネットワークから unix をマウントおよびコピーする。
rc.boot rc.single	/etc/rcS	/usr をマウントし、ファイルシステムをチェックする。
rc.local	/etc/rc2 /etc/rc3	システム構成スクリプト
/etc/config	modload /etc/system	必要に応じてシステムカーネルとロードモジュールをカスタマイズする。
PROM モニタ、シングルユーザ、マルチユーザ	実行レベル 0-6 と S	システム実行レベル
/dev/sd1g	/dev/dsk/c0t1d0s6	よりわかりやすい論理デバイス名。63ページの「デバイス命名規則」を参照。
MAKEDEV	boot -r, add_drv	デバイスノードを作成する。

init コマンドの使用

SunOS 4.x の fasthalt コマンドは、SunOS 5.6 では init (1M) コマンドに変更されています。init (1M) コマンドは、シングルユーザシステムをシャットダウンするのに使用してください。init を使用して、システムをパワーダウン状態 (init 0)、またはシングルユーザ状態 (init 1) にすることができます。

init コマンドの変更

次の init コマンドの変更点に注意してください。

- SunOS 5.6 システムソフトウェアには、8つの初期設定状態 (init 状態または実行レベル) があります。デフォルトの init 状態は、`/etc/inittab` ファイルに定義されています。
- SunOS 5.6 の init コマンドは、すべての実行レベルを `/etc/rc`、`/etc/rc.boot` および `/etc/rc.local` ファイルにまとめて入れるのではなく、それぞれの実行レベルに対して異なるスクリプトを使用します。実行レベルごとに名前が付けられているこれらのファイルは、`/sbin` ディレクトリにあります。

『Solaris のシステム管理 (第 1 巻)』では、このコマンドについて詳しく説明しています。

システム実行レベルの変更

SunOS 5.6 の init コマンドにより、システムの実行レベル (初期設定状態) を制御し、各種の動作モードを容易に切り替えることができます。SunOS 5.6 はすべてのシステム状態を1つのファイルに書き込むのではなく、`/sbin/rc` スクリプトを使用してそれぞれの実行レベルを制御します。これにより、新しいスクリプトを作成したり、既存のものを修正する場合、それぞれのファイルを変更することができます。SunOS 4.x システムでは、`/etc/rc`、`/etc/rc.boot`、`/etc/rc.local` ファイルを使用して実行レベルを制御していました。

SunOS 4.x には、`prom` モニタ、シングルユーザ、マルチユーザの3つの実行レベルがありました。これらは、SunOS 5.6 の実行レベル 0、1、3 に相当します。

表 8-3 に各実行レベルの `/sbin/rc` スクリプトの動作の概要を示します。

表 8-3 SunOS 5.6 初期設定実行レベル

実行レベル	デフォルトの SunOS 5.6 の機能
0	電源を切っても安全なように、システムをシャットダウンする。システムサービスとデーモンを停止させる。実行中のプロセスをすべて終了させる。すべてのファイルシステムのマウントを解除する。
1	システム上に 1 ユーザしか許さないシングルユーザ (システム管理者) 状態。システムサービスとデーモンを停止させる。実行中のプロセスをすべて終了させる。すべてのファイルシステムのマウントを解除する。
2	NFS システムをエクスポートしない通常のマルチユーザ状態。timezone 変数を設定する。/usr ファイルシステムをマウントする。/tmp と /var/tmp ディレクトリ内を削除する。ネットワークインタフェースをロードしプロセスを起動する。cron デーモンを起動する。uucp tmp ファイルをクリーンアップする。lp システムを起動する。sendmail デーモンを起動する。
3	NFS システムを共有するファイルサーバの通常のマルチユーザ状態。実行レベル 2 における作業をすべて実行する。NFS システムデーモンを起動する。
4	代替マルチユーザ状態 (未使用)。
5	電源を切っても安全なように、システムをシャットダウンする。この機能をサポートしているシステムでは自動的にシステムの電源を切る。
S,s	シングルユーザ状態。ファイルシステムを一部マウントし、アクセス可能にする。

シャットダウン

複数のユーザがいるシステムをシャットダウンするときに、shutdown(1M) コマンドを使用します。shutdown(1M) コマンドは、ログインしているユーザすべてに警告を送り、60 秒後にシステムをシングルユーザ状態にシャットダウンします。

- SunOS 4.x の fasthalt コマンドは、SunOS/BSD ソース互換パッケージがインストールされている SunOS 5.6 システムでのみ使用できます。
- SunOS 4.x の halt と reboot コマンドは、shutdown と init に相当します。

シャットダウン手順についての詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

SunOS 5.6 では、shutdown コマンドはシステムを停止またはシャットダウンするのに推奨できる方法です。shutdown と init は rc スクリプトを使用して、実行中のプロセスを終了します。halt コマンドは、SunOS 5.6 で使用できますが、サービスを通常の手順でシャットダウンするのではなく、すぐにシステムを停止します。表 8-4 に、SunOS 4.x から変更された SunOS 5.6 のコマンドを示します。

表 8-4 SunOS 5.6 のシャットダウンの変更 halt、fasthalt

SunOS 4.x コマンド	SunOS 5.6 変更コマンド
halt	shutdown -i 0, init 0
fasthalt	shutdown -i 0, init 0

shutdown コマンドと init コマンドは、シャットダウンシーケンスを制御する数値の「実行レベル」引数を使用できます。実行レベルの番号についての詳細は shutdown(1M) と init(1M) のマニュアルページを参照してください。

shutdown コマンドの変更

SunOS 5.6 の shutdown コマンドは、表 8-5 にあるオプションしか使用できません。このコマンドとそのオプションは、『Solaris のシステム管理 (第 1 巻)』で説明しています。

表 8-5 SunOS 5.6 の shutdown コマンドオプション

オプション	機能
-g	shutdown が始まるまでの「猶予」期間を選択する。
-i [init state]	初期実行レベルを指定する。(表 8-3を参照)。

表 8-5 SunOS 5.6 の shutdown コマンドオプション 続く

オプション	機能
-y	確認の質問をせずに shutdown を実行する。 すべての質問に対し「yes」の応答を仮定する。
-message	ユーザサポートのメッセージを指定する。語が複数の場合は、引用符でメッセージを囲む。

デフォルトでは、SunOS 5.6 の shutdown コマンドは、実際の shutdown が始まる前に確認を要求しますが、オペレータの介入なしに実行できるオプション `-y` もあります。

shutdown オプションは、Solaris 7 システムの BSD ソース互換モードにおいてのみ使用できます。

変更の一覧については、付録 A を参照してください。コマンドの機能に関する情報は、`shutdown(1M)` のマニュアルページを参照してください。

fasthalt コマンドと fastboot コマンドの使用

SunOS 4.x の `fastboot` コマンドと `fasthalt` コマンドは、Solaris 7 システムで SunOS/BSD ソース互換パッケージを実行している場合に使用できます。これらのコマンドのファイルシステムチェック機能は、Solaris 7 には対応していません。

halt コマンドと reboot コマンドの使用

`halt` コマンドと `reboot` コマンドは、`/sbin` にある `rc` スクリプトを実行しないため、お薦めできません。SunOS 5.6 システムの `halt` コマンドと `reboot` コマンドは、他の AT&T SVR4 システムにはないため、`shutdown` と `init` に `halt` と `reboot` コマンドに相当する機能があります。

ファイルシステムの管理

この章では、ファイルシステムのレイアウトの変更、ファイルシステムの変更、また、仮想ファイルシステム、ディレクトリ、ファイルの変更点について説明します。次のファイルシステム管理の変更点についても説明しています。

- ファイルシステムのマウント
- ファイルシステムの監視
- ファイルシステムの共有
- 新しいファイルシステムの作成
- ファイルシステムのチェック
- ファイルのバックアップと復元

この章は、次の節で構成されています。

- 80ページの「ファイルシステムの変更」
- 81ページの「デフォルトのファイルシステムとディレクトリ」
- 83ページの「仮想ファイルシステムアーキテクチャ」
- 89ページの「ディレクトリとファイルの変更」
- 98ページの「ファイルシステム管理コマンドの使用」

ファイルシステムの理解と管理についての詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

ファイルシステムの変更

SunOS 5.6 と SunOS 4.x のファイルシステムは類似しています。しかし、システムディレクトリとファイルの位置と名前は変更されています。また、新しいファイルシステムと新しい疑似ファイルシステムがあり、ディレクトリの1つが削除されました。

ファイルシステムの位置と名前についての変更は次のとおりです。

- /dev ディレクトリは、平坦な構造から階層構造に変更されました。
- /etc ディレクトリには、システム構成情報が含まれるようになりました。いくつかのファイルとサブディレクトリの追加、削除、変更が行われています。
- /etc/vfstab ファイルは、/etc/fstab に変更されました。
- /etc/printcap ファイルは、/etc/lp ディレクトリに変更されました。
- SunOS 5.6 の /sbin ディレクトリには、システムの実行レベルを変更する rc スクリプトと、ファイルシステムをマウントする前にシステムを初期設定する rcs スクリプトが含まれています。
- SunOS 5.6 の /usr ディレクトリには、システムの提供する共有可能ファイルと実行可能ファイルが含まれています。
- /var ディレクトリには、通常の使用時に大きさが変化するファイルが含まれています。いくつかのファイルと /var サブディレクトリの追加、削除、変更が行われています。
- /var/spool/mail ディレクトリは、/var/mail に変更されました。
- カーネルが動的にロードされるため、/sys ディレクトリは使用されなくなりました。
- /RFS ファイルシステムは削除されました。
- termcap データベースは、terminfo に変更されました。

疑似ファイルシステム

TFS 疑似ファイルシステムは SunOS 5.6 には含まれていません。

追加された疑似ファイルシステムは次のとおりです。

- CACHEFS 疑似ファイルシステムは、CD-ROM のような低速装置の性能を改善するために使用します。

- PROCFS 疑似ファイルシステムはメモリに常駐し、アクティブなプロセスのプロセス番号を /proc ディレクトリに保持しています。proc(4) のマニュアルページを参照してください。
- FDFS 疑似ファイルシステムは明示的なファイル名を提供し、ファイル記述子を使用してファイルをオープンします。
- FIFOFS 疑似ファイルシステムはパイプファイルを格納して、複数のプロセスが共通のデータにアクセスできるようにします。
- NAMEFS 疑似ファイルシステムは、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のが目的で、ほとんどの場合 STREAMS が使用します。
- SWAPFS 疑似ファイルシステムはデフォルトのスワップデバイスで、システムのブート時やシステム管理者が他のスワップ空間を作成するときに使用します。

ファイルシステムの追加

次のファイルシステムは SunOS 5.6 のディレクトリ構造に含まれています。

- unix と呼ばれるカーネルとカーネルモジュールが、/kernel ディレクトリに格納されています。
- オプションの /opt ファイルシステムは、サードパーティのソフトウェアや、アンバンドルのソフトウェアを格納するために使用します。/opt が単独のファイルシステムではない場合は、/usr/opt へのシンボリックリンクになっています。
- ファイルシステムは、ボリューム管理デーモン vold(1M) が使用するデフォルトのファイルシステムを提供します。volfs(7) のマニュアルページを参照してください。

デフォルトのファイルシステムとディレクトリ

SunOS 5.6 のファイルシステムは階層型です。図 9-1 は、SunOS 5.6 のデフォルトディレクトリとファイルシステム (点線で示されている) を示したものです。ここでは、ディレクトリ、またはファイルシステムが実際に持つサブディレクトリの例を示しています。表 9-1 に簡単な説明があります。

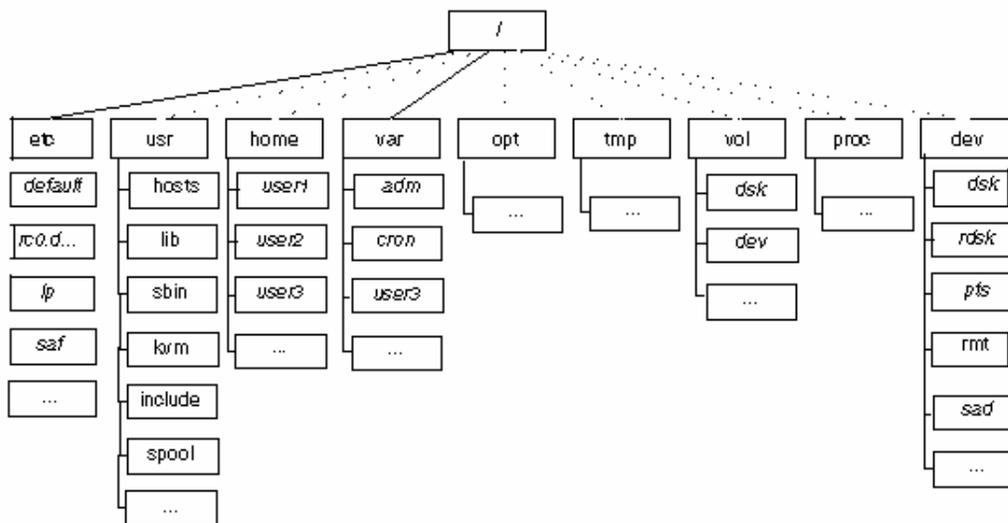


図 9-1 Solaris 7 のデフォルトファイルシステムとディレクトリ階層

Solaris 7 ソフトウェアには、ファイルシステムとディレクトリのデフォルトセットが含まれていて、一定の規則に従って類似した種類のファイルをグループにまとめています。表 9-1 で、デフォルトファイルシステムとディレクトリについて要約します。

表 9-1 Solaris 7 ファイルシステムとディレクトリ

ファイルシステム またはディレクトリ	形式	説明
/	ファイルシステム	階層型ファイルツリーの最上部。ルートディレクトリには、カーネル (/kernel/unix)、デバイスドライバとシステムのブートに使用されるプログラムなど、システムの動作に重要なディレクトリやファイルが格納される。また、ローカルファイルシステムとリモートファイルシステムをファイルツリーにリンクするマウントポイントディレクトリも格納される。
/etc	ディレクトリ	システム管理で使用されるシステム固有のファイルが格納される。
/usr	ファイルシステム	アーキテクチャ依存型と非依存型の共有可能ファイルが格納される。すべてのシステムタイプで使用可能なマニュアルページなどのファイルは /usr/share に格納される。

表 9-1 Solaris 7 ファイルシステムとディレクトリ 続く

ファイルシステム またはディレクトリ	形式	説明
/home	ファイルシステム	ユーザのホームディレクトリのマウントポイントで、ユーザのワークファイルが格納される。デフォルトでは、/home は現在自動マウントされる。
/var	ディレクトリ	ローカルシステムがある限り、変化したり増大したりする可能性があるシステムファイルやディレクトリが格納される。たとえば、システムログ、vi と ex のバックアップファイル、uucp ファイルなど。
/opt	ファイルシステム	オプションのサードパーティ製ソフトウェアのマウントポイント。システムによっては、/opt がローカルディスクパーティションの UFS ファイルシステムになる場合がある。
/tmp	ファイルシステム	システムがブートされるか /tmp のマウントが解除されるたびに消去される一時ファイル。
/vol	ファイルシステム	取り外し可能な媒体用のディレクトリがあり、vol1(1M) によって管理される。
/proc	ファイルシステム	アクティブなシステムプロセスの番号順のリストがある。このファイルシステムはディスク領域を使用しない。
/sbin	ディレクトリ	ブートプロセスと手作業のシステム回復で使用される重要な実行可能ファイル。

仮想ファイルシステムアーキテクチャ

SunOS 5.6 の機能である仮想ファイルシステム (VFS) アーキテクチャは、複数のファイルシステムをサポートするファイルシステム管理を簡略化します。

長年にわたって、それぞれ独自のファイルシステム管理用コマンドセットをもつ UNIX ファイルシステムが数種類開発されてきました。すべての種類を学習するのは混乱を招くものになり、また困難です。SunOS 5.6 は、ファイルシステム管理用

の汎用のコマンドセットでこの問題に対処しています。これらのコマンドは、管理に関するファイルシステム間の違いを意識させない共通の VFS インタフェースの一部です。以下の項では、サポートされるファイルシステムと汎用のファイルシステムコマンドの概要を示します。

サポートされるファイルシステムの形式

SunOS 4.x に含まれていたほとんどのファイルシステム形式は、SunOS 5.6 ソフトウェアにも含まれています。ただし、例外が1つあり、半透過ファイルシステム (TFS) 形式は、SunOS 5.6 では廃止されました。表 9-2 に、SunOS 4.x と SunOS 5.6 環境で使用できるファイルシステム形式をまとめます。

表 9-2 ファイルシステム形式の要約

カテゴリ	ファイルシステム名	説明	SunOS 4.x	SunOS 5.6
ディスクベース	UFS	UNIX ファイルシステム	あり	あり
	HSFS	CD-ROM ファイルシステム	あり	あり
	PCFS	PC ファイルシステム	あり	あり
ネットワークベース	NFS	Sun の分散ファイルシステム	あり	あり
疑似	SPECFS	特殊デバイスファイルシステム	あり	あり
	TMPFS	/tmp 一時ファイルシステム	あり	あり
	LOFS	ループバックファイルシステム	あり	あり
	TFS	半透過ファイルシステム	あり	なし
	PROCFS	プロセスアクセスファイルシステム	なし	あり
	FDFS	ファイル記述子ファイルシステム	なし	あり
	FIFOFS	FIFO/ パイプファイルシステム	なし	あり

表 9-2 ファイルシステム形式の要約 続く

カテゴリ	ファイルシステム名	説明	SunOS 4.x	SunOS 5.6
	NAMEFS	ネームファイルシステム	なし	あり
	SWAPFS	スワップファイルシステム	なし	あり
	CACHEFS	キャッシュファイルシステム	なし	あり

ファイルシステムについての詳細は、proc(4) と fd(4) のマニュアルページと『Solaris のシステム管理 (第 1 巻)』を参照してください。

キャッシュファイルシステム (CACHEFS)

キャッシュファイルシステムはリモートファイルシステム、または CD-ROM などの低速装置の性能を改善するのに使用されます。ファイルシステムがキャッシュされていると、リモートファイルシステムや CD-ROM から読み込まれたデータは、ローカルシステムのキャッシュに格納されます。

スワップファイルの変更

SWAPFS は、SunOS 5.6 では、システムをブートするとき、またはスワップ空間を追加するときのデフォルトのスワップデバイスです。このスワップデバイスはスワップ空間として物理メモリを使用しますが、ディスクにも物理スワップ空間が必要です。

SunOS 4.x システムでは、デフォルトの物理スワップデバイスはシステム構成に依存します。スタンドアロンシステムのデフォルト sd0b ディスクレスシステムは、スワップファイルを bootparam サーバから取得します。SunOS リリース 5.6 ソフトウェアではスワップファイルを、ディスク上のファイルを指定せず、デフォルトダンプデバイスとして使用します。

サポートされない SVR4 ファイルシステムの形式

表 9-3 に SunOS 5.6 ではサポートされない SVR4 ファイルシステム形式を示します。

表 9-3 サポートされない SVR4 ファイルシステム形式

ファイルシステム名	説明
BFS	ブートファイルシステム
S5	System V ファイルシステム
xnamefs	XENIX セマフォファイルシステム

汎用ファイルシステムコマンド

ほとんどのファイルシステム管理コマンドは、汎用コンポーネントとファイルシステムコンポーネントを持っています。ファイルシステムコンポーネントを呼び出す汎用コマンドを使用してください。表 9-4 は、`/usr/bin` ディレクトリにある汎用ファイルシステム管理コマンドの一覧です。

表 9-4 汎用ファイルシステム管理コマンド

コマンド	説明
<code>clri (1M)</code>	i ノードをクリアする。
<code>df (1M)</code>	空きディスクブロック数とファイル数を表示する。
<code>ff (1M)</code>	ファイルシステムに含まれるファイル名と統計情報を一覧表示する。
<code>fsck (1M)</code>	ファイルシステムの整合性を検査し損傷が見つければ修理する。
<code>fsdb (1M)</code>	ファイルシステムデバッガ
<code>fstyp (1M)</code>	ファイルシステムの形式を判定する。

表 9-4 汎用ファイルシステム管理コマンド 続く

コマンド	説明
labelit (1M)	テープヘコピーするときにファイルシステムのラベルを表示または提供する (volcopy コマンドでのみ使用)。
mkfs (1M)	新しいファイルシステムを作成する。
mount (1M)	ファイルシステムとリモートリソースをマウントする。
mountall (1M)	ファイルシステムテーブルで指定されたすべてのファイルシステムをマウントする。
ncheck (1M)	i 番号をもつパス名のリストを作成する。
umount (1M)	ファイルシステムとリモートリソースのマウントを解除する。
umountall (1M)	ファイルシステムテーブルで指定されたすべてのファイルシステムのマウントを解除する。
volcopy (1M)	ファイルシステムのイメージコピーを作成する。

また、これらのほとんどのコマンドには、ファイルシステムに対応するものがあります。



注意 - ファイルシステムコマンドを直接使用しないでください。このディレクトリをサポートしないファイルシステムで動作させると、汎用コマンドが次のエラーメッセージを表示します。

`[command: Operation not applicable for FSType type]`

汎用コマンドの構文

ほとんどの汎用コマンドは、次の構文を使用します。

`command [-F type] [-V] [generic-options] [-o specific-options] [special | mount-point] [operands]`

汎用コマンドのオプションと引数は次のとおりです。

`-F type`

ファイルシステムの形式を指定します。このオプションを使用しないと、このコマンドは `/etc/vfstab` ファイルの `special` または `mount point` と一致するエントリをさがします。それ以外の場合は、ローカルファイルシステムに対しては `/etc/vfstab` ファイル、リモートファイルシステムに対しては `/etc/dfs/fstypes` ファイルからデフォルトが取り出されます。

-V

完了したコマンド行を表示します。表示された行には、`/etc/vfstab` から入手した追加情報を含めることができます。このオプションを使用して、コマンド行の検査と妥当性検査を行います。このコマンドは実行されません。

generic-options

異なるファイルシステムの形式に共通のオプション。

-o *specific-options*

ファイルシステムの形式に固有のオプションのリスト。このフォーマットは、`-o` の後にスペースを入れ、さらに `keyword [=value]` のペアを、スペースではなくカンマで区切って続けます。

special | mount-point

ファイルシステムを指定します。この名前は、マウントポイントか、またはファイルシステムを持つスライス用の特殊デバイスファイルです。一部のコマンドでは、`special` ファイルは `raw` (キャラクタ型) デバイスでなければならないが、また、ブロック型デバイスでなければならないものもあります。場合によっては、この引数は他の情報を入力する `/etc/vfstab` ファイルの中で一致するエントリを検索するためのキーとして使用されます。ほとんどの場合、この引数は必須であり、*specific-options* の直後になければなりません。ただし、`/etc/vfstab` ファイルに一覧表示されたすべてのファイルシステム (オプションにより形式によって制限できる) でコマンドを動作させたいときには、この引数は必要ありません。

operands

ファイルシステムの形式に固有の引数。詳細については、コマンド (たとえば、`mkfs_ufs(4)`) の該当するマニュアルページを参照してください。

システム全体のデフォルトのファイルシステム形式

デフォルトのリモートファイルシステム形式は、`/etc/dfs/fstype` です。デフォルトのローカルファイルシステム形式は、`/etc/default/fs` です。詳細については、`default_fs(4)` のマニュアルページを参照してください。

コマンドの位置

以前の SunOS では、すべてのファイルシステムコマンドは、`/etc` ディレクトリにありました。SunOS 5.6 では、ファイルシステムコマンドは使いやすいように、別々の階層に編成されています。ファイルシステムコマンドは、`/usr/lib/fs/fstype` に格納されます。`/usr` がマウントされる前に必要なコマンドは、`/etc/fs/fstype` に複製されます。

汎用コマンドはすべて `/usr/sbin` にあります。`/usr` がマウントされる前に必要なコマンドは、`/sbin` に複製されます。

表 9-5 は、ファイルシステムコマンドの格納位置を示します。

表 9-5 ファイルシステムコマンドの格納位置

形式	プライマリバージョンの位置	複製バージョンの位置 (root)
汎用	<code>/usr/sbin</code>	<code>/sbin</code>
固有	<code>/usr/lib/fs</code>	<code>/etc/fs</code>

新しい UFS マウントオプション

ファイルのアクセス時刻を更新しない場合、UFS ファイルシステムをマウントするときに `-o noatime` オプションを指定できます。このオプションを指定すると、Usenet ニューススプールのようにアクセス時刻が重要でないファイルシステムのディスク使用率が削減されます。

ディレクトリとファイルの変更

この節では、SunOS 4.x と SunOS リリース 5.6 間の、ディレクトリとファイルに対する変更について説明します。

/dev ディレクトリ

/dev ディレクトリは、平坦なディレクトリから階層構造になりました。表 9-6 で追加されたサブディレクトリについて説明します。

表 9-6 /dev ディレクトリへの追加

サブディレクトリ	説明
/dev/dsk	ブロックディスクデバイス用
/dev/rdsk	raw ディスクデバイス用
/dev/pts	疑似端末 (pty) スレーブデバイス用
/dev/rmt	raw テープデバイス用
/dev/sad	STREAMS 管理ドライバのエントリポイント用
/dev/term	端末デバイス用

/etc ディレクトリ

/etc ディレクトリには、システム固有の構成情報が格納されます。ファイルやサブディレクトリの中には、削除または変更されたものがいくつかあり、また新たに追加されたものもあります。

- `mount*` のようなファイルシステムのコマンドは `/usr/lib/fs` ディレクトリのサブディレクトリへ移動
- SunOS 4.x `/etc/fstab` ファイルは `/etc/vfstab` に変更

`rc`、`rc.boot`、`rc.local`、`rc.single` といった初期設定用のスクリプトは SunOS 5.6 で利用できません。これらは表 9-7 に示すスクリプトに変更され、対応する実行制御ファイルによって実行されます。表 9-8 に、SunOS 5.6 の `/etc` ディレクトリに追加されたサブディレクトリを説明します。

表 9-7 初期設定用スクリプトと対応する実行制御ファイル

スクリプト	実行制御ファイル
/etc/rc0.d	/sbin/rc0
/etc/rc1.d	/sbin/rc1
/etc/rc2.d	/sbin/rc2
/etc/rc3.d	/sbin/rc3
/etc/rc4.d	/sbin/rc4
/etc/rc5.d	/sbin/rc5
/etc/rc6.d	/sbin/rc6
/etc/rcS.d	/sbin/rcS

表 9-8 /etc ディレクトリへの追加

サブディレクトリ	説明
/etc/default	デフォルトのシステム構成を定義
/etc/inet	インターネットサービスの構成を定義
/etc/lp	LP システムの構成を定義
/etc/opt	インストールされたオプションソフトウェアを定義
/etc/rcn.d	実行状態の変更操作を定義
/etc/saf	サービスアクセス機能 (SAF) の構成を定義

/etc/vfstab ファイル

SunOS 5.6 ソフトウェアでは、以前の /etc/vfstab ファイルが仮想ファイルシステムのファイル /etc/vfstab に変更されます。仮想ファイルシステムのアーキテクチャにおいて /etc/vfstab ファイルが提供するものは、ファイルシステム管理を行うための汎用コマンドが使用する、デフォルトのファイルシステムパラメータです。これらのコマンドに関する説明は、86ページの「汎用ファイルシステムコマンド」を参照してください。

名前が変更されたことのほかに、/etc/vfstab ファイルと /etc/vfstab ファイルには下記の相違点があります。

- device to fsck フィールドの追加。fsck によりチェックされる raw デバイスの名前を指定します。
- automount フィールドの追加。ファイルシステムを常にマウントするよう制御するため mountall が使用します (ただし、自動マウントデーモンはこのフィールドを使用しない)。
- freq フィールドの削除。以前はこのフィールドにダンプの間隔を日数で指定していました。

ファイルシステムテーブルには7つのフィールドがあり、タブで区切られています。表 9-9 にフィールドエントリを示します。

注 - /etc/vfstab ファイルのフィールドにはそれぞれエントリが必要です。
フィールドに設定する値がない場合は、ダッシュ (-) を入れてください。

表 9-9 /etc/vfstab ファイルのフィールド名とその内容

フィールド名	内容
device to mount	<p>このフィールドにあるエントリは次のとおり</p> <p>ローカルな UFS ファイルシステム用のブロック型特殊デバイス (たとえば /dev/dsk/c0t0d0s0 など)</p> <p>リモートファイルシステムのリソース名 (たとえば NFS ファイルシステム用の myserver:/export/home など)</p> <p>スワップ先のスライス名 (たとえば /dev/dsk/c0t3d0s1 など)</p> <p>/proc ディレクトリと proc ファイルシステム形式</p> <p>ファイルシステム形式が hsf s の CD-ROM</p> <p>/dev/diskette ファイルシステム形式が pcfs または UFS の /dev/diskette</p> <p>このフィールドはスワップファイルシステムを指定する場合にも使用。リモートファイルシステムの詳細については『NFS の管理』を参照のこと。</p>
device to fsck	<p>ファイルシステムに対応する raw (キヤラクタ型) 特殊デバイスは device to mount フィールドで指定する (たとえば /dev/rdisk/c0t0d0s0 など)。これにより fsck が使用する raw インタフェースが決定する。読み出し専用ファイルシステムやネットワークベースのファイルシステムのように、適用できるデバイスがない場合はダッシュ (-) を使用する。</p>
mount point	<p>デフォルトのマウントポイントディレクトリ (たとえば /dev/dsk/c0t0d0s6 用の /usr など)</p>
FS type	<p>device to mount フィールドで指定されるファイルシステム形式</p>

表 9-9 /etc/vfstab ファイルのフィールド名とその内容 続く

フィールド名	内容
fsck pass	<p>ファイルシステムをチェックするかどうかを判定するため、fsck が使用するパス番号。フィールドにダッシュ (-) が指定されていればファイルシステムはチェックされず、1 以上の値が設定されていればチェックされる。また UFS 以外のファイルシステムの場合、このフィールドに 0 が設定されていればファイルシステムはチェックされる。UFS ファイルシステムの場合のみ、フィールドの値が 0 のときファイルシステムはチェックされない。</p> <p>fsck が</p> <ul style="list-style-type: none"> ■ このフィールド (fsck pass) の値が 1 より大きい ■ preen オプション (-o p) が使用されている <p>という条件を満たす複数の UFS ファイルシステム上で実行される場合、別のディスクにあるファイルシステムを、効率を上げるためパラレルで自動的にチェックする。フィールドの値が 1 のとき、ファイルシステムは連続してチェックされるが、1 以外であればパス番号の値はまったく影響しない。SunOS 5.6 では、ファイルシステムがチェックされる順序を fsck pass フィールドで明示的に指定することはない。</p>
automount?	<p>システムのブート時、mountall によりファイルシステムを自動的にマウントするかどうかを「yes」または「no」で指定する。ここでは、SunOS 4.x /etc/fstab の 4 カラム目にある auto が「yes」、noauto が「no」と解釈される。このフィールドは automount プログラムとはまったく関係ない。</p>
mount options	<p>カンマで区切ったオプションの一覧 (スペースでは区切らない) で、ファイルシステムのマウントに使用される。オプションがない場合はダッシュ (-) を使う。利用可能なオプションについては、mount (1M) マニュアルページを参照のこと。</p>

コード例 9-1 は SunOS 5.6 /etc/vfstab ファイルの例です。

コード例 9-1 SunOS 5.6 /etc/vfstab ファイルの例

#device #to mount	device to fsck	mount point	FS type	fsck pass	auto mount?	mount options
#						
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/	ufs	1	no	-
/proc	-	/proc	proc	-	no	-
swap	-	/tmp	tmpfs	-	yes	-
/dev/dsk/c0t0d0s6	/dev/rdisk/c0t0d0s6	/usr	ufs	2	no	-
/dev/dsk/c0t3d0s7	/dev/rdisk/c0t3d0s7	/files7	ufs	3	no	-
oak:/export/home1/ignatz	-	/home/ignatz	nfs	-	yes	rw,intr

/etc/vfstab ファイルの詳細については『Solaris のシステム管理 (第 1 巻)』を参照してください。

/etc/shadow ファイル

etc/shadow ファイルは SunOS 5.6 環境に新しく追加されたファイルで、個々のユーザのログインアカウントに対するパスワード有効期限の設定を行うためのエントリや、暗号化されたパスワードが格納されます。/etc/shadow ファイルには、通常の読み出しパーミッションが設定されていません。これにより、以前は /etc/passwd ファイルに入っていた暗号化パスワードへの一般的なアクセスを防ぐことができます。

/sbin ディレクトリ

SunOS 5.6 の /sbin ディレクトリには、ファイルシステムのマウント以前に行われるシステムの初期化に使用される rcs スクリプトだけでなく、システムの実行レベルを変更するために使用される rc スクリプトが格納されます。rc については『SunOS Reference Manual』のセクション 1M 「System Administration Command」のマニュアルページを、スクリプトの説明は 75ページの「システム実行レベルの変更」を参照してください。

/usr ディレクトリ

SunOS 5.6 の /usr ディレクトリには、システムが提供する共有可能ファイルおよび実行可能ファイルが格納されます。表 9-10 では、SunOS 5.6 の /usr ディレクトリに対して、追加されたサブディレクトリについて説明します。

表 9-10 /usr ディレクトリへの追加

サブディレクトリ	説明
/usr/ccs	C コンパイルシステム
/usr/snadm	admintool により使用される実行可能ファイルおよび他のファイル

表 9-11 では、SunOS 5.6 において SunOS 4.x の /usr ディレクトリから移されたファイルを示します。

表 9-11 /usr ディレクトリにおいて変更されたファイル

SunOS 4.x における位置	SunOS 5.6 における位置
/usr/5bin	/usr/bin
/usr/5include	/usr/include
/usr/5lib	/usr/lib
/usr/etc	/usr/sbin
/usr/old	内容を削除
/usr/xpg2bin	/usr/bin
/usr/xpg2lib	/usr/lib
/usr/xpg2include	/usr/include

付録 E には、各ファイルシステムに含まれるディレクトリとファイルの詳細情報を説明する表があります。

/var ディレクトリ

/var ディレクトリには、通常の操作を行なっている間に大きさが変化するファイルが格納されます。/var ディレクトリに入っていたファイルやサブディレクトリの

中には、削除または変更されたものがいくつかあり、また新たに追加されたものもあります。

- `/var/opt/packagename` ディレクトリには、`log` や `spool` ファイルのように大きさが変化するソフトウェアパッケージオブジェクトが格納されます。
- `/var/sadm` ディレクトリには、ソフトウェアパッケージの管理ユーティリティによって管理されるデータベースが格納されます。
- `/var/saf` ディレクトリには、サービスアクセス機能 (SAF) のログファイルおよびアカウントファイルが格納されます。
- SunOS 4.x の `/var/spool/mail` ディレクトリは、`/var/mail` へ移動しました。

`/kernel` ディレクトリと `/opt` ディレクトリが SunOS 5.x で追加されました。

`/kernel` ディレクトリ

SunOS 5.6 の `/kernel` ディレクトリには、オペレーティングシステムのカーネルとカーネルレベルのオブジェクトモジュールが格納されます。これらは SunOS 4.x では `/sys` ディレクトリにありました。表 9-12 では、SunOS 5.6 で `/kernel` ディレクトリに追加されたサブディレクトリについて説明します。

表 9-12 `/kernel` ディレクトリへの追加

サブディレクトリ	説明
<code>/kernel/drv</code>	デバイスドライバと疑似デバイスドライバモジュール
<code>/kernel/exec</code>	実行可能ファイル ELF または <code>a.out</code> を実行するためのカーネルモジュール
<code>/kernel/fs</code>	<code>ufs</code> 、 <code>nfs</code> 、 <code>proc</code> 、 <code>fifo</code> などのファイルシステムを実装するカーネルモジュール
<code>/kernel/misc</code>	その他のモジュール
<code>/kernel/sched</code>	スケジューリングクラスと対応するディスパッチテーブルを含むモジュール
<code>/kernel/ strmod</code>	STREAMS モジュール

表 9-12 /kernel ディレクトリへの追加 続く

サブディレクトリ	説明
/kernel/sys	システムアカウントやセマフォ処理などのロード可能なシステムコール
/kernel/unix	ブート時にロードされるオペレーティングシステムのカーネル

/opt ディレクトリ

SunOS 5.6 の /opt ディレクトリには、オプションのアドオンアプリケーションソフトウェアパッケージが格納されます。SunOS 4.x システムでは、これらのパッケージが /usr ディレクトリにインストールされていました。

/sys ディレクトリ

/sys ディレクトリは削除されました。このディレクトリにはカーネルの再構成に使用されるファイルが入っていましたが、動的カーネルによりカーネルの再構成が行われなくなったため不要となりました。

ファイルシステム管理コマンドの使用

SunOS 5.6 ソフトウェアで変更されたファイルシステム管理コマンドは次のとおりです。

- ファイルシステムのマウント
- ファイルシステムの監視
- ファイルシステムの共有
- 新しいファイルシステムの作成
- ファイルシステムのチェック
- ファイルのバックアップと復元

SunOS 5.6 ソフトウェアでファイルシステムを管理する準備ができたなら、『Solaris のシステム管理 (第 1 巻)』で必要な作業を実行する際の詳細を参照してください。

ファイルシステムのマウントと autofs

マウント機能で大幅に変更されたのは、自動マウント autofs です。autofs プログラムは、たとえば `cd(1)` や `ls(1)` を使用してディレクトリにアクセスするときに、自動的にそのディレクトリをマウントします。この機能にはファイル階層、CD-ROM とフロッピーディスクのファイルシステムなどが含まれます。

システムが実行レベル 3 に入ると、自動的に autofs が起動します。また、ユーザーがシェルコマンド行から autofs を起動することもできます。

autofs は *maps* で指定されるファイルシステムと関係して動作します。これらのマップは NIS、NIS+、またはローカルファイルとして管理されます。autofs マップは特定のファイルに対していくつかのリモートのマウントポイントを指定できます。このようにして、サーバの 1 つがダウンしても、autofs は別のシステムからマウントを実行できます。それぞれのサーバに重み係数を割り当てて、マップの資源にどのサーバが適しているかを指定することができます。

autofs で一部のファイル階層をマウントできますが、マウントコマンドでもファイル階層をマウントできます。ディスクレスシステムは、/ (ルート)、/usr、/usr/kvm についてのエントリを /etc/vfstab ファイルの中に持っていなければなりません。共用ファイルシステムは常に使用できるようにしておく必要があるため、/usr/share のマウントに autofs を使用しないでください。

次の例では、mount コマンドを使用して /etc/vfstab ファイルに登録されているファイルシステムを手作業でマウントする方法を示します。

1. マウントポイントを作成したいディレクトリに変更します。
2. マウントポイントディレクトリを作成します。
3. マウントポイントまたはブロック型デバイスのどちらかを指定します。通常は、マウントポイントを指定する方が簡単です。その他の情報は、/etc/vfstab から読み込まれます。
4. **root** になって、マウントコマンドを入力し、マウントポイントまたはブロック型デバイスのどちらかを指定します。

通常は、マウントポイントを指定する方が簡単です。その他の情報は、`/etc/vfstab` から読み込まれます。

```
# mount mount-point
```

ファイルシステムがマウントされました。

`mount` にオプションを指定して、または指定しないで各種のファイルシステムをマウントする方法については、『*Solaris のシステム管理 (第 1 巻)*』を参照してください。

mount コマンドの変更

`mount` マウントコマンドの名前と書式の一部は、表 9-13 に示すように異なります。

表 9-13 `mount` コマンドの相違

SunOS 4.x	SunOS 5.6
<code>mount</code>	<code>mount</code>
<code>mount -a</code>	<code>mountall</code>
<code>umount</code>	<code>umount</code>
<code>umount -a</code>	<code>umountall</code>
<code>exportfs</code>	<code>share</code>
<code>exportfs -u</code>	<code>unshare</code>
<code>showmount -a</code>	<code>dfmounts</code>
<code>showmount -e</code>	<code>dfshares</code>

これらのコマンドの変更の詳細については、付録 A を参照してください。

/cdrom と /floppy の自動マウント

このリリースでは、取り外し可能な媒体がドライブに挿入されると、CD-ROM とフロッピーディスクのファイルシステムは、自動的に /cdrom や /floppy にマウントされます。これらのファイルシステムは、ボリュームマネージャのデーモン vold(1M) で管理されているため、ユーザが自分でこれらのデバイスをマウントすることはできません。詳細については、68ページの「ボリュームマネージャの使用」を参照してください。

/etc/vfstab ファイルのファイルシステム指定

SunOS 5.6 システムでは、システム起動時にマウントしたいファイルシステムを、/etc/fstab ファイルではなく、ユーザの /etc/fstab に指定する必要があります。/etc/vfstab の書式は /etc/fstab の書式とは異なります。/etc/vfstab ファイルの詳細については、92ページの「/etc/vfstab ファイル」を参照してください。

ファイルシステムの監視

表 9-14 はファイルとディレクトリの監視コマンドを示します。

表 9-14 ファイルとディレクトリ監視コマンド

コマンド	提供される情報	変更 (該当する場合)
ls	ファイルの大きさ、作成日、パーミッション、所有者	なし
du	ディレクトリの合計サイズと内容	なし
df	ファイルシステム、ディレクトリ、またはマウントされた資源で占められるディスク空間。使用済みと未使用のディスク空間。	SunOS 4.x の df コマンドの出力は、SunOS 5.6 の df コマンドとは多少異なる。SunOS 5.6 の -k オプションを使用する出力は、SunOS 4.x の -k オプションの出力と似ている。SunOS 4.x の df -t <i>filesystem</i> は、指定された形式のファイルを表示する。一方、SunOS 5.6 の df -t コマンドは、全リストを合計値付きで出力する。

表 9-14 ファイルとディレクトリ監視コマンド 続く

コマンド	提供される情報	変更 (該当する場合)
quot	ユーザが所有するブロック数	なし
find	検索基準を満たすファイル名	次の SunOS 4.x オプションは、SunOS 5.6 コマンドでは使用できない。-n <i>cpio-device</i> cpio -c フォーマットでデバイスにファイルを書き込む。

ファイルシステムの共用

SunOS 4.x では、ファイルシステムは他のシステムで使用するために「エクスポート」されていました。これは、`/etc/exports` ファイルと `exportfs` コマンドによって行われていました。ただし、エクスポートできるのは NFS システムファイルだけでした。

SunOS では、これと同じ概念を「資源の共用」と呼び、さらに多くのファイルシステムを含めるために拡張しました。ファイルシステムは、`share(1M)` コマンドと `share(1M)` コマンドを使用して共用されます。`share` コマンドは `exportfs pathname` コマンドに似ていて、`shareall` は `exportfs -a` コマンドに似ています。

`share -F fstype` オプションは、共用するファイルシステムの形式を指定します。`-F` オプションを指定しないと、`share` は `/etc/dfs/dfstab` ファイルに登録されている最初のファイルシステム形式を使用します。

自動的に共用したいファイルシステムは、`/etc/dfs/dfstab` ファイル (`/etc/export` ファイルが変更) に `share` コマンドエントリを持っていないければなりません。このファイルに指定されたコマンドは、システムが実行レベル 3 (ネットワークファイルを共用するマルチユーザモード) に入ったときに自動的に実行されます。

/etc/dfs/dfstab ファイルエントリの例

はじめのエントリは、mercury、venus、mars のクライアントに、/export/home1 の読み取り/書き込みを許可します。2行目のエントリは、saturn と jupiter のクライアントに、/export/news の読み取りだけを許可します。

```
share -F nfs -o rw=mercury:venus:mars -d ``Home Dir`` /export/home1
share -F nfs -o ro=saturn:jupiter -d ``News Postings`` /export/news
```

システムがマルチユーザモードで動作しているときに、登録されているクライアントがこれらのファイルシステムを使用できます。share コマンドは、ローカルシステムで共有されているすべての資源を表示します。

```
% share
-      /export/home1  rw=mercury:venus:mars  ``Home Dir``
-      /export/news   ro=saturn:jupiter   ``News Postings``
```

新しいファイルシステムの作成

newfs(1M) または mkfs(1M) コマンドのどちらかを使用するときは、新しいファイルシステムを定義し、指定、作成します。次の節では、newfs と mkfs コマンドの変更点について説明します。

newfs コマンド

SunOS 5.6 の newfs コマンドは、mkfs コマンドの便利なフロントエンドと考えられるコマンドです。newfs コマンドは、仮想ファイルシステムアーキテクチャをサポートしていません。このコマンドは UFS 形式のファイルシステムだけを作成するためのものです。newfs を使用すると、newfs は mkfs を呼び出して引数を渡します。ufs ファイルシステムの作成時に mkfs が実際の作業を行います。

newfs コマンドでは、SunOS 5.6 のデバイス命名規則に準拠する名前だけを使用できます (63ページの「デバイス命名規則」を参照してください)。

mkfs コマンド

SunOS 5.6 の mkfs コマンドは、SunOS 4.x の mkfs コマンドとは大幅に異なります。SunOS 5.6 では、異なるファイルシステム形式を提供し、そのコマンド構文はまったく異なります (86ページの「汎用ファイルシステムコマンド」を参照してくだ

さい)。mkfs では、newfs のように SunOS 5.6 デバイスの命名規則に準拠する名前だけを使用できます。

mkfs は異なる形式のファイルシステムをサポートしていますが、実際にはいつも ufs ファイルシステムを作成するために使用されています。しかし、mkfs は通常は直接実行されず、newfs コマンドで呼び出されるのが普通です。

詳細については、mkfs(1) のマニュアルページを参照してください。

ファイルシステムのチェック

SunOS 5.6 の fsck(1M) コマンドは、SunOS 4.x の fsck コマンドとは大幅に異なります。仮想ファイルシステム (VFS) アーキテクチャに対応するため、fsck ファイルチェックユーティリティには次の 2 つに分けられます。

- ファイルシステムの形式に関係なく最初に呼び出される汎用コマンド。
- 対象となるファイルシステムの形式に応じて、汎用コマンドによって呼び出される特定のコマンド (86ページの「汎用ファイルシステムコマンド」を参照)。

また、fsck では、SunOS 5.6 のデバイス命名規則に適合する名前だけを使用できます。SunOS 5.6 のデバイス命名規則の詳細については、63ページの「デバイス命名規則」を参照してください。

fsck コマンドは、マウント時に、より高速な整合性チェックを実施します。さらに、SunOS 5.6 ソフトウェアは、ルートと /usr ファイルシステムについて fsck を実行した後でシステムをリブートする必要はありません。この結果、従来の SunOS と比較し、システムの起動が高速化されます。fsck -m コマンドを使用すれば、ユーザはクリーンなファイルシステムのチェックを省略できます。詳細については、fsck(1m) のマニュアルページを参照してください。

ファイルのバックアップと復元

この節では、SunOS 4.x と SunOS 5.6 間のバックアップと復元コマンドの変更点と、ufsdump、ufsrestore、dd、tar、および cpio コマンドの使用方法について説明します。

SunOS 4.x は、ファイルをバックアップおよび復元するためのユーティリティ dump、restore、tar、cpio、dd、bar そしてアンバンドルの Backup CoPilot プログラムをサポートしています。このリリースは、bar と Backup CoPilot を除くユーティリティのすべてをサポートしています。SunOS 4.x の bar ファイルは、

SunOS 5.6 システムで復元することができますが、新しい bar ファイルを作成することはできません。dump(8) と restore(8) コマンドは、ufsdump(1M) と ufsrestore(1M) に名前が変更されました。SunOS 4.x の dump コマンドで作成されたファイルは、SunOS 5.6 システム上に ufsrestore で復元できます。

SunOS 5.6 ソフトウェアにはファイルシステムをコピーするための 2 つのユーティリティ volcopy(1M) と labelit(1M) が追加されました。

ufsdump コマンド

ufsdump コマンドは、SunOS 4.x の dump コマンドと同じコマンド構文を使用します。ufsdump は表 9-15 のオプションも使用できます。

表 9-15 dump コマンドで使用できない ufsdump コマンドのオプション

オプション	機能
-l	自動ロード。(ダンプを完了する前に) テープの終端に達したら、ドライブをオフラインにして、テープドライブが再び準備できるまで最高 2 分間待つ。これによって、自動ロード (スタックローダ) テープドライブに新しいテープをロードする時間を与える。2 分以内にドライブが準備できたらロードを続ける。2 分待っても準備ができない場合は、通常どおりオペレータに他のテープをロードするよう要求して待つ。
-o	オフライン。テープまたはフロッピーディスクが終了する (ダンプが完了するか、媒体の終わりに達する) と、ドライブをオフラインにする。フロッピーディスクドライブの場合は、フロッピーディスクをイジェクトする。テープドライブの場合は、テープを巻き戻す。これは別のプロセスがドライブを使用し、誤ってデータを変換してしまわないようにするためである。
-s	ダンプの大きさを見積る。ダンプするのに必要なスペースの合計を求める。そして、ダンプの見積りサイズをバイト数で出力する。これは、増分バックアップにもっとも有効である。

dump と異なり、ufsdump は媒体の終端を検出できるため、-s サイズオプションを使用してダンププログラムに媒体の終端に達する前に次のテープに移らせる必要はありません。ただし、restore コマンドの古いバージョンとの互換性を確保するために、ufsdump では -s オプションを使用できます。

ufsdump は現在では媒体の終端を検出できますが、-s オプションで媒体の大きさを指定しない限り、ダンプに必要なフロッピーディスクやテープの数を予測する方法はありません。したがって、媒体の大きさを指定しない限り、バックアップの開

始時に表示されるメッセージは、必要なフロッピーディスクやテープの数を表示しません。

`-w` と `-W` オプションは、SunOS 5.6 では多少異なります。SunOS 4.x では、これらのオプションは、`/etc/fstab` ファイルに指定されたバックアップ間隔に従って、バックアップのスケジュールが行われたすべてのファイルシステムを表示しました。SunOS 5.6 でこれに相当するファイルである `/etc/vfstab` には、バックアップ間隔を指定する手段がないため、これらのオプションは各ファイルシステムが毎日バックアップされるものと仮定しています。したがってこれらのオプションは、その日のうちにバックアップされていないファイルシステムをすべて表示します。

ネットワークでバックアップを実行するときは (ローカルファイルシステムをリモートテープドライブへバックアップする)、テープドライブを備えたシステムに適したデバイス命名規則を使用します。テープドライブを備えたシステムが SunOS 5.6 システムの場合、デバイス命名規則に従ってテープドライブを識別します。それ以外の場合、SunOS 4.x の規則に従います。

ufsrestore コマンド

SunOS 5.6 の `ufsrestore` コマンドは、SunOS 4.x の `restore` コマンドに似ています。SunOS 4.x の `dump` コマンドで作成した古いバックアップはすべて復元できます。ただし例外があり、フロッピーディスクから、複数のボリュームのバックアップを復元することはできません。`restore` を起動するバックアップスクリプトがある場合は、`ufsrestore` を起動するスクリプトに変更してください。

dd コマンド

SunOS 4.x の `dd` コマンドでは、サイズの接尾辞 `-w` (word の `-w`) は、サイズ単位が 4 バイトであることを意味します。SunOS 5.6 の `dd` コマンドでは、`-w` は 2 バイト単位であることを意味します。さらに、SunOS 5.6 では、`-unblock` と `-block` 変換オプションをサポートしています。

tar と cpio コマンド

`tar` と `cpio` コマンドはバイナリ以外のフォーマットを使用するため、これらのコマンドは、SVR4 の実装間でデータ交換が可能な唯一のユーティリティです。`ufsdump` や `dd` などの他のバックアップユーティリティは、ベンダに固有のもので、ある SVR4 の実装で正常に動作しても別の SVR4 でうまく動作するかどうかは保証されません。

tar コマンドはこのリリースでは変更されていないため、SunOS 4.x コマンドと同じオプションとコマンド構文を使用できます。しかし、SunOS 5.6 ソフトウェアのデバイス命名方法が変更されているため、*tarfile* (または、*device*) 引数が影響を受けます。-f 関数修飾子を使用するときは、デバイス引数を /dev/rmt/unit として指定します。ここで、unit はテープドライブ番号と密度です。表 9-16 は、テープデバイス名のテープドライブ密度を表す文字を示します。

表 9-16 テープデバイス名のテープドライブ密度

密度	説明
指定なし	デフォルトの「適切な」(最高)密度
l	低密度
m	中密度
h	高密度
c	圧縮
u	超高密度

tar コマンドでは、/dev/rmt8 をデフォルト出力デバイスとして使用しません。-f 修飾子を使用せず、TAPE 環境変数が設定されていないときには、tar コマンドは /etc/default/tar ファイルに設定されたデフォルトを使用します。

SunOS 5.6 の cpio コマンドは、SunOS 4.x のオプションとコマンド構文をサポートします。cpio は、表 9-17 に示す多くの新しいオプションを使用できるようにするため拡張されました。

表 9-17 追加された cpio オプション

オプション	オプションで使用できるコマンド	説明
-A	cpio -o	アーカイブにファイルを追加する。
-k	cpio -i	壊れたファイルヘッダと検出した入出力エラーをスキップする。このオプションは壊れた、または順序通りでない媒体からファイルをコピーする。
-L	cpio -o または cpio -p	シンボリックリンクをたどる。
-V	cpio -i, cpio -o または cpio -p	特殊な冗長表示。読み取った、または書き込んだ各ファイルに対してドットを表示する。このオプションは、ファイル名を表示しないで、cpio が動作中であることを保証する。
-C <i>bufsize</i>	cpio -i または cpio -o	<i>bufsize</i> で指定するバイト数単位で、入出力をレコードに分割する。ここで、 <i>bufsize</i> は正の整数。-C または -B を指定しないと、デフォルトのバッファサイズは 512 バイト。
-E <i>filename</i>	cpio -i	アーカイブから抽出するファイル名を含むファイルを指定し入力する。
-H <i>header</i>	cpio -i または cpio -o	<i>header</i> で指定するフォーマットのヘッダ情報を読み取るか、または書き込む。 <i>header</i> には、bar (読み取り専用)、 crc、CRC、odc、tar、 TAR、ustar、USTAR のいずれかを指定できる。
-I <i>filename</i>	cpio -i	入力アーカイブとして <i>filename</i> を読み取る。
-M <i>message</i>	cpio -i -I <i>filename</i> または cpio -o -O <i>filename</i>	媒体を切り替えるときに使用するメッセージを定義する。

表 9-17 追加された cpio オプション 続く

オプション	オプションで使用できるコマンド	説明
<code>-O filename</code>	<code>cpio -o</code>	出力を <i>filename</i> へリダイレクトする。
<code>-R userid</code>	<code>cpio -i</code> または <code>cpio -p</code>	各ファイルの所有権とグループ情報を <i>userid</i> に再度割り当てる。

注 - `cpio` による実行では、`-i` (コピーイン)、`-o` (コピーアウト)、または `-p` (パス) の 3 つのオプションの内のどれか 1 つだけを指定する必要があります。

UFS ログ

Solaris 7 では、UFS ログが提供されます。このプロセスは、UFS ファイルシステムに適用される前のトランザクション (完全な UFS 操作を行うための変更) をログに格納します。格納されたトランザクションは、後でファイルシステムに適用できます。

UFS ログの利点は 2 つあります。第一に、ファイルシステムの不整合を防止することで、`fsck(1M)` の実行が不要になる点です。第二に、`fsck` を迂回できるため、クラッシュまたは異常終了したシステムをリポートするための時間が UFS ログによって軽減される点です。

デフォルトでは、UFS ログは有効になっていません。有効にするには、ファイルシステムをマウントするときに `mount(1M)` コマンドに対して `-o logging` オプションを指定する必要があります。また、`fsdb(1M)` コマンドは、UFS ログをサポートする新しいデバッグコマンドで更新されています。

詳細については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

SunOS 4.x クライアントをサポートする Solaris 7 サーバの設定

この章では、discover4x、install4x、convert4x プログラムを使用して SunOS 4.x ディスクレスのサーバとして Solaris 7 を設定する方法を説明します。

SunOS 4.x から Solaris 7 へアップグレードする場合は、第 3 章を読んでおく必要があります。Solaris 7 を SunOS 4.x のサーバとして設定する場合、以下のことを行う必要があります。

- 111ページの「SunOS 4.x をサポートする機能を Solaris 7 サーバへ追加する」
- 113ページの「discover4x の実行」
- 113ページの「CD-ROM ドライブを install4x 用に設定する」
- 115ページの「install4x の実行」
- 118ページの「convert4x の実行」

SunOS 4.x をサポートする機能を Solaris 7 サーバへ追加する

この節では、SunOS 4.x ディスクレスクライアントにサービスを提供する Solaris 7 サーバを準備する方法について説明します。

注 - この手順でコマンドを使用する前に、すべてのシステムデータが復元されていることを確認してください。/export ファイルシステムにはクライアント情報が格納されているため特に重要です。SunOS 4.x を Solaris 7 にアップグレードする場合は、第 3 章を参照してください。

サイトによってはサーバが Solaris 7 ソフトウェアにアップグレードされた後も、引き続き SunOS 4.x クライアントを使用する必要があります。たとえば、Sun-3™TM システムは、Solaris 2.2 オペレーティング環境または互換性のあるバージョンのソフトウェアを実行することができないため SunOS 4.x を使用しなければなりません。

SunOS 4.x の /export パーティションを、Solaris 7 ソフトウェアが動作しているサーバ上で設定するとき、それをマルチ OS オペレーションと呼びます。マルチ OS オペレーションによって、サーバが Solaris 7 を実行していても、引き続き SunOS 4.x のクライアントにサービスを提供することができます。

マルチ OS オペレーションパッケージは SUNWhinst と呼ばれ、3 つのプログラムが含まれます。Solaris 7 サーバ上に SunOS 4.x の /export ディレクトリを設定するには、これらのプログラムを実行する必要があります。この 3 つのプログラムは次のとおりです。

- **discover4x** - このプログラムは、サーバが Solaris 7 環境に移行した後、SunOS 4.x クライアントをサポートするのに必要なコンポーネントを分析します。プログラムが SunOS 4.x クライアントサポートを調べて、Solaris 7 サーバ上に SunOS 4.x ディスクレスクライアントをインストールするのに必要なデータベースを作成します。あるアーキテクチャに対するクライアントサポートがない場合、discover4x はユーザに install4x を使用してこのサポートを再度インストールする必要があることを知らせます。Solaris 7 環境に移行したサーバと同じアーキテクチャをもつ SunOS 4.x クライアントがある場合は、install4x のコマンドを使用してそのアーキテクチャを再度インストールしなければなりません。
- **install4x** - このプログラムは Solaris 7 環境に移行する前に存在していたディスクレスクライアントをサポートするのに必要な SunOS 4.x システムのコンポーネントをインストールするのに使用します。
- **convert4x** - このプログラムは既存のすべての SunOS 4.x クライアントに関する情報を持つ Solaris 7 サーバを更新します。このコマンドは discover4x と install4x コマンドを実行した後で使用します。更新された情報により、既存の SunOS 4.x クライアントは Solaris 7 サーバのサービスを受けることができます。

これらのインストール手順を開始する前に、SUNWhinst パッケージが正しくロードされていることを確認します。pkginfo(1) コマンドを使用してインストールされているパッケージを一覧表示し、それを調べて SUNWhinst パッケージなど必要なパッケージがすべてインストールされていることを確認してください

パッケージの追加と削除についての詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

discover4x の実行

discover4x はサーバが Solaris 7 環境に移行した後、SunOS 4.x クライアントに必要なサポートを分析します。

スーパーユーザ (root) として次のように入力します。

```
# discover4x
```

discover4x プログラムの実行は、調べるソフトウェア量により 1 秒から 60 秒ぐらいかかります。

discover4x は次のようなメッセージを表示します。

```
Setting up proto root for sun4c arch
Updating server databases to include sun4c sunos 4.1.2 support
Support for sun4c clients must be added using install4x, if \
    sun4c clients are served by this machine.
```

サイトが Solaris 7 のカスタムインストールを行なって、/export ディレクトリの位置を変更した場合、ユーザがそのディレクトリ名を 1 つの引数に指定して discover4x を起動すると、discover4x はそのディレクトリを調べます。たとえば、/export ソフトウェアが /clients ディレクトリに格納されている場合は、次のコマンドを使用します。

```
# discover4x /clients
```

CD-ROM ドライブを install4x 用に設定する

次の節で説明する 3 つの手順の内の 1 つを使用して、Solaris 7 環境のサーバ上で install4x プログラムを実行します。

- システムにローカル CD-ROM ドライブがある場合は、114ページの「ローカル CD-ROM ドライブの使用」を参照してください。
- システムが Solaris 7 動作環境を実行しているシステム上でリモート CD-ROM ドライブを使用する場合は、114ページの「リモート CD-ROM ドライブ (Solaris 7) の使用」を参照してください。
- システムが SunOS 4.x を実行しているシステム上でリモート CD-ROM ドライブを使用する場合は、114ページの「リモート CD-ROM ドライブ (SunOS 4.x) の使用」を参照してください。

SunOS 4.X の CD を CD-ROM ドライブに挿入してから次に進んでください。

ローカル CD-ROM ドライブの使用

ローカルに CD-ROM ドライブを備えたシステムで `install4x` を実行している場合は、ドライブに CD を挿入すると、ボリュームマネージャが `/cdrom/volume1/s0` 上に CD ディレクトリを自動的にマウントします。

リモート CD-ROM ドライブ (Solaris 7) の使用

`install4x` が Solaris 7 環境を動作させているリモートシステムで CD-ROM ドライブを使用する場合は、CD をドライブに挿入すると、ボリュームマネージャが `/cdrom/volume1/s0` 上に CD ディレクトリを自動的にマウントします。その後、次のコマンドを入力してください。

```
# share -F nfs -o ro /cdrom/volume1/s0
```

ブート時に、他の NFS システムを共用していない場合は、`mountd(1M)` デーモンと `nfsd(1M)` デーモンを起動する必要があります。

ローカルシステムで次のコマンドを入力してください。

```
# mkdir /cdrom
# mount -F nfs -o ro cd-host:/cdrom/volume1/s0 /cdrom
```

リモート CD-ROM ドライブ (SunOS 4.x) の使用

`install4x` が SunOS 4.x を動作させているリモートシステムで CD-ROM ドライブを使用する場合は、リモートシステムでスーパーユーザとして次のように入力してください。

```
# mkdir /cdrom
# mount -t hsfs -r /dev/sr0 /cdrom
```

このコマンドを入力したら、`/etc/exports` を編集して、次の行を挿入します。

```
/cdrom -ro
```

次に、リモートシステムで次のコマンドを入力します。

```
# exportfs /cdrom
```

ローカルシステムで次のコマンドを入力します。

```
# mkdir /cdrom
# mount -F nfs -o ro cd-host:/cdrom /cdrom
```

install4x の実行

上記の手順のどれかを行うと、CD が `/cdrom` にマウントされます。ここで、次のように入力して、`install4x` を起動してください。

```
# /usr/sbin/install4x -m /cdrom/volume1/s0 -e /export
```

`-m` オプションを指定しないと、次のプロンプトが表示されます。

```
Enter name of directory where the 4.1* cd is mounted [/cdrom]:
```

`-e` オプションを指定しないと、次のプロンプトが表示されます。

```
Enter name of export directory [/export]:
```

前と同じように、サイトで `/export` ディレクトリの位置をカスタマイズしている場合、次のように `install4x` に別の引数を指定して、ソフトウェアを異なるディレクトリにロードするよう指示できます。

```
# /usr/sbin/install4x -m /cdrom -e /clients
```

ロードするソフトウェアの選択

install4x は、次のような Install Main Menu を表示します。

```
*** 4.1* Install Main Menu ***

Choose an Architecture (then select modules to load):

                                Modules
                                Selected      Loaded
[a] sun4.sun4c.sunos.4.1.2      8          0
[b] sun4.sun4c.sunos.4.1.2      8          0
[c] sun4.sun4m.sunos.4.1.2      7          0

or begin the loading process for all selected modules:

  [L] Load selected module      +-----+
or abort without loading any modules | Disk Usage: |
                                     | 0K Selected |
  [Q] Quit without loading       | 53634K Free |
                                     +-----+

Type any bracketed letter to select that function.

Type ? for help.
```

Install Main Menu 画面にはいくつかのオプションがあります。最初の設定(ここでは a、b、c で表示)は、ソフトウェアをロードするアーキテクチャを指定するのに使われます。他のオプションを使用してユーザは、ソフトウェアのロードの開始 (L)、プログラムの中止 (Q)、ヘルプの表示 (?) を選択できます。

適切なアーキテクチャを選択した後、プログラムは **Module Selection** を表示します。

```
Select sun4.sun4c.sunos.4.1.2 modules:
+[a] R proto root.....240K | [o] User_Diag.....6352K
+[a] R proto root.....240K | [o] User_Diag.....6352K
+[b] R usr.....26240K | [p] Manual.....7456K
+[c] R Kvm.....4832K | +[q] D TLI.....48K
+[d] R Install.....936K | [r] D RFS.....912K
[e] D Networking.....1040K | [s] D Debugging.....2928K
[f] D System_V.....4008K | [t] SunView_Programmers.....1840K
[g] D Sys.....5288K | [u] Shlib_Custom.....1376K
[h] C SunView_Users.....2664K | [v] Graphics.....1784K
[i] SunView_Demo.....512K | +[w] uucp.....608K
+[j] Text.....712K | +[x] Games.....3136K
[k] Demo.....4264K | [y] Versatec.....5960K
[l] C OpenWindows_Users.....25936K | [z] Security.....312K
[m] C OpenWindows_Demo.....4288K | [A] OpenWindows_Programmers..10200K
[n] C OpenWindows_Fonts.....7840K
```

```

Module      + = already loaded      R = Required      C= Common
Legend:    ** = selected for loading  D = Desirable    Others are optional

Select [a-A] or a Quick-Pick Option:
[1] All Required Modules      [4] All Optional Modules
[2] All Desirable Modules     [5] All Modules
[3] All Common Modules
or [D] (done) to return to the main screen
+-----+
| Disk Usage:                |
| 0K Selected                |
| 53634K Free                |
+-----+

```

すでにロードされているパッケージは、**Module Selection** 画面の選択文字の前に + を付けて表示されます (つまり、上記の画面で a、b、c、d、j、q、w、x に対応するパッケージはすでにロードされています)。多数のパッケージが共用されているため、1つのアーキテクチャ用にパッケージをロードするときに、それらのパッケージが他のアーキテクチャ用に「ロードされている」と表示される場合がありますので注意してください。

角括弧内の文字を入力して、ロードするモジュールを選択します。モジュールに関連付けられているキーを押すと、選択状態を切り替えます (つまり、前の状態によって、モジュールを選択したり、選択を解除したりします)。ロードするよう選択されたモジュールには、選択文字の前に ** が表示されます。明らかに重複する処理ですが、プロンプトに対して Y または y で応答すると、現在あるモジュールを再度ロードすることができます。

サンでは、リリースが正常に動作するためにロードしなければならないソフトウェア (選択文字の右に R を表示)、一般にロードされるソフトウェア (C を表示)、ロードするのが望ましいソフトウェア (D を表示) を決めています。

さらに、**Module Selection** 画面では、ロードするモジュールのグループを簡単に選択できます。1 を入力すると、ロードに必要なすべてのモジュールがマークされます。2 を入力すると、推奨されるすべてのモジュールがマークされます。3 を入力すると、通常ロードされるすべてのモジュールがマークされます。4 を入力すると、すべてのオプションモジュールがマークされます。5 を入力すると、**Module Selection** 画面に表示されるすべてのモジュールがマークされます。

D を入力して、**Install Main Menu** に戻ります。

```

*** 4.1* Install Main Menu ***

Choose an Architecture (then select modules to load):

                Modules
                Selected   Loaded
[a] sun4.sun4c.sunos.4.1.2    8         0
[b] sun4.sun4c.sunos.4.1.2    8         0

```

(続く)

```

[c] sun4.sun4m.sunos.4.1.2    7          0

or begin the loading process for all selected modules:

[L] Load selected modules          +-----+
or abort without loading any modules: |  Disk Usage:  |
                                       |  0K Selected  |
[L] Quit without loading           |  53634K Free  |
                                       +-----+

Type any bracketed letter to select that function.
Type ? for help.

```

Install Main menu で L を入力すると、選択したすべてのモジュールをロードできます。次のような出力が表示されます。

```

Installing module 'proto root' [size: 248K]
      in directory /export/exec/proto.root.sunos.4.1.2 ...

Updating server databases ...

Press any key to continue:

```

convert4x の実行

convert4x は、SunOs 4.x クライアントに関するすべての情報で Solaris 7 サーバを更新します。convert4x を実行すると、次のファイルとディレクトリが更新されます。

- /tftpboot – ネットワークブート可能イメージを格納するディレクトリ
- /etc/dfs/dfstab – NFS を経由してエクスポートされるファイルシステムを指定するファイル
- /etc/inet.conf – inetd(1M) がインターネットの要求を受け取ったとき、inetd(1M) が呼び出すサーバのリストを格納するファイル
- /etc/bootparams – クライアント別のブートの指定を格納するファイル
- /etc/hosts – IP と ホスト名のマップを格納するファイル

convert4x を実行する前に、変換するクライアントの /etc/ethers ファイルにイーサネットアドレスが入力されていることを確認します。これは、convert4x が rpc.rarpd(1m) デーモンを起動するために必要です。

スーパーユーザとして、次のコマンドを入力して `convert4x` を実行します。

```
# /usr/sbin/convert4x
```

オプションで、クライアント用ベースディレクトリのフルパスを指定できます。デフォルトでは、`convert4x` は `/export` を検索します。

`convert4x` を実行すると、スクリプトが実行する処理についての情報を画面上に表示します。クライアント情報に矛盾があれば警告します。クライアントの情報が不十分な場合、`convert4x` はエラーを報告し終了します。

`convert4x` が既存のクライアントに対して正常に実行された場合は、`Solstice Host Manager` を使用して既存のクライアントを再度追加する必要はありません。

プリンタ、端末、モデムの管理

この章では、印刷の管理方法と、Solaris 7 環境における印刷コマンドの相違点について説明します。また、Admintool またはサービスアクセス機能 (SAF) によるシリアルポートの管理 (これは端末とモデムの接続を可能にする) についても説明します。

- 121ページの「印刷の変更点の要約」
- 122ページの「プリンタコマンドの使用」
- 125ページの「端末とモデムの管理」
- 125ページの「サービスアクセス機能 (SAF)」

印刷

この節では、Solaris 7 をインストールした後でプリンタを設定し、管理する方法について説明します。また、この章では、SunOS 4.x 環境と Solaris 7 環境でのプリンタコマンドの相違点についても説明します。

印刷の変更点の要約

SunOS 5.6 LP 印刷サービスは、lpd デーモンと lpr、lpq、lprm、lpc の各コマンドで提供された SunOS 4.x 印刷機能に代わるものです。Admintool によって、グラフィカルユーザインタフェースを通してプリンタを設定し管理することができます。また、LP 印刷サービスのコマンド行インタフェースを使用して、SunOS 5.6 プリン

タを管理できます。Admintool の情報と LP 印刷サービスのコマンド行インタフェースの詳細については、『Solaris のシステム管理 (第 2 巻)』を参照してください。

SunOS 4.x ソフトウェアの `/etc/printcap` ファイルにより提供されたサービスは、Solaris 7 環境では `terminfo` データベースと `/etc/lp` ディレクトリ内のファイルによって処理されます。

印刷コマンドと互換パッケージ

システムが SunOS/BSD ソース互換パッケージを実行している場合、多くの SunOS 4.x 印刷コマンドを使用することができます。互換モードは SunOS 4.x コマンド名を、Solaris 7 LP 印刷サービスへのインタフェースとして使用し、実際に SunOS 4.x システムが行うようにそれらを実行するわけではありません。ユーザが SunOS 4.x コマンドを入力して印刷を設定するか、Solaris 7 システムからファイルを出力すると、SunOS 5.6 LP 印刷サービススケジューラによって処理されるメッセージファイルが作成されます。

Solaris 7 での印刷には、SunOS 4.x システムでは使用できない機能が追加されています。これらの機能は、書式、印字ホイール、インタフェースプログラムの制御、ネットワーク印刷サービスの設定などを可能にします。

プリンタコマンドの使用

前の節で説明したとおり、SunOS/BSD ソース互換パッケージがあれば、SunOS 4.x 印刷コマンドを継続して使用することができます。表 11-1 は、基本的なユーザ印刷コマンドの比較を示します。

表 11-1 ユーザ印刷コマンドの比較

SunOS 4.x	SunOS 5.6	機能
<code>lpr filename</code>	<code>lp filename</code>	デフォルトプリンタにファイルを印刷する
<code>lpr -P printer filename</code>	<code>lp -d printer file</code>	特定のプリンタにファイルを印刷する
<code>lpq</code>	<code>lpstat -o printer</code>	デフォルトプリンタでの印刷を待っているファイルの一覧を見る

表 11-1 ユーザ印刷コマンドの比較 続く

SunOS 4.x	SunOS 5.6	機能
/etc/printcap を確認	lpstat -d	どれがデフォルトプリンタかを判別する
/etc/printcap を確認	lpstat -a	どのプリンタを使用できるか判別する
lprm <i>jobnumber</i>	cancel <i>jobid</i>	デフォルトプリンタでの印刷ジョブを取り消す

SunOS release 5.7 プリンタ管理コマンドの使用

この節では、SunOS 4.x と Solaris 7 との間でのプリンタの設定と管理の相違について説明します。実際のシステムサービスはすべて、Solaris 7 環境でしか利用できません。これに対応する SunOS 4.x サービスは、互換モードでも使用できません。

代わりに、System V プリンタ管理コマンド、lpadmin(1M) と lpsystem(1M) を使用しなければなりません。terminfo データベースと /etc/lp ディレクトリ内の構成ファイルを使用してください。詳細は『Solaris のシステム管理 (第 2 巻)』を参照してください。

表 11-2 は印刷設定のためのコマンドの比較を示します。

表 11-2 プリンタ管理、設定、ファイルの比較

SunOS 4.x	SunOS 5.6	機能
lpc	lpadmin	ラインプリンタ機能を制御する
/etc/printcap	terminfo データベースと /etc/lp/printers/ <i>printername</i> /*	プリンタ機能を定義するファイル
/var/spool	/var/spool/lp	印刷システムがスプールとロックファイルを格納するディレクトリ

表 11-2 プリンタ管理、設定、ファイルの比較 続く

SunOS 4.x	SunOS 5.6	機能
なし	lpmove	プリンタ間で印刷待ち行列を転送する
lpc down	reject	プリンタへの待ち行列の実行を停止する

troff による印刷

SunOS 4.x では、troff ファイルをデフォルトプリンタに送信するために次のコマンドが必要です。

```
% troff filename
```

Solaris 7 環境では、パイプ (|) を使用して lp コマンドにファイルを出力する必要があります。表 11-3 は SunOS 5.6 の troff コマンドを示します。

表 11-3 SunOS 5.6 の troff コマンド

SunOS 5.6 コマンド	機能
troff file /usr/lib/lp/postscript/dpost lp	troff ジョブをサポートするデフォルトプリンタに送信する
troff file /usr/lib/lp/postscript/dpost lp -d printer	特定のプリンタに送信する
troff file lp-Ttroff	troff ジョブをサポートする任意のプリンタに送信する

シリアルポート管理

この節では、Admintool またはサービスアクセス機能 (SAF) を使用して端末やモデムの接続ができるシリアルポートの管理について説明します。

『Solaris のシステム管理 (第 2 巻)』では、シリアルデバイスについての Solaris 7 の設定とインストール手順について詳しく説明しています。

端末とモデムの管理

Admintool を使うと、端末とモデム用のシリアルポートソフトウェアを簡単に設定し、変更することができます。

Admintool は次の機能を備えています。

- 一般的な端末とモデム構成のテンプレート
- 複数のポート設定、変更、削除
- 各ポートの状態表示

このツールは、サービスアクセス機能の `pmadm` コマンドの機能を提供するものです。

サービスアクセス機能 (SAF)

SAF を使用すると、すべてのサービスへのアクセスを、ネットワーク上にあるか、ローカルシステムだけに接続されたものかにかかわらず、同じ方法で管理できます。SAF はサービスアクセス制御 (SAC) コマンドを使用してサービスを設定し、管理します。次のようなシステムサービスに同じ方法でアクセスします。

- 端末回線設定を追加、削除、変更する
- ポートモニタを追加する、使用可または使用不可にする、削除する
- 管理データベースファイルから情報を印刷する
- ポートモニタを使用し、管理する
- `listen(1M)` ポートモニタを追加する、使用可または使用不可にする、削除する

以前の SunOS バージョンでは、デバイスを制御する方式は、アクセスを与えるデバイスとそのデバイスの位置の両方に依存していました。ユーザアクセスの管理には多くのデバイスファイルを編集する必要がありました。

SAF はデバイスに依存するシステム管理をなくし、次の機能を含む一連のサービスを管理する共通のインタフェースを提供します。

- ログイン (ローカル、またはリモート)
- ネットワークを通してのファイルへのアクセス

SAF の共通インタフェースは、主に 2 つのコマンド、`sacadm` と `pmadm` を使用します。`sacadm` コマンドはポートモニタ デモンを制御します。`pmadm` コマンドは、ポートモニタに関連付けられたサービスを制御します。

ポートモニタの制御

SAF の共通インタフェースは、ポートモニタと呼ばれる制御サービスを支援します。「ポートモニタ」は、ログインする要求や、プリンタまたはファイルにアクセスする要求を絶えず監視するプログラムです。

ポートモニタは、要求を検出すると、オペレーティングシステムとサービスを要求するデバイスとの間で通信を設定するために必要なパラメータをすべて設定します。その後で、ポートモニタは必要なサービスを提供する他のプロセス (例: `login` プログラム) に制御を渡します。

Solaris 7 動作環境に組み込まれているポートモニタは、`ttymon` と `listen` の 2 種類です。`listen` ポートモニタはネットワークサービスへのアクセスを制御し、リモート印刷およびファイルシステム要求を処理します。`ttymon` ポートモニタは、モデムと英数字端末が必要とするログインサービスへのアクセスを提供します。

SAF 機能と関連プログラム

SAF の共通インタフェースは、主に 2 つのコマンド、`sacadm` と `pmadm` で構成されます。`sacadm` コマンドはポートモニタを制御します。`pmadm` コマンドは、ポートモニタに関連付けられているサービスを制御します。

`sacadm` コマンドは、ポートモニタの追加と削除を行えるようにします。また、`sacadm` コマンドを使ってポートモニタの状態をリスト表示し、ポートモニタをカスタマイズするための構成スクリプトを管理することもできます。

`pmadm` コマンドを使用すると、サービスを追加または削除したり、サービスを使用可能または使用不可の状態に設定できます。たとえば、1 つの `pmadm` コマンドです

すべてのリモートログインを使用不可にできます。また、サービスごとの構成スクリプトをインストールまたは置換したり、サービスに関する情報を表示することもできます。

sacadm コマンドと pmadm コマンドを使用するだけで、システム管理者はリソースへのアクセスを完全に制御できます。ただし、この2つのコマンドは、統合管理環境を可能にするプログラムとプロセスの SAF 一式へのインタフェースにすぎません。機能と関連プログラムは、次のとおりです。

- 全体的な管理 - sacadm
- ポートモニタサービス管理 - pmadm
- サービスアクセス制御 - sac
- ポートモニタ - ttymon および listen
- サービス - ログイン、遠隔手続き呼び出し

サービスアクセス制御 sac は、SAF 群の中で最も重要なプログラムです。マシンが最初に起動される時に、init プログラムによって起動されます。さらに、sac がその管理ファイルに登録されているすべてのポートモニタを起動します。

SAF についての一般的な情報、sacadm コマンドと pmadm コマンドの使用上の相違点についての詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。

ネットワークサービスの管理

この章では、ネットワーク機能の TCP/IP と UUCP の変更について説明します。

- 129ページの「TCP/IP の変更」
- 130ページの「NFS の変更」
- 131ページの「PPP」
- 131ページの「UUCP」

TCP/IP の変更

TCP/IP のユーザインタフェースは既存の Solaris ソフトウェアと実質的には同じですが、NIS+ マップの管理が Admintool を通して処理されます。これは、SunOS リリース 4.x と従来の AT&T SVR4 とは異なります。

Admintool によって管理される NIS+ マップには、以下のものが含まれます。

- ホスト
- サービス
- RPC
- Ethers

SunOS 5.6 の TCP/IP 機能を設定するに当たっては、『TCP/IP とデータ通信』を参照してください。

また、Solaris 7 ソフトウェアには一般的な traceroute ユーティリティが同梱されています。このユーティリティを使用して、インターネットホストまでの IP パケッ

トの経路を追跡します。tracertoute ユーティリティが特に役立つのは、経路の設定や経路指定パスの誤りを判別するときです。

TCP と SACK

TCP 選択肯定応答 (TCP SACK) は、RFC 2018 に記述されたサポートを提供します。これによって、特に衛星リンクまたは大陸間リンクを介して TCP ラージウィンドウ (RFC 1323) を使うアプリケーションで、情報の流れの混雑およびパケット欠落に関連する問題が解決されます。

NFS の変更

Solaris 7 環境では、新しいコマンドとファイルを使用して資源を共用し、NFS 資源を管理できます。とくに、`exportfs` と `/etc/exports` は `share`、`shareall` および `/etc/dfs/dfstab` に置き換えられています。この新しいコマンドセットは、将来の分散ファイルシステムタイプが使用できるように設計されています。

NFS に関連付けられているデーモンのいくつかは、名前が変更されています。`rpc.statd`、`rpc.lockd`、および `rpc.mountd` は現在は、単に `statd`、`lockd` および `mountd` と呼ばれています。

SunOS 4.x 環境とは異なり、Solaris 7 にはクライアント側のブロック I/O デーモン (`biod`) はありません。これはカーネルスレッドに置き換えられています。また、NFS デーモンである `nfsd` も、複数のコピーを生成して要求を平行処理しないように変更されています。

このリリースに組み込まれているその他の機能は、次のとおりです。

- TCP 上の NFS
- NFS バージョン 3
- 改善された NFS ロックマネージャ
- アクセス制御リスト (ACL) のサポート
- WebNFS
- NFS クライアントフェイルオーバー
- NFS ファイルシステムの Kerberos サポート
- NFS 大型ファイルのサポート

これらの機能については、『NFS の管理』で説明しています。

PPP

Solaris 7 の PPP は、インターネットプロトコル群に含まれる標準データリンクレベル、ポイントツーポイント・プロトコル (PPP) の非同期方式の実装です。PPP によって、ネットワーク管理者はモデムと電話回線を使用して通信リンクを確立することができます。PPP でのネットワーク展開についての詳細は『TCP/IP とデータ通信』を参照してください。

LDAP

Lightweight Directory Access Protocol (LDAP) は、オープンスタンダードでプラットフォームに依存しない、X.500 非公式モデルに基づくアクセスプロトコルです。このプロトコルは、TCP/IP で実行するように設計されており、単純な文字列エンコード方式を採用しています。LDAP アプリケーションは、クライアント/サーバアプリケーションです。このリリースに含まれるクライアントライブラリを使用すれば、開発者は LDAP アプリケーションを作成し、ユーザーは LDAP 対応アプリケーションを実行することができます。

IIIMP

Solaris 7 ソフトウェアに実装された Internet Intranet Input Method Protocol (IIIMP) によって、Solaris、Java、非 X Windows アプリケーションで提供される入力方式の間にシームレスな相互運用性が実現します。

UUCP

Solaris 7 UNIX-to-UNIX Copy (UUCP) は、SunOS 4.x システムで利用できる HoneyDanBer UUCP と似ています。これは、同じ構成ファイル、スクリプト、コマ

ンドを使用します。したがって、SunOS 4.x ファイルで行なった変更をこのリリースで復元し、またスクリプトを実行することができます。ただし、Solaris 7 のスプールディレクトリはジョブの処理順序により編成が異なります。ジョブの処理順序は作業負荷を分類し優先順位を決めるメカニズムです。

表 12-1 に、SunOS 4.x にはなかった、Solaris 7 UUCP と対応する新しいファイルおよびコマンドについて説明します。表 12-2 では、Solaris 7 UUCP に追加されたログファイルについて説明します。

表 12-1 新しい SunOS 5.6 UUCP ファイルおよびコマンド

コマンドまたはファイル	機能
D. データファイル P. データファイル	これらのデータファイルは、UUCP コマンド行がソースファイルをスプールディレクトリへコピーすることを指定すると作成される。データファイルのフォーマットはすべて <i>systemxxxxyyy</i> 。 <i>system</i> はリモートシステムの最初の 5 文字。 <i>xxxx</i> は UUCP が割り当てる 4 桁のジョブシーケンス番号。 <i>yyy</i> は、作業ファイル (c.) 用に作成される D. ファイルを区別するためのサブシーケンス番号。
/etc/uucp/Grades	テキストグレード名をシステム名にマップする。
/etc/uucp/Limits	実行できる並列 UUCP セッションの数を指定する。前のバージョンの <i>Maxuuscheds</i> と <i>Maxuuxqts</i> ファイルは削除される。
/etc/uucp/Config	UUCP の調整可能なパラメータを無効にする情報を含む。現在、利用可能なパラメータは <i>Protocol</i> のみ。したがって、システム管理者は通常このファイルを修正する必要はない。
uuglist	<i>uucp(1C)</i> と <i>uucp(1C)</i> の <i>-g</i> オプションによりシステム上で利用できるサービスグレードを一覧表示する。

Solaris 7 UUCP には、システム管理に影響を与える機能がいくつか追加されています。

- チェックポイント設定・再起動機能
- UUCP 伝送を制御するジョブの処理順序
- システムが実行できる並列 UUCP セッションの数を制限し、また調整可能な UUCP パラメータを無効にする 2 つの新しい構成ファイル。

このあとの節では、これらの変更によって生じたシステム管理の相違について説明します。

チェックポイントからの再起動

通信リンクの異常により SunOS 4.x システム間の UUCP 伝送が中断したとき、通信が再起動されると、伝送がファイルの先頭からふたたび始まりました。Solaris 7 UUCP を実行する 2 つのシステム間の通信では、先頭から始まるのではなく、中断された位置で始まります。これにより、特に不安定な、またはノイズの多い伝送回線でのスループットが改善できます。

システムは、2 つの新しいファイルを使用して送受信されたデータを格納し、ファイルの大きさを比較して伝送をどこで再開するかを判定します。システムは .P ファイルを使用して受信したデータを格納し、.D ファイルを使用して送信されたデータを格納します。これらのファイルは前の UUCP バージョンの TM. ファイルを置き換えるものです。ただし、1 つのシステムだけが SunOS 5.6 UUCP を実行している場合は、比較は行われず、伝送は先頭から再開します。

ユーザジョブの処理順序

ジョブの処理順序機能によって、管理者はジョブを作業負荷に分割し、大きさ、タイプ、優先順位のどれかが類似する、またはそれらすべてが類似するほかの作業負荷と競合させることができます。これらの要素のどれか 1 つ、または組み合わせで、作業負荷を分類することができます。また、アクセスパーミッションを設定して、ユーザとグループに各グレードの UUCP サービスを獲得させることもできます。

SunOS 4.x では、ジョブを依頼するときユーザが処理順序を選択しなければなりません。処理順序は 1 文字で表され、Solaris 7 環境のように名前ではありません。Solaris 7 では、管理者はサイト全体のジョブの処理順序を定義できます。

Limits ファイル

/etc/uucp/Limits ファイルは、システムで許可される並列 uucico、uuxqt、uusched プロセスの最大数を指定します。前のリリースの Maxuusched パラメータと Maxuusched パラメータがこの1つのファイルに置き換わります。

Config ファイル

/etc/uucp/Config ファイルには、調整可能な UUCP パラメータを無効にする情報が入っています。現在利用できる唯一のパラメータは Protocol で、通常はシステム管理者が変更すべきではありません。

ログファイル

Solaris 7 UUCP には、前バージョンで提供される4つのログファイルに加えて、4つの新しいログファイルがあります。これらのファイルは、アカウント、コマンド、性能、セキュリティ情報を記録します。コマンドログファイルとセキュリティログファイルは、それらが存在しなければ作成されます。アカウントと性能ログファイルについては、それらがすでに存在する場合のみ書き込まれます。

表 12-2 新しい SunOS 5.6 UUCP ログファイル

ファイル名	機能
/var/uucp/.Admin/アカウント	請求書発送のアカウント情報を記録する。
/var/uucp/.Admin/perflog	uucico 操作上の統計情報を記録する。
/var/uucp/.Admin/security	セキュリティ侵犯行為を記録する。
/var/uucp/.Admin/コマンド	ユーザまたは管理者によって発行されたコマンドに関する情報を記録する。

SunOS 5.6 UUCP の設定と使用準備ができたなら、詳細については『TCP/IP とデータ通信』を参照してください。

ネームサービスの使用方法

SunOS 4.x 環境の一部であるネットワーク情報サービス (NIS) は、広範囲にわたってネットワーク情報サービスプラス (NIS+) に移行しつつあります。NIS+ は SunOS 5.0 システムで導入され、クライアント/サーバ環境の変化を考慮して、完全に設計し直されたネームサービスです。DNS (ドメインネームシステム) は、企業間のインターネット通信で現在すでによく使用されるネームサービスです。この章では NIS+ について説明し、それを NIS および DNS と比較します。

- 136ページの「ネームサービススイッチ」
- 136ページの「NIS+」
- 136ページの「DNS」
- 136ページの「DNS と NIS+ の比較」
- 137ページの「NIS と NIS+ の比較」
- 139ページの「NIS+ アップグレードの計画」

NIS+ アップグレードの計画と NIS+ のインストールについての詳細は、『NIS+ への移行』、『Solaris ネーミングの設定と構成』を参照してください。

注 - Solaris 7 環境のシステム管理マニュアルセットでは、NIS+ を使用しているシステムに重点を置いています。

ネームサービススイッチ

Solaris 7 環境は、標準ネーミングインタフェース (たとえば、`gethostbyname`) を使用して複数のネームサービス (特に NIS、NIS+、DNS) をサポートし、それによってアプリケーションは異なるサービスから透過的にデータにアクセスすることができます。この一例が Solaris 7 環境でのネームサービススイッチ機能であり、アプリケーションは UNIX 標準ネームインタフェース (たとえば、`getxxbyyy` インタフェース) をこのように使用することができます。詳細は `nsswitch.conf(4)` のマニュアルページを参照してください。

NIS+

NIS+ は、ONC トランスポート独立遠隔手続き呼び出し (TI-RPC) インタフェースの最上部に構築されるネームサービスです。NIS+ は、セキュリティ、性能、スケーラビリティ、管理の点で NIS を大幅に上回っています。

DNS

DNS は、自動的に管理されるネームサーバにより階層名前空間モデルをサポートします。NIS+ は DNS 階層ネーミングモデルを使用しますが、企業ネットワークの変化するシステム管理データやその他の要件のサポートに重点を置いています。

したがって、DNS と NIS+ とは相互に補足し合うネームサービスです。

- DNS は企業間通信に使用されます。
- NIS+ は企業ネットワークの管理をサポートします。

DNS と NIS+ の比較

表 13-1 は、NIS+ と比較した DNS の機能と利点を示します。

表 13-1 DNS と NIS+ の機能と利点の比較

機能	DNS	NIS+
セキュリティ	データへの無制限のアクセス	操作はすべて、オプションにより認証可能 オブジェクトおよびエントリに対するアクセス権は、システム管理者が設定できる。
API とヒューマンインタフェース	ネームサービスへの読み取り専用アクセスを許可	ネームサービスへの読み書きアクセスを許可 - ネットワーク環境の変化に対する効率的なサポート - API は管理操作をサポート - 管理とほかの分散アプリケーションのサポート
更新	ゾーンマスタファイルの転送による	増分データ転送による - ネットワーク環境の変化に対するすばやいサポート - 強い一貫性
NIS との互換性	なし	既存の NIS アプリケーションはスムーズに移行可能
データサポート	ASCII データ (パケットサイズに制約あり)	バイナリと ASCII データ - 多種多様の情報のサポート - より大きなオブジェクトのサポート

DNS の主な特長は、階層データベースパーティションや、比較的静的な情報のエントリ (ホスト名や IP アドレスなど) を収めた複製をサポートすることです。DNS により、インターネットとの接続が保証されます。

一方、NIS+ は変化する社内ネットワーク管理情報 (email 別名、イーサネットアドレス、RPC プログラム番号など) の安全なリポジトリと考えることができます。

NIS と NIS+ の比較

NIS+ には、NIS に追加された機能があります。その要約を表 13-2 に示します。

表 13-2 NIS と NIS+ の機能の比較

機能	NIS	NIS+
名前空間	平坦で階層型でない構造に編成。独立したネットワークドメインごとの集中化平坦ファイルデータベース	階層構造に編成。各ネットワークサブセットまたは自立したドメインをサポートするために、ディレクトリに分割
データ格納方式	キーと値の対がある複数の 2 列「マップ」	複数の検索可能な列があるテーブル
ドメイン間の資源アクセス	サポートなし	認定ユーザに対して許可
更新の特権	更新には、マスタサーバ上のスーパーユーザ特権が必要	更新は、権限を持つ管理者によってリモートに行うことが可能
更新プロセス	更新では、マスタサーバ上の <code>make</code> ファイルの使用が必要	更新は、コマンド行インタフェースを通じて容易に実行
更新の通知	管理者が行い、マップ全体の転送が必要	増分転送による自動および高速伝播認証
セキュリティ	データベースが安全ではない	NIS+ ディレクトリ、テーブル列、エントリに対する詳細なアクセス制御
コマンドおよび関数の接頭辞	接頭辞として文字 <code>yp</code> が付く (例: <code>ypmatch(1)</code> および <code>ypcat(1)</code>)	接頭辞として文字 <code>nis</code> が付く (例: <code>nismatch(1)</code> および <code>nischown(1)</code>)

NIS+ には、NIS サイトが新しいネームサービスに円滑に段階的な方法で移行できるようにする機能が組み込まれています。NIS+ に移行する NIS サイトには、次のような利点があります。

- 許可を持つユーザによるネットワークドメインの分散リモート管理
- 階層ドメインに対するサポート
- マスタから複製サーバへの更新の高速自動伝達
- テーブルとネットワーク資源に対する詳細なアクセス制御
- より簡単で整合性のある管理操作
- 強化されたネームサービス信頼性と可用性

NIS+ アップグレードの計画

NIS+ は、次のオペレーティング環境の組み合わせをサポートします。

- すべてのサーバとクライアントにインストールされた SunOS 5.6
- 1つのサーバにインストールされているが、いくつかの SunOS 4.x サーバとともに動作する SunOS 5.6

ネットワークについては、次のように NIS から NIS+ のネームサービスへの 3 つの主な移行方法があります。

- すべてのサーバとクライアントを NIS+ にアップグレードする
- すべてのサーバを同時に NIS+ にアップグレードし、SunOS 4.x クライアントをサポートできるように互換モードを有効にする
- NIS と NIS+ が共存するように、異なるドメイン名を使用する

ネットワークのアップグレードを行う最初の手順は、どのサーバを NIS+ ネームサービスにアップグレードし、どのサーバで NIS の実行を継続するかを決めることです。詳細は『NIS+ への移行』を参照してください。

Solaris 共通デスクトップ環境

各種ワークステーションのメーカー間での互換性を持つ Solaris 共通デスクトップ環境 (CDE) は、Solaris 7 ソフトウェアまたは互換バージョンを実行している Sun™ Workstation™ のデスクトップグラフィカルインタフェースとして機能します。このウィンドウ環境は、ユーザの作業の編成と管理に役立ちます。デスクトップには、各種のウィンドウ、ワークスペース、コントロール、メニュー、およびフロントパネルが用意されています。はじめてウィンドウ環境にログインする時には、デフォルトのデスクトップとして OpenWindows または Solaris CDE のどちらを使うか選択できます。

- 141ページの「Solaris 共通デスクトップ環境について」
- 142ページの「デスクトップの概要」
- 145ページの「OpenWindows 環境から CDE への移行」

Solaris 共通デスクトップ環境について

1993年3月、サン・マイクロシステムズ社、Hewlett-Packard Company、IBM (International Business Machines Corp.) および Novell, Inc. の各社は UNIX ベースの主要ワークステーションとデスクトップコンピュータに一貫性のある見た目と使い心地を提供するグラフィカルユーザインタフェースの開発協定を発表しました。CDE の開発に向けて、エンドユーザとアプリケーション開発者にとって UNIX をよりいっそう使いやすいものにするという目標が当初から設定されました。

この共同開発の結果生まれたのが共通デスクトップ環境 (CDE) です。CDE は Solaris 7 環境にパッケージングされた 2 つのデスクトップのうちの 1 つです (もう 1

つは OpenWindows デスクトップ)。いずれ、CDE はサン・マイクロシステムズ社、Hewlett-Packard、IBM、Novell、さらに UNIX ワークステーション市場の多くの企業にとって標準のデスクトップになるでしょう。サン・マイクロシステムズ社は Solaris 7 において、CDE の旧バージョンにない新しいデスクトップ機能で CDE を拡張しました。新機能のいくつかについては、この章で後述しています。

Solaris CDE には、デスクトップサーバ、セッションマネージャ、ウィンドウマネージャ (Hewlett-Packard の Visual User Environment に基づくもの)、および多数のデスクトップユーティリティが組み込まれています。

Solaris CDE の使用方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』を参照してください。

開発者、エンドユーザ、および CDE

CDE は主要 UNIX プラットフォームに一貫性のあるコンピューティング環境を提供するため、エンドユーザは異機種間を容易に移動できます。また、規格合致しているサン・マイクロシステムズ社、Hewlett-Packard、IBM、Novel の各プラットフォーム向けにプログラミングインタフェースの標準セットを 1 つ提供することによって、CDE はアプリケーション開発を容易にします。1 つの API を使うため、開発者は CDE に準拠したシステム間で一貫した表示形式と動作を備えたアプリケーションを作成することができます。

CDE 開発環境は X11R5 サーバをベースとしており、OSF/Motif 1.2 仕様に基づく見た目と使い心地を備えたアプリケーションが作成できます。

デスクトップの概要

Solaris CDE デスクトップ機能には、次のものがあります。

- フロントパネル
- スタイル・マネージャ
- ファイル・マネージャ

フロントパネル

フロントパネルは、ディスプレイの下部にある特殊ウィンドウです。これは、ユーザが日常的に使用する各種コントロール、インジケータ、サブパネルを提供します。また、フロントパネルにはワークスペースを選択するためのワークスペーススイッチも提供します。

フロントパネルにある多くのコントロール (たとえば、ファイル・マネージャ・コントロール) をクリックすると、アプリケーションが起動されます。プリンタ・コントロールのように、ドロップ領域としても機能するコントロールもいくつかあります。ファイル・マネージャからファイル・アイコンをドラッグして、それをプリンタコントロールの上にドロップすると、そのファイルを印刷できます。

フロントパネルにある多数のコントロール上の矢印ボタンはそれぞれサブパネルに対応しています。矢印ボタンをクリックすると、サブパネルがオープンします。

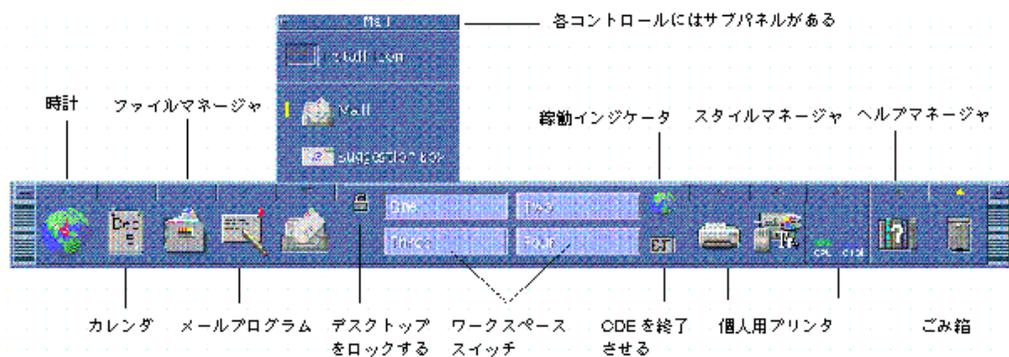


図 14-1 フロントパネルの各種コントロール

上図のフロントパネルでは、メールのアイコンの上にある矢印アイコンをクリックされ、メールプログラムのサブパネルが表示されています。時計のアイコンをクリックすると、デフォルトの Web ブラウザが起動します。

スタイル・マネージャ

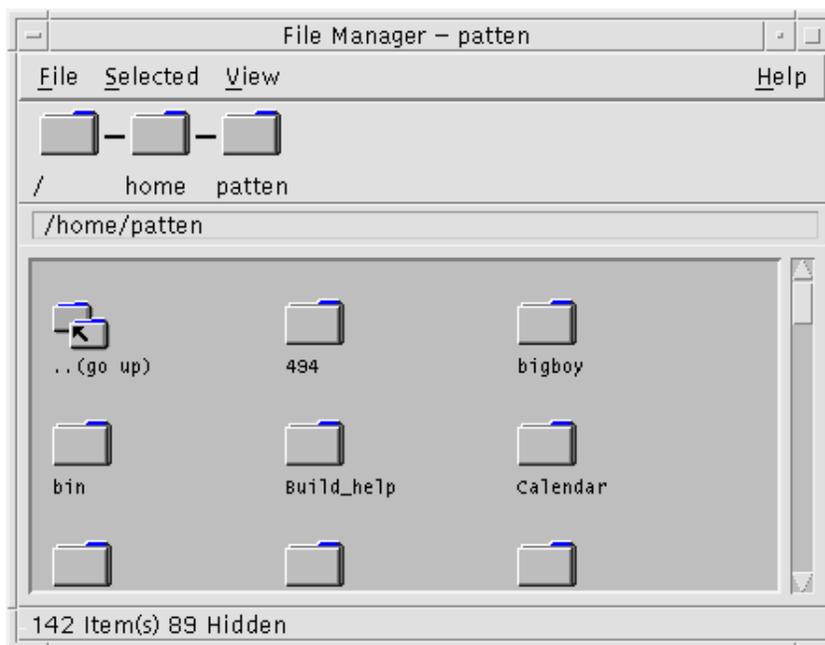


このアイコンをクリックして、起動します。

スタイル・マネージャを使用すると、次に示すようなデスクトップの多数の要素を簡単にカスタマイズできます。

- カラー
- ワークスペースの背景
- フォントサイズ
- キーボード、マウス、およびウィンドウの動作

ファイル・マネージャ



ファイル・マネージャのアイコンをクリックして、起動します。

各種ファイル、フォルダ、アプリケーションをシステム上にアイコンとして表示します。

OpenWindows 環境から CDE への移行

Solaris 7では、ログイン画面から OpenWindows デスクトップと CDE デスクトップのどちらにログインするか選択できます。ログイン方法の詳細については、ログイン・マネージャヘルプボリューム、または『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第2章「デスクトップセッションの開始」を参照してください。

デスクトップサービス

OpenWindows 環境で使い慣れていたデスクトップサービスは、Solaris CDE では別の場所に配置されています。表 14-1 に、位置の変更されたデスクトップサービスの主なものを示します。

表 14-1 デスクトップサービスの位置

デスクトップサービス	OpenWindows	CDE
ログアウト	ワークスペースメニュー	フロントパネル
画面のロック	ユーティリティメニュー	フロントパネル
ワークスペースのカスタマイズ	ワークスペースメニュー	スタイル・マネージャ
ワークスペースの保存	ユーティリティメニュー	スタイル・マネージャ
再表示	ユーティリティメニュー	フロントパネル
プロパティ	ワークスペースメニュー	スタイル・マネージャ
ヘルプ	ワークスペースメニュー	フロントパネル、アプリケーション・マネージャ、ワークスペースメニュー

CDE でのウィンドウ、メニュー、ボタンおよびマウスの使用

Solaris CDE での各種ウィンドウ、メニュー、ボタンおよびマウスの使い方は、OpenWindows 環境とは多少異なります。ウィンドウ、メニュー、ボタンおよびマウスの使い方の詳しい説明については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第 1 章「基本スキル」を参照してください。

ワークスペースアプリケーションメニューへのアクセス

OpenWindows 環境では、アプリケーションを起動する主な方法はワークスペースメニューを使うことでした。ワークスペースメニューは Solaris CDE でも使用できますが、ワークスペースの機能への主なアクセスポイントはフロントパネルになっています。

ワークスペースメニューから利用できるアプリケーションにはフロントパネル上の各項目の他に、アプリケーション・マネージャ内部で利用できるアプリケーションのサブセットもあります。アプリケーション・マネージャの詳しい説明については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第6章「デスクトップからのアプリケーションの実行」を参照してください。

スタイル・マネージャおよびワークスペースのカスタマイズ

スタイル・マネージャから使用できる項目はカラー、フォント、背景、キーボード、マウス、ビープ音、画面、ウィンドウ、および起動です。これは、OpenWindows 環境でのワークスペースプロパティウィンドウに代わるものです。スタイル・マネージャの詳しい説明については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第7章「デスクトップ環境のカスタマイズ」を参照してください。

CDE での OpenWindows アプリケーションの実行

CDE アプリケーション・マネージャにある OpenWindows というタイトルのフォルダには、OpenWindows アプリケーションが含まれています。

コマンド行から OpenWindows アプリケーションを実行した場合、Solaris CDE ではそのアプリケーションを同じように端末エミュレータ (端末アプリケーション) から実行できます。アプリケーション・マネージャの詳しい説明については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第6章「デスクトップからのアプリケーションの実行」を参照してください。

アプリケーションの設定とプロパティ

OpenWindows 環境では、アプリケーション全体の設定値は編集メニューからアクセスしたプロパティダイアログボックスで設定されます。CDE では、アプリケーション全体の設定値はオプション領域から設定されます。オプションの選択項目は

通常はアプリケーションのファイルメニューの下、または別のメニュー項目「オプション」に配置されています。

CDE では、プロパティ (アプリケーションに存在する場合) はアプリケーションの編集メニューの下に表示され、日付や名前などのオブジェクトの特性の設定に使われたり、活字などのオブジェクトの識別特性の表示に使われます。CDE では、フォーマットの設定値は通常はフォーマットメニューの下に表示され、段落、ファイル、またはメッセージごとに余白と段落位置合わせを設定できます。

CDE グローバル・オプションは、OpenWindows 環境のワークスペースメニューから設定したプロパティに似ています。これらのプロパティは、CDE のスタイル・マネージャアプリケーションから設定します。『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第7章「デスクトップ環境のカスタマイズ」を参照してください。

キーボードのデフォルト値の変更

OpenWindows 環境でキーボードのデフォルト値を変更していない場合、その値は CDE 内でも同じままです。デフォルト値を変更するには、スタイル・マネージャのキーボードダイアログボックスを使用します。『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第7章「デスクトップ環境のカスタマイズ」を参照してください。UNIX キーボードの割り当てを変更する必要がある場合は、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第10章「テキストエディタの使い方」を参照してください。

マウスのデフォルト値の変更

OpenWindows 環境でマウスのデフォルト値を変更していない場合、その値は CDE 内でも同じままです。デフォルト値を変更するには、スタイル・マネージャのマウスダイアログボックスを使用します。機能の名前がいくつか変更されています。ダブルクリック、アクセラレーション、しきい値はそのまま使用できます。CDE のマウスボタン順序は「利き腕」と呼ばれています。『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の第1章「基本スキル」を参照してください。

パート II 開発者用移行情報

C 言語とその関連ツールは、SunOS 4.x から Solaris 7 になって大幅に変更されています。これらの変更は、すべての開発者にさまざまな影響を与えます。オペレーティングシステムのカーネルと、そのインターフェースも SunOS 4.x ソフトウェアとは大幅に異なっています。パート II では、これらの違いについて説明し、両リリース間の類似点を指摘し、既存のソフトウェアを移植したり、Solaris 7 用に新しいソフトウェアを開発したりするために必要な情報を提供し、プログラミング環境との関連についても説明します。



コンパイラ、リンカ、デバッガ

この章では、コンパイラ、リンカ、デバッガについて説明します。この章の内容は次のとおりです。

- 151ページの「コンパイラ」
- 152ページの「リンカ」
- 160ページの「デバッガ」

コンパイラ

SunOS 4 から Solaris 7 に移行する開発者にとって最も大きな変更点は、C コンパイラがバンドルされなくなったことです。コンパイラをバンドルしない理由の1つに、動的なカーネルがあります。動的なカーネルでは、必要に応じて自動的にデバイスが追加されるので、カーネルの再構築にコンパイラを使用する必要がありません。

Sun WorkShopTM には、ANSI C 互換コンパイラのほか、拡張されたデバッグ機能とプログラム開発環境があります。このコンパイラは、Solaris 7 のネイティブオブジェクト形式である 実行形式リンク形式から成る ELF 形式で実行可能ファイルを作成します。lint と lint ライブラリも Sun WorkShop の一部として提供されています。lint と lint ライブラリも Sun WorkShop の一部として提供されます。

Sun WorkShop については、<http://www.sun.com> にアクセスしてください。

『*Making the Transition to ANSI C*』は、バンドル製品の Sun OS 4.x C コンパイラとアンバンドル製品の Sun WorkWhop C コンパイラのそれぞれによって実装される C 言語の違いを解説しています。一方のコンパイラ用のソースを他方に移植すると

きに参照してください。このマニュアルは、<http://docs.sun.com> の Programming/Languages で、*Sun WorkShop Compiler C4.2 AnswerBook Collection* の一部として閲覧できます。

Sun WorkShop C コンパイラのオプションフラグ -Xs は、K&R C と ANSI C で動作が異なる言語構成要素について警告します。この特別なフラグについては、『C User's Guide』を参照してください。このマニュアルは、<http://docs.sun.com> の *Sun WorkShop Compiler C 4.2 AnswerBook Collection* でも閲覧できます。

リンカ

このリリースではリンクエディタ ld(1) に対していくつかの変更があります。最も重要な変更は新しい ELF のファイルフォーマットを処理する機能です。

注 - ライブラリと実行可能プログラムを構築するには、リンカを直接起動することよりもコンパイラドライバによる方法をお勧めします。コンパイラは、リンカが必要とする多数のファイルを自動的に供給します。

ライブラリを混合することはできません。32 ビットプログラムは 32 ビットライブラリ、64 ビットプログラムは 64 ビットライブラリとリンクする必要があります。ELF32 オブジェクトは他の ELF32 オブジェクト、ELF64 オブジェクトは他の ELF64 オブジェクトとリンクします。

リンクエディタオプションの相違

新しいリンカでリネームされたオプションもあれば同じものもあり、また不要になったオプションもあります。表 15-1 では SunOS 4.x の ld を Solaris 7 の ld コマンドと比較します。

表 15-1 に続く節で、リンク作業がオプションの相違によってどのように影響を受けるかについて説明します。

表 15-1 ld オプションの比較

SunOS 4.x のオプション	Solaris 7 での変更	注
-align <i>datum</i>	-M <i>mapfile</i>	<i>mapfile</i> と異なるセクションの使用
-assert definitions	デフォルト	
-assert nodefinitions	-znodefs	警告ではなく致命的エラーを発行
-assert nosymbolic	-zdefs	警告ではなく致命的エラーを発行
-assert pure-text	-ztext	警告ではなく致命的エラーを発行
-A <i>name</i>	変更なし	dlopen(3X) と dlclose(3X) はこの動作に接近可能
-Bdynamic	-Bdynamic	共用ライブラリの取り込みにのみ適用される。動的にリンクされた実行可能プログラムを構築するには -dy (デフォルト) を使用。155ページの「実行可能ファイルの作成」を参照。
-Bnosymbolic	-zdefs	
-Bstatic	-dn & -Bstatic	動的なリンクを完全に除去するには、-dn オプションを指定しなければならない。アーカイブライブラリを取り込むために動的モードで -Bstatic を使用 (トグルとして使用。155ページの「実行可能ファイルの作成」を参照)。
-Bsymbolic	-Bsymbolic	このオプションを付けて -assert nosymbolic も取得する。
-d -dc -dp	デフォルト	オフに設定するには、SVR4 で -b オプションを使用しなければならない。
-D hex	-M <i>mapfile</i>	<i>mapfile</i> には、希望する結果を達成するためにいろいろなメカニズムが含まれる。
-e <i>entry</i>	-e <i>entry</i>	

表 15-1 ld オプションの比較 続く

SunOS 4.x のオプション	Solaris 7 での変更	注
no -e	-G	共有オブジェクトを作成する。
-lx[.v]	-lx	共用ライブラリのメジャー番号が示すバージョンだけが現在サポートされている。
-Ldir	-Ldir	<i>dir</i> は実行可能プログラムに記録されない。かわりに、-R オプションを使用。
-M	-m	
-n	デフォルト	SVR4 の実行可能プログラムのフォーマットは、ディスクイメージを -n として圧縮
-N	変更なし	
-o <i>name</i>	-o <i>name</i>	
-P	デフォルト	-M <i>mapfile</i> で取り消し可能。
-r	-r	
-S	変更なし	
-s	-s	
-t	変更なし	
-T <i>hex</i>	-M <i>mapfile</i>	<i>mapfile</i> には、希望する結果を達成するためにいろいろなメカニズムが含まれる。
-Tdata <i>hex</i>	-M <i>mapfile</i>	<i>mapfile</i> には、希望する結果を達成するためにいろいろなメカニズムが含まれる。

表 15-1 ld オプションの比較 続く

SunOS 4.x のオプション	Solaris 7 での変更	注
-u <i>name</i>	-u <i>name</i>	
-x	変更なし	
-X	変更なし	
-y <i>sym</i>	変更なし	
-z	デフォルト	-z としての SVR4 実行可能プログラムフォーマットのデマンドページ

共用ライブラリの作成

Solaris 7 で共用ライブラリを作成するには、`-G` オプションを指定する必要があります。SunOS 4.x では、`-e` オプションなしで共用ライブラリを作成することをリンカが自分で判断しますが、Solaris 7 では共用ライブラリがエントリポイントを持つ可能性があるため、このオプションは使用できなくなりました。

実行可能ファイルの作成

`-Bdynamic` と `-Bstatic` オプションはまだ利用できますが、その動作はかなり異なります。現在では、これらのオプションは実行可能バインディングではなくライブラリのインクルードを指します。実行可能バインディングは、Solaris 7 で新しい `-dy` と `-dn` オプションでのみ排他的に設定されます。`-dy` オプションがデフォルトです。これは、動的にリンクされた実行可能ファイルを作成するために必要です。`-dn` オプションは、静的にリンクされた実行可能ファイルを作成するために必要です。

`-Bdynamic` と `-Bstatic` オプションは、`-dy` オプションを使用したときだけ適用されます。`-Bdynamic` はリンクエディタに共用ライブラリを含めるように指示し、`-Bstatic` はアーカイブライブラリを含めるように指示します。これらのオプションは、次の `-Bdynamic` または `-Bstatic` オプション指定が現れるまで、`-l` 引数を管理する切り替え (トグル) として機能します。

次の例に、同様の実行可能プログラムを作成するのに使用できる SunOS 4.x と Solaris 7 コマンドを示します。

■ `sunos4.1% ld Bstatic test.o -lx`

`libx.a` を使用して、静的な実行可能ファイルを作成する。

■ `sunos5.x% ccdn test.o -lx`

`libx.a` を使用して、静的な実行可能ファイルを作成する。

■ `sunos4.1% ld Bdynamic test.o -lx`

`libx.so` を使用して、動的な実行可能ファイルを作成する。

■ `sunos5.x% cc test.o -lx`

`libx.so` を使用して、動的な実行可能ファイルを作成する。

■ `sunos4.1% ld Bdynamic test.o Bstatic -lx`

`libx.a` を使用して、動的な実行可能ファイルを作成する。

■ `sunos5.x% cc test.o Bstatic -lx`

`libx.a` を使用して、動的な実行可能ファイルを作成する。

ライブラリ検索パスの指定

SunOS 4.x では、`-L` オプションを付けて指定したディレクトリはリンク時に検索され、その情報は実行時に使用するために保持されていました。この動作は現在では、`-L` と `-R` オプションに分けられています。`-L` オプションはリンク時に検索するディレクトリを指定し、`-R` オプションはリンクに対して、実行時に使用するために保持する検索パスを指示します。詳細については、156ページの「検索パスの規則」を参照してください。

`-Bdynamic` と `-Bstatic` オプションと同様に、`-L` オプションの位置には意味があります。これは、それに続く `-l` オプションにだけに適用されます。

検索パスの規則

動的リンクと実行時リンクが SunOS 4.x リンカによって使用されたのとは異なるアルゴリズムを使って検索パスを決定します。

以下の例では、SunOS 4.x と Solaris 7 の動的リンクおよび実行時リンクの検索パスを比較します。Solaris 7 では、リンクエディタと実行時リンクの検索パスは

LD_LIBRARY_PATH 設定値の影響を受けることに注意してください。ただし、実行時リンカでは、プログラムが LD_LIBRARY_PATH を設定しないで共用ライブラリを検索できるほか、共用ライブラリのローディングがさらに効率的になります。したがって、Solaris 7 では \$ORIGIN を代用することをお勧めします。prog がインストールされた位置からの組み込みライブラリの相対パスを指定してプログラムをビルドしなければならないからです。たとえば、.../package/bin/prog は、.../package/lib/libmine.so.1 を使用します。

SunOS 4.x リンカ検索パス

- リンクエディタ: `-L, LD_LIBRARY_PATH, /usr/lib, /usr/local/lib`
- 実行時リンカ: `LD_LIBRARY_PATH, -L, /usr/lib, /usr/local/lib`

LD_LIBRARY_PATH=dirlist1 がある Solaris 7 リンカ検索パス

- リンクエディタ: `-L, dirlist1, /usr/ccs/lib, /usr/lib`
- 実行時リンカ: `dirlist1, -R, /usr/lib`

LD_LIBRARY_PATH=dirlist1, dirlist2 がある Solaris 7 リンカ検索パス

- リンクエディタ: `dirlist1, -L, dirlist2, /usr/ccs/lib, /usr/lib`
- 実行時リンカ: `dirlist1, dirlist2, -R, /usr/lib`

Solaris 7 リンカは、\$ORIGIN を使ってパスを検索します。

- 実行時リンカ: `-R, $ORIGIN/./lib`

また、Solaris 7 では、LD_LIBRARY_PATH_64 は LD_LIBRARY_PATH の 64 ビット専用バージョンです。

バージョン番号

SunOS 4.x は、共用ライブラリに対してメジャーとマイナーの両方のバージョン番号をサポートしていました。Solaris 7 は、メジャーバージョン番号だけをサポートします。バイナリ互換性のサポートについては、メジャーおよびマイナーバージョン番号は SunOS 4.x 共用ライブラリで認識されます。これらのライブラリは、SunOS 4.x ソフトウェアにあったのと同じメジャーおよびマイナーバージョン番号を保持するために必要となります。

表 15-2 は、SunOS 4.x および Solaris 7 の共用ライブラリのバージョンを示します。

表 15-2 共用ライブラリの例

SunOS 4.x	Solaris 7
libc.so.1.7	libc.so.1
libdl.so.1.0	libdl.so.1

SunOS 4.x システムソフトウェアにおいては、`-l` オプションを指定した場合、`build environment linker` はメジャーおよびマイナー番号をともに持つライブラリを検索しました。たとえば、`-ldl` を指定した場合、ライブラリ、`libdl.so.1.0` がリンクされます。Solaris 7 環境では、メジャー番号はサポートされていますが、デフォルトのリンクエディタはバージョン番号を無視します。前の例では、`build environment linker` は現在では `libdl.so` と特定のバージョンのファイルを指すシンボリックリンクを検索します。

リンカによって参照された時、デフォルトでは、動的に実行可能なオブジェクトまたは共有オブジェクト中の *dependency* レコードは関連する共有オブジェクトのファイル名です。依存性の指定をより一貫した方法にするために、共有オブジェクトは実行時に参照されるべきファイル名をそれ自身に記録することができます。これはライブラリファイルをリンクする時に `-h` オプションによって指定します。

Solaris 7 では、シンボリックリンクはほとんどのライブラリに対して作成されています。メジャー番号をつけて新しい共用ライブラリを構築し、それから最もよく使用するライブラリのバージョンを指すシンボリックリンクを作成してください。

例

新しいユーティリティの `dump(1)` (104ページの「ファイルのバックアップと復元」を参照) により、オブジェクトファイルのデバッグ、または静的および動的リンクのチェックが容易になります。`dump -L` オプションは、実行可能プログラムに含まれる実行時リンカに必要な情報を表示します。この情報は、ELF ファイルの動的セクションに含まれます。`RPATH` エントリは、`ld` の `-R` オプションにより指定された検索パスを表示します。

例を以下に示します。

- `libx.o` を使用して共用ライブラリ `libx.so.1` を構築します。
- `libx.so.1` から `libx.so` へのリンクを作成します。

- `-h` オプションで渡された情報を格納した `dump` の出力を `SONAME` フィールドを含めて示します。

```
examples% cc -G -o libx.so.1 -h libx.so.1 libx.o
examples% cp libx.so.1 /mylibs
examples% ln -s /mylibs/libx.so.1 /mylibs/libx.so
examples% dump -Lv libx.so.1
```

```
libx.so.1:
**** DYNAMIC SECTION INFORMATION ****
.dynamic :
[INDEX] Tag      Value
[1]     INIT      0x3b8
[2]     FINI      0x3f4
[3]     SONAME    libx.so.1
[4]     HASH      0x94
[5]     STRTAB   0x33c
[6]     SYMTAB   0x14c
[7]     STRSZ    0x62
[8]     SYMENT   0x10
[9]     PLTGOT   0x10404
[10]    PLTSZ    0xc
[11]    PLTREL   0x7
[12]    JMPREL   0x3ac
[13]    RELA     0x3a0
[14]    RELASZ   0x18
[15]    RELAENT  0xc
```

ライブラリが他の動的ライブラリを必要とするときは、次の例に示すように、`RPATH` と共に動的ライブラリを指定するようにします。

次の例では `prog.c` をコンパイルし、(前の例で構築された) `libx.so` を動的にリンクし、バイナリが実行のためカレントディレクトリ情報を保持するように指定します。この例は、コンパイル済みプログラムの `prog.c` についての `dump` 出力を示します。ここで、前の例の `SONAME` フィールドに格納された情報は、`prog` により `NEEDED` として示されます。`prog` が実行されると、`libx.so.1` は、`libx.so` でも、異なるバージョンにリンクされます。

```
examples% cc -o prog prog.c -L/mylibs -R/mylibs -lx
example% dump -Lv prog
```

```
prog:
**** DYNAMIC SECTION INFORMATION ****
.dynamic :
[INDEX] Tag      Value
[1]     NEEDED   libx.so.1
[2]     NEEDED   libc.so.1
[3]     INIT      0x1b1ac
[4]     FINI      0x1b248
[5]     RPATH     /mylibs
[6]     HASH      0x100e8
[7]     STRTAB   0x17f90
```

```
[8] SYMTAB 0x12be0
[9] STRSZ 0x31e1
[10] SYMENT 0x10
[11] DEBUG 0x0
[12] PLTGOT 0x2b25c
[13] PLTSZ 0x30
[14] PLTREL 0x7
[15] JMPREL 0x1b180
[16] RELA 0x1b174
[17] RELASZ 0x3c
[18] RELAENT 0xc
```

デバッグ

この節ではデバッグツールの変更について説明します。

dbx と dbxtool

dbx と dbxtool は、デフォルトのシステムソフトウェアには含まれません。アンバンドル製品の Sun WorkShop には、これらの改良版が含まれています。

adb と kadb

adb と kadb は、Solaris 7 オペレーティングシステムにバンドルされていて、SunOS 4.x のツールと同じ機能を提供します。kadb はマルチプロセッサで使用できるよう改良されました。kadb のプロンプトにはプロセッサ ID が表示されます。以下の例では、プロセッサ ID が 0 になっています。

Solaris 7 環境で容易にカーネルのデバッグを行うには、次のようにします。

- savecore を使用可能にします (/etc/init.d/sysetup ファイルの savecore 行のコメントを解除します)。
- kadb 下でブートします (システムクラッシュ時に \$c と入力します)。
- adb と crash を使用します。

adb は、64 ビット用として次のように拡張されています。

- `?`、`/`、`=` 修飾子に対応する拡張形式文字。K は、`long` 型またはポインタを 16 進形式で出力するために使用されます (32 ビットプログラムでは 4 バイト、64 ビットプログラムでは 8 バイトを表示)。
- 64 ビット SPARC マクロのパス: `/usr/lib/adb/sparcv9` と `/usr/platform/platformname/lib/adb/sparcv9`。

kadb マクロ

以下の `kadb` マクロは、マルチスレッドカーネルといっしょに使用すると特に有効です。

現在のスレッドを表示します。現在のスレッドポインタは、SPARC グローバルレジスタ `g7` です。

```
kadb[0]: <g7$<thread
```

`threadlist` は、システム内のすべてのカーネルスレッドのスタックトレースを表示します。このリストは非常に長くなることがあります。

```
kadb[0]: $<threadlist
```

`mutex` は、所有スレッドのアドレスを表示します。この例では、グローバルで危険なドライバ `mutex` を使用しています。

```
kadb[0]: unsafe_driver$<mutex
```

```
kadb[0]: moddebug/W 0x80000000
```

`moddebug` は、モジュールのロードを監視できるようにします。デバッグ専用を使用する `moddebug` の有効な値については、`<sys/modctl.h>` の最後を参照してください。

動作中のカーネルのデバッグ

稼働中のカーネルをデバッグするには、次のコマンドを使用します。

```
# adb -k /dev/ksyms /dev/mem
```

/dev/ksyms は、稼働中のカーネルの完全な名前を含む擬似デバイスです。

truss コマンド

truss は、実行したシステムコール、受信シグナル、ハードウェア障害などを追跡するために開発された新しいユーティリティです。truss には、 エントリを有効にして追跡対象のプロセスで実行されたユーザーレベルの関数呼び出しを終了するオプションのほか、フォークされたプロセスの追跡やマルチスレッドプロセスの処理のように SunOS 4.x の trace(1) コマンドにない大幅な改良も加えられています。

また、truss は、プロセスのシステムコール、シグナル、ハードウェア障害を追跡します。このユーティリティには、エントリを有効にして、追跡対象のプロセスで実行されたユーザーレベルの関数呼び出しの追跡を終了する新しいオプションが追加されています。

次の例は、date コマンドの追跡結果を要約したものです。-c オプションを指定すると、truss は行単位の追跡を表示せず、システムコール、シグナル、フォルトの回数をカウントして、その合計を表示します。

```
example% truss -c date
Fri Sep 18 14:31:30 PDT 1992
syscall      seconds    calls  errors
__exit          .00         1
read            .00         7
write           .00         1
open            .03        12
close           .00        12
time            .00         1
brk             .01         4
lseek           .00         1
fstat           .00         4
ioctl           .00         1
execve          .00         1
mmap            .01        17
munmap          .00         8
-----
sys totals:    .05         70         0
usr time:      .03
elapsed:      .28
```

truss オプションの詳細については、truss(1) のマニュアル・ページを参照してください。Solaris 7 ではこの他に pmap(1) のような proc(4) を基本としたデバッグツールが数多く用意されています。

ツールと資源

この章では、開発環境におけるツールと資源の変更について説明します。この章の内容は次のとおりです。

- 163ページの「`ioctl()` 要求」
- 166ページの「`ptrace()` 要求値」
- 168ページの「ライブラリ」
- 171ページの「`make` の使用」
- 172ページの「SCCS の使用」
- 172ページの「アプリケーション互換性の判断」
- 173ページの「アプリケーションパッケージ作成」
- 175ページの「ツールキット」
- 176ページの「SunOS 4.x ツールの検索」

`ioctl()` 要求

`dkio(7I)`、`filio`、`mtio(7I)`、`sockio(7I)`、`streamio(7I)`、`termio(7I)`、`termios(7I)` に関連するすべての `ioctl` は、Solaris 7 でサポートされます。

SunOS 4.x の `termios` 構造体と Solaris 7 の `termios` 構造体との間に、互換性のない部分がいくつかあります。例えば SunOS 4.x の `termios` 構造体にはある `c_line` フィールドが、Solaris 7 には含まれていません。

`<sys/ttold.h>` に定義がある次の `ioctl` は、実装されていません。

- TIOCMODG
- OTTYDISC
- TABLDISC
- KBLDISC
- TIOCMIDS
- TIOCSETX
- NETLDISC
- NTABLDISC
- TIOCGETX
- NTTYDISC
- MOUSELDISC

次の `ttycom ioctl` 要求は Solaris 7 にはありません。

- TIOCSCTTY
- TIOCNOTTY
- TIOCISPACE
- TIOCPKT
- TIOCGETPGRP
- TIOCISIZE
- TIOCUCNTL
- TIOCOUTQ
- TIOCTCNTL
- TIOCCONS

Solaris 7 でサポートされる `ioctl()` を表 16-1 に示します。

表 16-1 ioctl() のサポート

ioctl()	説明
DKIOCGPART	これらの要求は、Solaris 7 では DKIOCGAPART と DKIOCSAPART に置き換えられる。
DKIOCGCONF	この要求は、Solaris 7 では、SunOS 4.x の DKIOCGCONF と DKIOCINFO 構造体の情報を合わせたものが含まれる DKIOCINFO に置き換えられる。
DKIOCSCMD	この要求は、IPI ドライブに対してのみ正常に実行される。この ioctl は、SCSI デバイスでは異常終了する。SCSI デバイスに対しては、USCSI ioctl を使用する。
DKIOCGLOG	EINVAL が戻される。DKIOCWCHK コマンドは、フロッピーデバイスに対する書き込みチェックを切り換える。
filio	次の filio ioctl 要求は、Solaris 7 または SVR4 ではサポートされていない：FIOSETOWN、FIOGETOWN、FIOCLEX、FIONCLEX、filio。ioctl 要求は ABI または SVID では、定義されていない。
mtio	Solaris 7 では mtio ioctl 要求のすべてをサポートしていないデバイスもある。マニュアルページの mtio(7) を参照のこと。
sockio	次の sockio ioctl 要求は、SVR4 と Solaris 7 において実装されている：SIOCSPGRP、SIOCGPGRP、SIOCATMARK。sockio ioctl 要求は ABI または SVID では、定義されていない。
streamio	すべての SunOS 4.x の streamio ioctl 要求は、Solaris 7、ABI、SVID、および SVR4 に実装されている。I_FDINSERT 要求は、strfdinsert 構造体を指す引数が必要となる。SunOS 4.x の strfdinsert 構造体は fd (int) フィールドを含むが、ABI、SVID、または SVR4 strfdinsert 構造体は fildes (int) フィールドを含む。

表 16-1 ioctl() のサポート 続く

ioctl()	説明
audioio	SunOS 4.x の <sun/audioio.h> ファイルは、Solaris 7 では <sys/audioio.h> に移動されている。さらに、Solaris 7 では、インタフェースの機能が強化されている。詳細については、audio(7)、audioamd(7)、dbri(7) マニュアルページを参照のこと。
termio, termios	すべての SunOS 4.x の termio、および termios ioctl 要求は、Solaris 7、ABI、SVID、および SVR4 に実装される。SunOS 4.x termios 構造体と Solaris 7、あるいは ABI、SVID、または SVR4 termios 構造体との間には一部互換性がない。SunOS 4.x の termios 構造体には、c_line フィールドがある。c_cflag (端末のハードウェア制御) の内容は SunOS 4.x ソフトウェアでは CRTSCTS (RTS/CTS フロー制御を有効にする) が可能だが、この値は Solaris 7 リリース、ABI、SVID、または SVR4 には定義がない。しかし、機能性は termiox(7) インタフェースを通してサポートされる。

ptrace() 要求値

ptrace() 機能は /proc の先頭に実装されています。新しいアプリケーションは直接 proc(4) を使用してください。

Solaris 7 では、ptrace() ルーチンは BCP モードで実行するアプリケーションをサポートするために単独で存在します。ptrace() ルーチンは要求値として整数 1~9 を使用しますが、SunOS 4.x ルーチンは <sys/ptrace.h> で要求値をシンボリック定数として定義します。次のシンボリック定数は、Solaris 7 と互換性があります。

- PTRACE_TRACEME
- PTRACE_PEEKTEXT
- PTRACE_PEEKDATA
- PTRACE_PEEKUSER
- PTRACE_POKETEXT
- PTRACE_POKEDATA
- PTRACE_POKEUSER
- PTRACE_CONT

- PTRACE_KILL
- PTRACE_SINGLESTEP

SunOS 4.x の PTRACE_CONT *addr* 引数は、中断しているプロセスが実行を再開すべき場所を指定します。ただし、プロセスが中断したところから実行が再開する *addr* = 1 の場合は除きます。Solaris 7 は、*addr* が常に 1 に等しく、実行が常にプロセスが中断したところから再開することを要求します。また、データによって指定されたものを除いたすべてのペンディングシグナルをキャンセルします。SunOS 4.x の PTRACE_CONT は、ペンディングシグナルをすべてキャンセルするとは限りません。

表 16-2 に示す SunOS 4.x の有効な要求は、Solaris 7 `ptrace()` ルーチンではサポートされません。

表 16-2 Solaris 7 でサポートされていない `ptrace()` 要求

PTRACE_ATTACH	PTRACE_GETWINDOW
PTRACE_DETACH	PTRACE_SETWINDOW
PTRACE_GETREGS	PTRACE_22
PTRACE_SETREGS	PTRACE_23
PTRACE_GETFPREGS	PTRACE_26
PTRACE_SETFPREGS	PTRACE_27
PTRACE_READDATA	PTRACE_28
PTRACE_WRITEDATA	PTRACE_SYSCALL
PTRACE_READTEXT	PTRACE_DUMPCORE
PTRACE_WRITETEXT	PTRACE_SETWRBKPT
PTRACE_GETFPAREGS	PTRACE_SETACBKPT
PTRACE_SETFPAREGS	PTRACE_CLRDR7

ライブラリ

Solaris 7 は、System V Inrerface Definition, Third Edition (SVID 3) に準拠しています。SunOS 4.1 System V ライブラリとともに書かれたプログラムは Solaris 7 への移植が簡単ですが、SunOS 4.x BSD C ライブラリを使用するプログラムにとっては多くの労力を必要とします。

再編成ライブラリ

機能や機能グループの中には、Solaris 7 では別のライブラリに移動されたものがあります。このため、SunOS 4.x のアプリケーションを Solaris 7 でコンパイルする時に、これらの移動された機能を参照することで未定義とフラグされる可能性があります。

コンパイルした後、未定義とフラグされた任意の機能のマニュアルページを確認してください。機能説明のところに、`-l` リンカーオプションと、シンボルを解決する必要のある任意のインクルードファイルの両方がリストされます。

共用ライブラリ

共用ライブラリは現在、マイナーバージョン番号をサポートしません。

共用初期設定データファイル (`.sa`) はすでに不要となっており、`.sa` ファイルは Solaris 7 では提供されません。

資源の制限

Solaris 7 での資源の制限は大きく異なります。前リリースでは、静的テーブルの割り当てがファイル記述子やアクティブなプロセスなどの資源に使用されました。これらの資源は、現在は動的に割り当てられます。つまり、空いている物理メモリによって制限されることを意味します。表 16-3 に資源の制限を示します。

表 16-3 資源の制限

構成	制限
RLIMIT_CORE	プロセスによって作成できるコアファイルの最大サイズ (バイト単位)
RLIMIT_CPU	プロセスが使用できる最大 CPU タイム (秒単位)
RLIMIT_DATA	プロセスのヒープの最大サイズ (バイト単位)
RLIMIT_FSIZE	プロセスによって作成できるファイルの最大サイズ (バイト単位)
RLIMIT_NOFILE	プロセスによって作成できるファイル記述子の最大数より 1 大きい値
RLIMIT_VMEM	プロセスのマップされたアドレスサイズが増大できる最大サイズ (バイト単位)
RLIMIT_STACK	プロセスのスタックの最大サイズ (バイト単位)

注 - ネットワークライブラリを必要とする共有オブジェクトはすべて動的にリンクしなければなりません。ネットワークライブラリは、libdl.so.1 を必要とし、アーカイブライブラリは利用できません。

表 16-4 に SunOS 4.x ライブラリと Solaris 7 ライブラリ、およびそれらの位置を示します。

表 16-4 ライブラリ位置の比較

ライブラリ名	SunOS 4.x ディレクトリ	Solaris 7 ディレクトリ
libbsdmalloc.a	/usr/lib	/usr/lib
libc.a	/usr/lib および /usr/5lib	/usr/lib
libc.so.1.7	/usr/lib	/usr/lib
libc.so.2.7	/usr/5lib	/usr/lib
libc_p.a	/usr/5lib	なし

表 16-4 ライブラリ位置の比較 続く

ライブラリ名	SunOS 4.x ディレクトリ	Solaris 7 ディレクトリ
libcurses.a	/usr/lib および /usr/5lib	/usr/ucblib および /usr/ccs/lib
libcurses_p.a	/usr/5lib	なし
libdbm.a	/usr/lib	/usr/ucblib
libdl.so.1.0	/usr/lib	/usr/lib
libg.a	/usr/lib	なし
libkvm.a	/usr/lib	なし
libkvm.so.0.3	/usr/lib	/usr/lib
libl.a	/usr/lib	/usr/ccs/lib
libln.a	/usr/lib	なし
liblwp.a	/usr/lib	なし
libm.a	/usr/lib	/usr/lib および /usr/lib/libp
libmp.a	/usr/lib	/usr/lib
libnbio.a	/usr/lib	なし
libnsl.a	/usr/lib	/usr/lib
libpixrect.a	/usr/lib	なし
libpixrect.so.2.14	/usr/lib	なし
libposix.a	/usr/lib	なし
libresolv.a	/usr/lib	/usr/lib

表 16-4 ライブラリ位置の比較 続く

ライブラリ名	SunOS 4.x ディレクトリ	Solaris 7 ディレクトリ
librpsvc.a	/usr/lib	/usr/lib
libsuntool.so.0.54	/usr/lib	なし
libsunwindow.so.0.55	/usr/lib	なし
libsvdm.a	/usr/5lib	なし
libsvdm_p.a	/usr/5lib	なし
libtermcap.a	/usr/lib および /usr/5lib	/usr/ucblib および / usr/ccs/lib
libterm.lib.a	/usr/lib および /usr/5lib	/usr/ccs/lib
libxgl.so.1.1	/usr/lib	/opt/SUNWits/ Graphics-sw/xgl/lib
libxpg.a	/usr/xpg2lib	なし
liby.a	/usr/lib および /usr/5lib	/usr/ccs/lib

make の使用

Solaris 7 で利用できる make ユーティリティは 2 種類あります。デフォルトである /usr/ccs/bin/make は、SunOS 4.x の make コマンドと同じです。SVR4 版は /usr/ccs/bin/make で利用できます。

デフォルトの `make` を使うと、`Makefile` を変更する必要はありません。ただし、`Makefile` で使用するコマンドのいくつかは変更されている可能性があります。たとえば、`Makefile` で一般に使用される `install(1)` は、オプションに変更が加えられたために次の例のように予期しない結果を生むことがあります。

■ SunOS 4.x `Makefile` - `install`:

```
install -o bin -g bin -m 444 target.c /usr/bin/target
```

■ SunOS 5.6 `Makefile` - `install`:

```
install -u bin -g bin -m 444 target.c /usr/bin/target
```

`/usr/ueb` にある `install(1B)` のバージョンは SunOS 4.x のバージョンと互換性があります。

個々のインタフェースに関する情報については、付録 A の互換性に関する表で確認してください。

SCCS の使用

Solaris 7 のソースコード管理システム (SCCS) は SunOS 4.x の SCCS とは少し異なっています。コマンドとサブコマンドの同じセットが両方の環境でサポートされています。SunOS 4.x システムで使用される SCCS ディレクトリおよび `s.files` は Solaris 7 システムでも同様に動作します。

SunOS 4.x ソフトウェアでは、SCCS コマンドは `/usr/sccs` ディレクトリに置かれていました。Solaris 7 ではこれらのコマンドは他のプログラミングツールとともに `/usr/ccs/bin` に置かれています。

SunOS 4.x と Solaris 7 ユーティリティの相違の 1 つに読み取り不可能な `s.file` の処理があります。SunOS 4.x コマンドは、読み取り不可能な `s.file` が出現すると、エラーを出力して続行します。Solaris 7 コマンドはエラーを無視します。

アプリケーション互換性の判断

バイナリ互換パッケージは開発環境としては提供されていませんが、将来のリリースとのバイナリ互換性を改善できる適切なプログラミングが必要です。

バイナリ互換パッケージは、部分的に静的にリンクされているかあるいは動的にリンクされているハイブリッドと同じく、動的または静的にリンクされたアプリケーションと互換性があります。

バイナリ互換パッケージは、「お行儀の良い」ユーザアプリケーションで使用することができます。「お行儀の良い」アプリケーションとは、次の条件を満たすアプリケーションを指します。

- 直接カーネルにトラップしない
- どのシステムファイルにも直接書き込まない
- /dev/kmem、 /dev/mem または libkvm を使用しない
- 公表されていない SunOS インタフェースを使用しない
- カスタマ供給のドライバに依存しない

上記の条件を満たしていないと、アプリケーションは予想できない結果を生じることがあります。

バイナリ互換パッケージの使用方法に関する情報は、『バイナリ互換性ガイド』に説明があります。

アプリケーションパッケージ作成

Solaris 7 環境は現在パッケージという単位でバンドルされています。これらのパッケージには、システムに追加したり、システムから削除したりする必要があるファイルおよび情報のすべてが含まれます。

パッケージは次のようなコンポーネントで構成されます。

- `pkginfo` ファイル – これはパッケージの特性を設定する ASCII ファイルです。パッケージを記述してそのインストールのための制御パラメータを設定する `macro=value` の対で構成されています。詳細については、`pkginfo(4)` のマニュアルページを参照してください。
- `prototype` ファイル – これはパッケージの内容を定義する ASCII ファイルです。そのまま出荷できるオブジェクト (たとえば、ファイル、ディレクトリ、リンク) ごとに 1 つのエントリがあります。また、`pkginfo`、`depend` および `copyright` ファイルなどのパッケージ情報ファイルおよびスクリプトのインストールエントリを含みます。詳細については、`prototype(4)` のマニュアルページを参照してください。

- `copyright`ファイル – これはパッケージの著作権に関する記述を提供する ASCII ファイルです。コメント行を含むその内容はパッケージのインストール時に表示されます。詳細については、`copyright(4)` のマニュアルページを参照してください。
- パッケージの内容 – パッケージの内容が含まれます。
- スクリプト – スクリプトを使用してパッケージのインストールおよび削除を制御したり、ユーザに入力を要求したり、または特定のクラスのすべてのオブジェクトに対して処理を実行できます。スクリプトは Bourne シェルが実行できるものでなければなりません。

アドオンアプリケーションソフトウェアは、フロッピーディスク、テープ、または CD-ROM から Solaris 7 システムにインストールできるようにパッケージ化されていなければなりません。『*Application Packaging Developer's Guide*』では、パッケージを作成するためのガイドラインを記載しています。

パッケージ作成ユーティリティ

パッケージを作成し、操作するためのユーティリティがいくつか提供されます。表 16-5 にパッケージの作成に便利なコマンドを示します。

表 16-5 パッケージ作成用コマンド

<code>pkgproto</code>	<code>pkgmk</code> コマンドへ入力するプロトタイプファイルのエントリを生成する
<code>pkgmk</code>	インストール可能なパッケージを生成する
<code>pkgtrans</code>	パッケージフォーマットを変換する

表 16-6 にパッケージの追加と削除に便利なコマンドを示します。

表 16-6 パッケージの追加と削除用コマンド

pkgadd	システムにソフトウェアパッケージを追加する
pkgask	要求スクリプトに対する応答を格納する
pkgrm	システムからパッケージを削除する
pkgchk	インストールの結果をチェックする

表 16-7 にパッケージに関する情報を提供するコマンドを示します。

表 16-7 Cパッケージに関する情報を提供するコマンド

pkginfo	インストール済みパッケージに関するソフトウェアパッケージ情報を表示する
pkgparam	パッケージパラメータ値を表示する

ツールキット

この節では OPEN LOOK Intrinsic ToolKit (OLIT) と XView について説明します。

OLIT

OPEN LOOK Intrinsic Toolkit (OLIT) は Xt Intrinsic をベースにしています。このツールキットは多くのウィジェットセットに共通な関数セットを提供し、X 環境のユーザインタフェースコンポーネントを作成したり、流用したり、削除したりします。

XView

XView Window Toolkit は OPEN LOOK グラフィカルユーザインタフェース (GUI) 仕様を実装しています。

XView は `varargs` に基づく可変長の属性値リストを使用し、ウィンドウ、メニュー、およびスクロールバーなど、作成するオブジェクトを指定します。このツールキットでは、すでに通常の動作が定義されているため、手続き型プログラミングによくあるように定型のコードを繰り返す必要がありません。

SunOS 4.x ツールの検索

ほとんどの SunOS 4.x のプログラミングツールが利用でき、同じ機能を提供しますが、多くのものが新しい位置にあります。現在バンドルされるプログラミングツールはすべて 2 つのディレクトリ、`/usr/ccs/bin` と `/usr/ccs/lib` にあります。表 16-8 にプログラミングツールと SunOS 4.x の位置を示します。

表 16-8 バンドルされるプログラミングツール

SunOS 4.x コマンド	SunOS 4.x での位置
<code>admin</code>	<code>/usr/sccs</code>
<code>ar</code>	<code>/usr/bin</code>
<code>as</code>	<code>/usr/bin</code>
<code>cdc</code>	<code>/usr/sccs</code>
<code>comb</code>	<code>/usr/sccs</code>
<code>cpp</code>	<code>/usr/lib/cpp</code>
<code>delta</code>	<code>/usr/sccs</code>
<code>error</code>	<code>/usr/ucb</code>
<code>get</code>	<code>/usr/sccs</code>
<code>help</code>	<code>/usr/sccs</code>
<code>ld</code>	<code>/usr/bin</code>

表 16-8 バンドルされるプログラミングツール 続く

SunOS 4.x コマンド	SunOS 4.x での位置
lex	/usr/bin
lorder	/usr/bin
m4	/usr/bin
make	/usr/bin
nm	/usr/bin
prof	/usr/bin
prs	/usr/sccs
prt	/usr/sccs
ranlib	/usr/bin
rmdel	/usr/sccs
sact	/usr/sccs
sccs	/usr/ucb
sccsdiff	/usr/sccs
size	/usr/bin
strip	/usr/bin
symorder	/usr/ucb
tsort	/usr/bin
unget	/usr/sccs
unifdef	/usr/ucb

表 16-8 バンドルされるプログラミングツール 続く

SunOS 4.x コマンド	SunOS 4.x での位置
val	/usr/sccs
vc	/usr/old
what	/usr/sccs
yacc	/usr/bin
yaccpar	/usr/lib

表 16-9 に、新しい Solaris プログラミングツールとその説明を示します。

表 16-9 新しいプログラミングツール

新しいコマンド	説明
dis	COFF のオブジェクトコード逆アセンブラ
dump	オブジェクトファイルの選択された部分をダンプする
exstr	ソースファイルから文字列を抽出する
mcs	オブジェクトファイルのコメントセクションを操作する
regcmp	正規表現コンパイラ
truss	システムコールとシグナルを追跡する
ptools	多方面の /proc ユーティリティ

表 16-10 に、現在アンバンドル製品である SunOS 4.x コマンドを示します。

表 16-10 アンバンドル製品のプログラミングツール

アンバンドルのコマンド	説明
cb	簡単な C プログラム整形ツール
cc	C コンパイラ
cflow	プログラムにフローグラフを生成する
cscope	対話方式で C プログラムを検査する
ctrace	C プログラム実行追跡を行う
cxref	C プログラムクロスリファレンスを行う
dbx	ソースレベルデバッガ
dbxtool	ウィンドウベースのソースレベルデバッガ
gprof	call-graph プロファイルデータを表示する
indent	C プログラムソースファイルをインデントおよびフォーマットする
inline	インラインのプロシージャコールの展開
lint	C プログラムベリファイア
objdump	COFF オブジェクトファイルの選択された部分をダンプする
tcov	test coverage 解析および文単位のプロファイルを構築する

ネットワークと国際化機能

この章では、プログラミング環境に関連する Solaris 7 のネットワーク機能と、改善された国際化機能について説明します。

- 181ページの「ネットワーク」
- 182ページの「国際化」

ネットワーク

Solaris 7 には、次のネットワーク機能があります。

- ファイルシステムユーティリティを集中制御する分散ファイルシステム(DFS)
- NFS を含んだネットワーク情報サービスプラス (NIS+)
- ネームサービススイッチファイル

これらのサービスについての詳細は、『NIS+ への移行』と『NFS の管理』を参照してください。

NIS と NIS+

Solaris 7 は、ネットワーク情報サービス (NIS)、SunOS 4.x のネームサービス、ネットワーク情報サービスプラス (NIS+)、異機種分散システムの企業ネームサービスをサポートしています。Solaris 7 で使用できる NIS サポートについての詳細は、136 ページの「NIS+」を参照してください。

NIS+ は、名前空間オブジェクトの詳細なモデル、改善されたセキュリティ、NIS より高速な更新処理などを提供します。

NIS+ のプログラマインタフェースについては、『*SunOS Reference Manual*』のセクション 3N 「Library Routines」を参照してください。

nsswitch.conf ファイル

nsswitch.conf ファイルは、ネームサービス管理を簡略化するために設計されました。アプリケーションは、nsswitch.conf ファイルを使用してネームサービスを選択できます。これにより、ネームサービス情報をネットワークサービス内で直接定義する必要がなくなりました。nsswitch.conf ファイルの書式についての詳細は、nsswitch.conf(4) のマニュアルページを参照してください。

Network Interface Tap

SunOS 4.x で提供されていた Network Interface Tap (NIT) は Solaris 7 では必要なくなりました。Solaris 7 では、イーサネットドライバが真の STREAMS ドライバに変更されたので、ドライバを直接オープンして通信できます。

pfmod(7M)、bufmod(7M)、dlpi(7P) のマニュアルページを参照してください。

Solaris 7 のイーサネットドライバとその他のデータリンクドライバは、コネクションレスの Data Link Provider Interface (DLPI) バージョン 2 をサポートしています。

ソケット

ソケットは Solaris 7 でサポートされています。SunOS 4.x と違って、ソケットはカーネルの中にはまったく実装されなくなり、ライブラリ libsocket として STREAMS 上に実装されています。

国際化

Solaris 7 での変更のほとんどは以前の国際化機能の改善です。国際化サポートに関する詳細な情報については、『プログラミングの国際化』を参照してください。

プログラムの国際化に関わるアプリケーション開発者は次のガイドラインに従ってください。

- `setlocale(3C)` を呼び出して `LANG` 環境変数を設定する
- 標準のコードセットを使用し、8 ビット境界を守る
- `strftime(3C)` を使用して日付および時刻を出力する
- ユーザが照合のために判読できるように `strcmp(3)` を `strcoll(3C)` に置き換える
- `gettext(3C)` もしくは `catget(3C)` を呼び出してロケール固有のメッセージカタログから翻訳された文字列を検索する

文字サポート

Solaris 7 環境は拡張 UNIX コード (EUC)、VTF8、PCK、V165 をサポートしています。これにより、1 つのシステムで複数バイトと複数のコードセットを利用できます。

SunOS 4.x は ASCII 以外の文字のシングルバイト表現をサポートしていません。Solaris 7 では、複数バイト表現がサポートされています。このサポートは数千文字もあるアジア系言語の文字セットに必要です。

`libc` に含まれる複数バイトライブラリには次のような機能があります。

- 複数バイト文字からワイド文字への文字変換
- ワイド文字の標準入出力
- ワイド文字の分類
- ワイド文字のフォーマット化

Solaris 7 は複数バイトファイル名をサポートしていますが、ログイン名とマシン名は ASCII 文字に制限するようにしてください。

メッセージカタログ

SunOS 4.x のメッセージカタログのサポートは Solaris 7 で強化され、複数バイト文字を使ってメッセージカタログを作成できるようになりました。

メッセージカタログを使うと、アプリケーションはアプリケーションが実行された母国語で実行時のメッセージを表示できます。これらのメッセージカタログは、言語ロケールによって指定される母国語用にはじめに作成しなければなりません。

ロケールデータベース

SunOS リリース 5.6 のロケールデータベース (`/usr/lib/locale/locale`) は、SunOS 5.x のロケールデータベースとは全く異なります。ただし、ユーザ側からは違いは分かりません。

コマンド

Solaris 7 のほとんどのシステムコマンドはメッセージ化されました。これらコマンドの多くには複数バイト機能があります。つまり、複数バイト文字表現が可能になっています。より多くのコマンドがメッセージ化されたことにより、ローカリゼーションの労力は軽減されます。

`installtxt(1)` コマンドは `msgfmt(1)` に変更されました。メッセージを抽出するには新しい `xgettext(1)` コマンドを使用します。

`strftime(3C)` を変更すると、日付および時刻フォーマットに影響を与えます。`date(1)` コマンドの出力フォーマットに依存するシェルプログラムは、新しいフォーマットを処理できるように修正しなければなりません。

`chrtbl(8)` と `catdef(8)` は、`localedef(1)` に置き換えられました。

ライブラリ

`/usr/xpg2lib/libxpg2.a` アーカイブライブラリは利用できません。これらのルーチンは、`libc` に入りました。

表 17-1 にこれらのインタフェースの新しい位置を示します。

表 17-1 xpg2lib ライブラリルーチンの位置

ルーチン	Solaris 7 での位置
<code>bindtextdomain</code>	<code>/usr/lib/libc</code>
<code>chroot</code>	<code>/usr/lib/libc</code>
<code>catgets</code>	<code>/usr/lib/libc</code>
<code>dgettext</code>	<code>/usr/lib/libc</code>

表 17-1 xpg2lib ライブラリルーチンの位置 続く

ルーチン	Solaris 7 での位置
getcwd	/usr/lib/libc
getut	/usr/lib/libc
l3tol	未サポート
logname	/usr/lib/libc
malloc	/usr/lib/libc
swab	/usr/lib/libc
langinfo	/usr/lib/libc
gettext	/usr/lib/libc
sbrk	/usr/lib/libc
textdomain	/usr/lib/libc

これらのルーチンを使用するプログラムは `-lxpg2` を C コンパイラに渡す必要はありませんが、`libintl.h` を含む必要があるものが現在あります。(これらのルーチンについては、表 17-1 を参照してください)。

`catgetmsg(3C)` ルーチンは利用できません。

`setlocale(3C)` によって戻される文字列におけるロケールカテゴリの順位は、SunOS 4.x と Solaris 7 では異なります。この文字列は通常 `setlocale(3C)` への次の呼び出しによって使用され、順位は問題とされません。アプリケーションはロケールカテゴリの特定の順位に依存しないようにしてください。

システムとデバイスの構成

オペレーティングシステムのカーネルとそのインタフェースは大幅に変更されています。SunOS 4.x のデバイスドライバは、バイナリ互換を提供していません。この章では、カーネルおよびシステム開発者に影響を与える Solaris 7 の変更点について説明します。

- 187ページの「システム構成」
- 191ページの「再構成ブート」
- 191ページの「開発者に関するデバイスの命名規則」

システム構成

システム構成の変更点には、動的にロード可能なカーネルとカーネルの配置、config コマンドと boot コマンド、/etc/system ファイルがあります。

動的にロードされるカーネル

以前の SunOS リリースと異なり、Solaris 7 のカーネルは動的に構成されます。現在のカーネルは小さな静的コアと動的にロードできる多くのカーネルモジュールで構成されます。ドライバ、ファイルシステム、STREAMS モジュール、またその他のモジュールは、ブート時または実行時に、必要に応じて自動的にロードされます。これらのモジュールは使用されなくなるとアンロードされます。モジュールは、そのメモリ領域が必要になるまで、メモリ内に維持されます。modinfo(1M) は、現在システムにロードされているモジュールに関する情報を提供します。

modload(1M) コマンドと modunload(1M) コマンドは、Solaris 7 ではまだ使用できませんが、動作が異なります。Solaris 7 では、これらのコマンドの使用方法に制限があり、ロード可能なドライバをシステムに正しくインストールするには不十分です。modunload は現在アンロード可能な (ビジー状態ではない) モジュールをすべてアンロードする機能が含まれます。次のように modunload を使用してください。

```
# modunload -i 0
```

カーネルの配置

以前は 1 つのファイル /vmunix にあったカーネルの内容は、現在ではディレクトリ階層の複数のモジュールに別れています。デフォルトでは、ディレクトリ階層は /platform/'uname -i'/kernel、/kernel、/usr/kernel です。

モジュールに対するディレクトリ検索パスは、/etc/system ファイルの moddir 変数により設定できます。system(4) のマニュアルページを参照してください。通常、最初にロードされるのは /platform/'uname -i'/kernel/unix です。kernel(1M) のマニュアルページを参照してください。

config コマンド

SunOS 4.x リリースでは、config コマンドを使用して、/vmunix がオブジェクトファイルから再リンクできるようにシステム構成ファイルを生成しました。次の Solaris 7 の機能により、このコマンドは必要なくなります。

- ロード可能モジュール
- /etc/system ファイル (system(4) のマニュアルページを参照)
- OpenBoot PROM (OBP) からのデバイスツリー情報
- /kernel/drv と /usr/kernel/drv にある driver.conf ファイル

/etc/system ファイル

システム構成情報は、現在 /etc/system ファイルに設定されています。また、このファイルはロード可能なモジュールのカーネルの処理方法も変更します。このファイルには、次の形式のコマンドが含まれます。

```
set parameter=value
```

たとえば、SunOS 4.x ソフトウェアにおいて、MAXUSERS は config(8) を使用して設定されました。Solaris 7 では、/etc/system ファイルの中の次のような行により設定されます。

```
set maxusers = number
```

ロード可能なモジュールに影響を与えるコマンドは、次の形式になります。

```
set module:variable=value
```

/etc/system ファイルに対して行われた変更は、システムをリブートする際に影響を与えます (system(4) のマニュアルページを参照)。

boot コマンド

Solaris 7 では、次のブートプログラムが使用できます。

- ufsboot - ディスクまたは CD からブートする
- inetboot - ネットワークからブートする

ディスクからブートする場合、PROM は、一次ブートブロックがローカルディスクのブロック 1 から 15 にあるものと仮定とします。installboot(1M) を使用し、次のようにブートブロックを作成します。

```
# installboot /usr/platform/'uname -i'/lib/fs/ufs/bootblk \  
/dev/rdisk/c0t3d0s0
```

システムファームウェアは、一次ブートストラップ (ブートブロック) プログラムをメモリにロードし、それを実行します。ブートブロックは、UFS ファイルシステムを読み取るプログラムで、二次ブートプログラム (/platform/'uname -i'/ufsboot) をメモリにロードします。

ufsboot は /kernel/unix をロードします。それから /kernel/unix は、ルートファイルシステムのマウントが可能となるまで、ufsboot を使って /kernel ディレクトリ階層からモジュールをロードします。

これらの動作の間、ブートブロックと ufsboot は、ファームウェアによって提供されるドライバを使用します。ufsboot またはブートブロックのいずれにも、ドライバコードはまったく含まれません。ufsboot が SBus カード PROM ドライバを使用するため、ufsboot コードを変更して新しいディスクタイプで新しい SBus カードを取り込む必要はありません。

ネットワークを通してブートする場合、ブートプログラムは SunOS 4.x ソフトウェアのディスクレスブートと同じように実行されます。ただし、現在、ブートプログラムは inetboot と呼ばれ、クライアントの vfstab ファイルエントリは異なります。ディスクレスのブート時の情報については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

ブートの相違点の要約

表 18-1 には、SunOS 4.x と Solaris 7 とのブートシーケンスの相違点を要約します。

表 18-1 ブートの相違点の要約

SunOS 4.x	Solaris 7	説明
ブートブロック	bootblk	ディスクから ufsboot をロードする
ブートプログラム	ufsboot	ディスクから unix をロードする
vmunix	unix	ブート可能なカーネルイメージ
boot.sun4c.sunos.4.1.1	inetboot	ネットワークから unix をマウントしてコピーする
rc.boot, rc.single	/etc/rcS	/usr をマウントし、ファイルシステムをチェックする
rc.local	/etc/rc2, /etc/rc3, /etc/rc2.d, /etc/rc3.d	システムの構成スクリプト

表 18-1 ブートの相違点の要約 続く

SunOS 4.x	Solaris 7	説明
config	modload, /etc/system, add_drv, rem_drv	システムカーネルをカスタマイズし、必要なモジュールをロードする
PROM モニタ、シングルユーザ、マルチユーザ	実行レベル 0~6、および S	システム実行レベル

再構成ブート

再構成ブートは、接続されたすべてのデバイスをチェックし、/devices と /dev にそれらの名前を構築するようシステムに指示します。新しいハードウェアをシステムに追加したときは、再構成ブートを行います。次のように `-r` オプションを使ってブートを開始します。

```
ok> boot -r
```

既存のタイプ (ドライバはすでにインストールされている) のデバイスを別に追加して、再構成ブートを忘れた場合、次のコマンドを使用して新しいデバイスを認識するようにシステムに指示することができます。

```
# touch /reconfigure
# _INIT_RECONFIG=YES /etc/init.d/drvconfig
# _INIT_RECONFIG=YES /etc/init.d/devlinks
```

開発者に関するデバイスの命名規則

この節では、63ページの「デバイス命名規則」の説明を拡張して、システムとカーネル開発者に関するデバイス命名規則を中心に説明します。

/devices

/devices ツリーは、カーネルで認識されたデバイスのツリーを表します。このツリーは `drvconfig(1M)` プログラムによって構成されます。通常 `drvconfig(1M)` は、システムが `-r` フラグでブートされた場合のみ実行されます。191ページの「再構成ブート」を参照してください。`drvconfig` は、ブート時に接続されて準備しているデバイス (ドライバのある) を格納するように /devices を構成します。

デバイスドライバがデバイスの存在を確認すると、デバイスドライバは `ddi_create_minor_node(9F)` を呼び出してエントリを作成します。

デバイスをシステムに追加するには `add_drv(1M)` コマンドを使用します。ドライバが正常に追加された場合、`add_drv(1M)` は `drvconfig` も実行します。

/dev

Solaris 7 では、/dev は /devices の中の実際のエントリへシンボリックリンクを作成するユーティリティプログラムによって管理されます。

- `disks(1M)`
- `tapes(1M)`
- `ports(1M)`
- `devlinks(1M)`

スクリプトを実行して、/dev から /devices へ適切なリンクを作成することができます。/devices 名がハードウェアの一意の名前であるのに対し、/dev 名はより簡単に親しみやすいという利点があります。

デバイスドライバの命名規則

システムにおける各デバイスは、デバイスドライバによって駆動されます。デバイスドライバは、デバイスの多くのインスタンスを管理します。デバイスは以下のような名前を与えられます。

- 物理名
- 論理名
- インスタンス名

物理名

物理名は `/devices` に格納されています。物理名はハードウェアについて記述し、プラットフォームおよび構成に依存します。以下に例を示します。

```
/devices/vme/xdc@6d,ee80/xd@0,0:g
```

物理名を使用すると、どのハードウェアが使用されているかを識別することができます。たとえば、`xdc@6d,ee80` は、VME A16, D32 空間のアドレス `0xee80` にあるディスクコントローラを指します。`vme(4)`、`driver.conf(4)` のマニュアルページを参照してください。

論理名

論理名は `/dev` に格納されています。論理名はデバイスの物理名のプラットフォーム固有の内容をできるだけ抽象化しています。たとえば `xd` というデバイスの論理名は次のようになります。

```
/dev/dsk/c2d0s6 (コントローラ 2、スレーブ 0、スライス 6 (4.x パーティション 'g'))
```

また、`sd` というデバイスの論理名は次のようになります。

```
/dev/dsk/c0t3d0s0 (コントローラ 3、ターゲット 0、lun 0、スライス 0 (4.x パーティション 'a'))
```

論理名は、コントローラのタイプについてはなにも表していません。つまり、SCSI でも IPI でも差はなく、両方とも単にディスクであるということです。

ディスク名

ディスク名は、SunOS 4.x リリースで使用されていた英字 `a~h` ではなくスライス番号 `0~7` の SVR4 規約に従っています。

ディスク名は、ブロックディスクデバイスについては `/dev/dsk/*`、raw ディスクについては `/dev/rdisk/*` という SVR4 規約に従っています。詳細については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

インスタンス名

インスタンス名とは、システムの n 番目のデバイスを意味します。たとえば、`sd20` のようになります。

インスタンス名は、ドライバエラーメッセージでレポートされることがあります。次のように `dmesg(1M)` の出力を見ると、物理名へのインスタンス名のバインディングを知ることができます。

```
sd9 at esp2: target 1 lun 1
sd9 is /sbus@1,f8000000/esp@0,800000/sd@1,0
<SUN0424 cyl 1151 alt 2 hd 9 sec 80>
```

インスタンス名がデバイスに割り当てられると、その名前がそのデバイスにバインドされたままになります。

インスタンス番号はデバイスのマイナー番号でコード化されます。リブートしてもインスタンス番号を一貫したものにするために、システムはそれらを `/etc/path_to_inst` ファイルに記録します。このファイルは起動時にだけ読み込まれ、現在は `add_drv(1M)` および `drvconfig(1M)` コマンドによって更新されます。`/etc/path_to_inst` ファイルについては、`path_to_inst(4)` のマニュアルページを参照してください。

コマンドリファレンス

この付録はユーザコマンドとシステム管理コマンドの情報を表形式で収録しています。すべての SunOS 4.x コマンドインタフェースを示し、Solaris 7 環境および SunOS/BSD ソース互換パッケージにおける使用方法の違いを示します。

リファレンスの使い方

- インタフェースに C (変更) とマークされている場合、SunOS 4.x コマンドと Solaris 7 コマンドとの違いについて簡単な説明があります。
- インタフェースに S (同じ) とマークされている場合、Solaris 7 インタフェースは SunOS 4.x インタフェースの機能をすべてサポートします。場合によってはインタフェースは拡張されていますが、これは SunOS 4.x インタフェースの完全なスーパーセットと考えることができます。
- インタフェースに A (置換) とマークされている場合、その置換については注を読んでください。
- インタフェースに N (利用不可) とマークされている場合、その置換については注を読んでください。利用できる別のコマンドがある場合、SunOS 5.6 の欄に示されます。

注 - SunOS 5.6 のディレクトリ構造は SunOS 4.x の構造とは異なります。したがって、同じように動作するコマンドでも、パス名が異なることがあります。たとえば、SunOS 4.x /usr/etc/newfs コマンドは現在 /usr/sbin/newfs に置かれていますが、インタフェースは変更されていません。このようなコマンド、またはこれに類するその他のコマンドは、この表のガイドラインに従い同じ「S」であると解釈されます。

/usr/bin と /usr/5bin の両方に存在するコマンドの場合、2つのエントリがリファレンス表にあり、第1のエントリは /usr/bin コマンドを説明し、第2のエントリは /usr/5bin コマンドを説明します。

すべての Solaris 7 インタフェースに関する詳細な情報については、『SunOS Reference Manual』のセクション1「User Commands」を参照してください。

例

表 A-1 から表 A-4 にリファレンスのエントリのサンプルを、その説明とともに示します。

表 A-1 例 1

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
fasthalt (8)	A	init 0 コマンドが同様の機能を提供。	S

fasthalt コマンドは、Solaris 7 ベースの製品では利用できません。SunOS/BSD 互換パッケージをインストールした場合に利用できます。init 0 コマンドが fasthalt コマンドに置き換わります。スクリプトまたはアプリケーションで互換パッケージの fasthalt コマンドを使用する場合には、これらのスクリプトまたはアプリケーションは他の SVR4 システムでは動作しません。互換パッケージコマンドはこのパッケージがインストールされたシステムの /usr/ucb にあります。コマンドの説明は fasthalt (1B) のように『SunOS Reference Manual』のセクション 1B 「User Commands」にあります。

表 A-2 例 2

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
cc (1V)	N	C コンパイラは C 言語の別パッケージ製品のツールでのみ利用可能。	C

C コンパイラは SunOS リリース 5.6 では利用できません。C コンパイラは SunOS/BSD 互換パッケージで利用できますが、別パッケージの C コンパイラを必要とし、SunOS 4.x コンパイラと同じインタフェースおよび出力は提供しません。

表 A-3 例 3

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
date (1V) -SysV	S		N
date (1V)	C	日付を設定する際に使用されるフォーマットは、SunOS リリース 5.6 ではわずかに異なっている。詳細については、date (1) のマニュアルページを参照のこと。	N

SunOS 4.x には、(SysV エントリで示される) /usr/5bin/date および (第 2 のエントリで示される) /usr/bin/date の 2 つの date コマンドがありました。/usr/5bin/date コマンドは SunOS 5.6 コマンドと同じです。使用しているパスに /usr/bin より前に /usr/5bin があつた場合、SunOS 5.6 のこのコマンドとの違いにまったく気がつかないでしょう。SunOS 4.x /usr/bin/date コマンドの使用に慣れたら、システムの日付を設定する前に SunOS 5.6 date (1) のマニュアルページを見るようにしてください。

表 A-4 例 4

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rev (1)	N		N

SunOS 4.x rev コマンドは SunOS 5.6 または BSD リリースにおいても利用できません。利用できる代替コマンドはありません。

コマンドリファレンス

以下の表は、SunOS 4.x の全コマンドインタフェースを一覧表示し、Solaris 2.6 環境と SunOS/BSD ソース互換パッケージにおける使用方法について説明します。

コマンドリファレンス (A, B)

表 A-5

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
ac (8)	A	システムアカウンティングリソースパッケージ (SAR) には ac で利用可能なアカウンティング機能のほとんどが提供される。	N
acctcms (8)	S		N
acctcom (8)	S		N
acctcon1 (8)	S		N
acctcon2 (8)	S		N
acctdisk (8)	S		N
acctdusg (8)	S		N
acctmerg (8)	S		N
accton (8)	S		N
acctprc1 (8)	S		N
acctprc2 (8)	S		N
acctwtmp (8)	S		N
adb (1)	S		N

表 A-5 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
adbgen(8)	S		N
add_client(8)	N	admintool(1M)	N
add_services(8)	A	swmtool(1M) コマンドが同様の機能を提供。	N
addbib(1)	S		N
adjacentscreens(1)	A	OpenWindows 環境には、複数のディスプレイを提供する方法が2つある。1つは、ある1つのマシンで2つのサーバを起動して、それぞれのサーバが特定のディスプレイを制御する方法。もう1つは、openwin -dev オプションを使って2つのディスプレイがある1つのサーバを起動する方法。	N
admin(1)	C	次の SunOS 4.x オプションは SunOS 5.6 システムソフトウェアでは利用できない。 -l release[,release ...]: 指定するリリースを delta に対してロックする。	N
adv(8)	N	RFS は存在しない。この機能は -f フラグによって引き続き利用可能。	N
aedplot(1G)	N		S
align_equals(1)	A	OpenWindows テキストメニュー indent コマンドが同様の機能を提供。	N
analyze(8)	A	コアファイルで adb(1) を使用してクラッシュを解析する。	N
apropos(1)	C	SunOS 4.x コマンドでは whatis データベースを使用したがる、SunOS 5.6 ではこのデータベースを windex といい、そのフォーマットはわずかに異なる。	N
ar(1V)	S		N
ar(1V) -SysV	C		N

表 A-5 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
arch(1)	C	オプションなしで、このコマンドは現在は「sun 4」を返す。この使用は勧められない。代わりに、uname(1)を使用のこと。オペレーティングシステム名とリリースレベルを判別するには、uname -srを使用。	S
arp(8C)	S		N
as(1)	C	SunOS 4.x オプション、-d2、-h、-j、-J、-k、-L、-O[n]は、SunOS 5.6 コマンドでは利用できない。	N
at(1)	S	SunOS 5.6 システムの at、atq、および atrm コマンドは、SunOS 4.x システムの場合とは動作が少し異なる。非特権ユーザに対するセキュリティが SunOS 5.6 システムではより強化されている。非特権ユーザは他のユーザのジョブを表示できない。	N
atoplot(1G)	N		S
atq(1)	C	SunOS 5.6 システムの at、atq、および atrm コマンドは、SunOS 4.x システムの場合とは動作が少し異なる。SunOS 4.x コマンドでは、ユーザ名が指定されなければ、待ち行列全体が表示される。SunOS 5.6 システムソフトウェアでは、呼び出し側が特権ユーザである場合に限り待ち行列全体が表示され、そうでない場合には呼び出し側に属するジョブだけが表示される。非特権ユーザは他のユーザのジョブを表示できない。非特権ユーザに対するセキュリティは SunOS 5.6 システムではより強化されている。	N
atrm(1)	C	SunOS 5.6 システムの at、atq、および atrm コマンドは、SunOS 4.x システムの場合とは動作が少し異なる。SunOS 4.x の「-」フラグは SunOS 5.6 コマンドでは -a とリネームされている。非特権ユーザに対するセキュリティは SunOS 5.6 システムではより強化されている。	N

表 A-5 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
audit(8)	C	-d または -u オプションは使用できない。 このコマンドは、Basic Security Module (BSM) が有効になっている場合だけ使用できる。	N
audit_warn	S		N
auditd(8)	S		N
automount(8)	C	次の SunOS 4.x オプションは、SunOS 5.6 コマンドでは利用できない。 -m: ディレクトリマップペアの初期化を抑制する。 auto.master と auto.home ファイルは、auto_master と auto_home にリネームされている。デフォルトのホームディレクトリパスは /export/home/ <i>username</i> となる。	N
awk(1)	S		N
banner(1V) -SysV	S		N
bar(1)	tar, cpio	tar(1) コマンドは、ほとんどの場合 bar の代わりに使用できる。cpio -iH bar を使用して既存の SunOS 4.x bar バックアップを復元できる。今後、bar 形式ファイルは作成できない。	N
basename(1)	S	SunOS 5.6 および SunOS/BSD 互換バージョンは、ともに SunOS 4.x と互換性があるが、引数を構文解析する方法が異なる。つまり、SunOS 5.6 は、3 つ以上の引数を受け付けない。また、SunOS/BSD 互換バージョンは 2 番目よりあとの引数をすべて無視する。	S
batch(1)	S	デフォルトでは、SunOS 5.6 の batch ジョブ <i>queuename</i> は指定されないで、SunOS 4.x コマンドではジョブは常に待ち行列 <i>b</i> に入れられていた。	N

表 A-5 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
bc (1)	S		N
bgplot (1G)	N		S
biff (1)	chmod	biff n: % chmod u+x 'tty' biff y: % chmod u-x 'tty'	S
bin-mail (1)	S	SunOS 5.6 の mail (1) コマンドと同じ。	N
biod (8)	N		N
boot (8S)	C	詳細については、boot (1M) のマニュアル ページを参照のこと。	N
bootparamd (8)	S		N

コマンドリファレンス (C, D)

表 A-6

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
C2conv (8)	N	本製品に関する情報については、購入先に 問い合わせること。	N
C2unconv (8)	N	本製品に関する情報については、購入先に 問い合わせること。	N
cal (1)	S		N
calendar (1)	S		N
cancel (1)	S		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
capitalize(1)	C	このコマンドの OpenWindows 版は OpenWindows テキストエディタで利用できる。	N
captain(8V) -SysV	S		N
cat(1V) -SysV	S		N
cat(1V)	S	SunOS 5.6 の cat コマンドでは、-t および -e オプションには -v オプションが必要。SunOS 5.6 の -t オプションは、SunOS 4.x の -v オプションと同等で、FORMFEED 文字を表示する。	N
catman(8)	S		N
cb(1)	S		N
cc(1V) -SysV	N		N
cc(1V)	N	C コンパイラは、C 言語別パッケージのツールでだけ利用できる。	C
cd(1)	S		N
cdc(1)	C	2つのバージョンは、読み取り不能である s.file を処理する方法が異なる。SunOS 4.x コマンドはエラーを出力する。SunOS 5.6 コマンドはエラーを表示しないで無視する。	N
cflow(1V) -SysV	N	cflow コマンドは、現在は別パッケージの製品で利用できる。	N
cflow(1V)	N	cflow コマンドは、現在は別パッケージの製品で利用できる。	N
chargefee(8)	S		
checkeq(1)	S		N
checknr(1)	S		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
chfn(1)	N		N
chgrp(1)	C	シンボリックリンクのデフォルトの動作は、SunOS 4.x から SunOS 5.6 との間で変更された。SunOS 4.x では、chgrp はシンボリック自体の所有権を変更していた。SunOS 5.6 では、chgrp はリンクをたどる。SunOS 5.6 でシンボリックリンクの所有権を変更するには、-h オプションを使用する。	N
chkey(1)	S		N
chmod(1V) -SysV	C	シンボリックリンクに出会うと、SunOS 5.6 での -R オプションはターゲットのモードを変更する。	N
chmod(1V)	S	シンボリックリンクに出会うと、SunOS 5.6 での -R オプションはターゲットのモードを変更する。 SunOS 5.6 コマンドは、「I」および「T」の 2 つのパーミッションもサポートする。	N
chown(8)	C	シンボリックリンクのデフォルトの動作は変更された。SunOS 4.x の chown はシンボリックリンクの所有権を変更していた。SunOS 5.6 の chown はリンクをたどる。リンクの所有権を変更するには、chown -h を使用する。SunOS 5.6 の chown コマンドでは、ファイルのグループ ID を変更できない。	S
chroot(8)	S		N
chrtbl(8)	A	SunOS 5.6 の localedef(1) でローカルデータベースを作成する。	N
chsh(1)	N		N
ckpacct(8)	S		N
clear(1)	S		N
clear_colormap(1)	N		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
clear_functions(1)	S		N
click(1)	N		N
clock(1)	A	Open Windows コマンドは /usr/demo/clock で利用できる。詳細については、clock(1) のマニュアルページを参照のこと。	N
clri(8)	S		N
cmdtool(1)	A	このコマンドは OpenWindows のコマンドツールに置き換えられる。	N
cmp(1)	S		N
coll(V) -SysV	S		N
col(1V)	C		N
colcrt(1)	N		N
colldef(8)	A	SunOS 5.6 では、local edef(1) でロケールデータベースを作成する。	N
coloredit(1)	A	このコマンドの機能は現在、OpenWindows プロパティウインドウによって処理される。	N
colrm(1)	N		N
comb(1)	C	2つのバージョンは、読み取り不能の s.file を処理する方法が異なる。SunOS 4.x コマンドはエラーを出力するが、SunOS 5.6 コマンドはエラーを表示しないで無視する。	N
comm(1)	S		N
compress(1)	S		N
config(8)	N		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
cp(1)	C	-R オプションは、SunOS 5.6 コマンドでは -r オプションに置き換えられる。	N
cpio(1)	S		N
cpp(1)	S		N
crash(8)	C	SunOS 4.x リリースで使用するデフォルトのネームリストは /vmunix となっているが、SunOS 5.6 では /kernel/unix となっている。	N
cron(8)	S		N
crontab(1)	S		N
crtplot(1G)	N		S
crypt(1)	S		N
cs(1)	S		N
csplit(1V) -SysV	S		N
ctags(1)	S		N
ctrace(1V) -SysV	N	次の SunOS 4.x オプションは、SunOS 5.6 コマンドでは利用できない。 -b: コードを追跡するのに基本的な関数だけを使用する。このオプションは、signal()、fflush()、longjmp()、または setjmp()x 関数が利用できないオペレーティングシステムで動作するのに必要。 -r オプションの構文は、SunOS 4.x および SunOS 5.6 の間で異なる。4.1 フォーマットは -rf。現在は、-r f。ctrace は別パッケージの製品として利用できる。	N
cu(1C)	S		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
cut (1V) -SysV	S		N
cxref (1V) -SysV	S		N
cxref (1V)	N	cxref は別パッケージの製品として利用できる。	N
date (1V) -SysV	S		N
date (1V)	C	日付を設定するとき使用するフォーマットは、SunOS 5.6 ではわずかに異なる。詳細については、date (1) のマニュアルページを参照のこと。	N
dbconfig (8)	S		N
dbx (1)	N	別パッケージの SPARCworks™ で利用できる。	N
dbxtool (1)	N	別パッケージの SPARCworks で、コマンドデバッグとして利用できる。	N
dc (1)	S		N
dcheck (8)	A	通常の整合性検査に対して fsck (1M) コマンドを使用する。ncheck (1M) コマンドは dcheck -i 番号の機能を置き換える。	N
dd (1)	C	SunOS 4.x コマンドでは、大きさの接尾辞 w (ワード) で使用される大きさは 4 バイト単位となっているが、SunOS 5.6 システムソフトウェアでは、w は 2 バイト単位。k、b、または w は、1024、512、または 2 の倍数をそれぞれ指定するための接尾辞として使用する。unblock と block 変換オプションは新たに採用された。	N
defaults_from_input (1)		このコマンドの機能は現在、OpenWindows プロパティウィンドウによって処理される。	N
defaults_merge (1)	S		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
defaults_to_indentpro(1)		このコマンドの機能は現在、OpenWindows プロパティウィンドウによって処理される。	N
defaults_to_mailrc(1)		このコマンドの機能は現在、OpenWindows プロパティウィンドウによって処理される。	N
defaultsedit(1)		このコマンドの機能は現在、OpenWindows プロパティウィンドウによって処理される。	N
delta(1)	C	引数としてディレクトリが指定されると、そのディレクトリのすべてのファイルが処理される。SunOS 4.x では、ディレクトリのファイルがエラーを起こすと、エラーが生成される。SunOS 5.6 コマンドでは、そのようなファイルは表示を行うことなく無視される。	N
deroff(1)	S		N
des(1)	S		N
devinfo(8S)	C	prtconf(1M) コマンドに同様の機能がある。	N
devnm(8)	C	SunOS 4.x と SunOS 5.6 システムソフトウェアの間の出力フォーマットはまったく異なる。 SunOS 4.x では、名前引数はオプション。SunOS 5.6 システムソフトウェアでは、必須。	N
df(1V) -SysV	C		N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
df (1V)	C	SunOS 4.x の df には、SunOS 5.6 の df コマンドとある程度異なる出力を内容とする異なる出力フォーマットがある。SunOS 5.6 で <code>-k</code> オプションを指定すれば、SunOS 4.x コマンドに似た出力フォーマットを提供する。SunOS 4.x では <code>df -t</code> オプションでファイルシステムの形式を指定すると、該当する形式ファイルシステムの形式でレポートを行うが、SunOS 5.6 で <code>df -t</code> オプションを指定すると合計の付いた完全なリストを出力する。SunOS 5.6 では <code>df -l</code> オプションを使用すると、ローカルファイルシステムを表示できる。	S
diff (1)	C	いくつかのフラグの動作は 2 つのバージョンの間で異なる。SunOS 4.x では、 <code>-c</code> オプションは、オプションの引数を相違ごとに表示する行数とみなす。引数が与えられなければ、デフォルトは 3 行。SunOS 5.6 では、 <code>-s</code> オプションとその引数の間にスペースが必要。	N
diff3 (1V) -SysV	S		N
diff3 (1V)	S		N
diffmk (1)	S		N
dircmp (1V) -SysV	S		N
dirname (1V) -SysV	S		N
dis (1)	C	次の SunOS 4.x でのオプションは SunOS 5.6 では利用できない。 <code>-da sec: sec</code> をデータとして逆アセンブルし、データの実際のアドレスを出力する。SunOS 5.6 では <code>-D sec</code> オプションを使用すると、同じ処理が行われる。	N
diskusg (8)	A	acctdusg (1M) コマンドが同様の機能を提供。	N

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
dkctl(8)	N		N
dkinfo(8)	A	prtvtoc(1M) コマンドが同様の機能を提供。	N
dmesg(8)	S		N
dname(8)	N	RFS は利用できない。	N
dodisk(8)	S		N
domainname(1)	S		N
dorfs(8)	N	RFS は利用できない。	N
dos2unix(1)	S		N
du(1V) -SysV	S		N
du(1V)	C	SunOS 4.x ではディスク使用量を 1 キロバイト単位で表示するが、SunOS 5.6 ではディスク使用量を 512 バイトブロック単位で表示する。-k オプションを使用すると、使用量を 1 キロバイト単位で表示することができる。	S
dumbplot(1G)	N		S

表 A-6 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
dump (8)	A	<p>ufsdump コマンドが同様の機能を提供する。次の SunOS 4.x でのオプションは SunOS 5.6 にはない。</p> <p>-a <i>archive-file</i>: SunOS 5.6 では -a オプションはアーカイブの各メンバのアーカイブヘッダをダンプする。</p> <p>-D: フロッピーディスクをダンプ媒体として指定する。SunOS 5.6 では -D オプションはデバッグ情報をダンプする。</p> <p>-v: ダンプしようとするファイルシステムを照合する。SunOS 5.6 では -v オプションは、数値表現ではなく文字で情報をダンプする。</p>	N
dumpadm(8)	New	<p>システム管理者がオペレーティングシステムのクラッシュダンプを設定できる。このように設定すると、ダンプデータはダンプデバイスに圧縮形式で格納される。専用のダンプデバイス (一次スワップ領域ではない) がダンプ構成に含まれている場合、コアファイルの保存がバックグラウンド処理で実行される。</p>	N
dumpfs (8)	A	<p>fstyp -F -ufs -v コマンドが同様の機能を提供。</p>	N
dumpkeys (1)	S		N

コマンドリファレンス (E, F, G, H, I, J)

表 A-7

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
e(1)	A	ex(1) コマンドが同様の機能を提供。	S
echo(1V) -SysV	S		N
echo(1V)	C	-n オプションは SunOS 4.x では改行を抑制した。SunOS 5.6 では \c を使用する。	S
ed(1)	S		N
edit(1)	S		N
edquota(8)	S		N
eprom(8S)	S		N
egrep(1V)	S		N
eject(1)	S		N
enroll(1)	N		N
env(1)	S		N
eqn(1)	S		N
error(1)	S		N
etherd(8C)	A	snoop(1M) コマンドが同様の機能を提供。	N
etherfind(8C)	A	snoop(1M) コマンドが同様の機能を提供。	N
ex(1)	S		N
expand(1)	S		N
exportfs(8)	A	share(1M) コマンドが同様の機能を提供。	N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
expr (1V) -SysV	S		N
expr (1V)	C		S
extract_files (8)	A	pkgadd (1M) コマンドが同様の機能を提供。	N
extract_patch (8)	A	pkgadd (1M) コマンドが同様の機能を提供。	N
extract_unbundled (8)	A	swmtool (1M) コマンドが同様の機能を提供。	N
false (1)	S		N
fastboot (8)	A	init 6 コマンドが同様の機能を提供。	S
fasthalt (8)	A	init 0 コマンドが同様の機能を提供。	S
fdformat (1)	S		N
fgrep (1V)	S		N
file (1)	C	次の SunOS 4.x でのオプションは SunOS 5.6 にはない。 -L: ファイルがシンボリックリンクならば、リンク自体ではなくリンクによって参照されるファイルをテストする。	S
find (1)	C	次の SunOS 4.x のオプションは SunOS 5.6 では利用できない。 -n cpio-device: cpio -c フォーマットで現在のファイルをデバイスに書き込む。	N
finger (1)	S		N
fingerd (8)	S		N
fmt (1)	C		N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
fmt_mail(1)	N		N
fold(1)	S		N
fontedit(1)	N		N
foption(1)	N		N
format(8S)	S		N
fpa_download(8)	N		N
fparel(8)	N		N
fpaversion(8)	N		N
fpurel(8)	N		N
fpuverson4(8)	A	psrinfo -v で情報の入手可能。	N
from(1)	N		S
fsck(8)	C	SunOS 4.x の fsck コマンドは SunOS 5.6 とはかなり異なる。SunOS 5.6 では、ファイルシステム形式を指定した後で大部分のオプションを指定する。fsck -m はファイルシステムの高速度チェックを行う。-w オプションは利用できない。新しいオプションは、-f、-v、および -o。	N
fsck-cdrom(8)	N		N
fsirand(8)	S		S
ftp(1C)	S		N
ftpd(8C)	S		N
fumount(8)	S	RFS は利用できない。	N
fusage(8)	S	RFS は利用できない。	N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
fuser(8)	S		N
fwtmp(8)	S		N
gcore(1)	S		N
generic_args(1)	N		N
get(1)	C	SunOS 5.6 では ASCII ファイルだけを生成する。SunOS 4.x にはそのような制約はない。ディレクトリが指定されていて、ディレクトリ内のファイルが正常に得られない場合は、SunOS 4.x ではエラーを表示する。SunOS 5.6 では無視し、なにも表示しない。	N
get_alarm(1)	N		N
get_selection(1)	A	xv_get_sel(1) コマンドが同様の機能を提供。	N
getopt(1V) -SysV	S		N
getoptcvt(1)	S		N
getopts(1)	S		N
gettable(8C)	S		N
getty(8)	S		N
gfxtool(1)	N		N
gigiplot(1G)	N		S
glob(1)	S		N
goto(1)	S		N
gpconfig(8)	N		N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
gprof(1G)	S		N
graph(1G)	S		N
grep(1V)	S		N
grep(1V) -SysV	C	次のオプションが変更されている。 -w: 正規表現を\< および \> で囲まれているかのようにワードとして検索する。	N
groups(1)	S		S
grpck(8V)	S		N
gxtest(8S)	N		N
halt(8)	S		N
hashcheck(1)	S		N
hashmake(1)	S		N
hashstat(1)	S		N
head(1)	S		N
help(1)	S		N
help_open(1)	S		N
hostid(1)	S		S
hostname(1)	S		S
hostrfs(8)	N	RFS は利用できない。	N
hp7221plot(1G)	N		S
hpplot(1G)	N		S

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
htable(8)	S		N
i386(1)	S		N
iAPX286(1)	S		N
icheck(8)	A	fsdb() が代替コマンド。	N
iconedit(1)	A	このコマンドは OpenWindows のアイコンエディットツールに置き換えられる。	N
id(1)			
id(1V) -SysV	S		N
idload(8)	N	RFS は利用できない。	N
ifconfig(8C)	S		N
imemtest(8C)	N		N
implot(1G)	N		Y
in.comsat(8C)	S		N
in.fingerd(8C)	S		N
in.ftpd(8C)	S		N
in.named(8C)	S		N
in.rexecd(8C)	S		N
in.rlogind(8C)	S		N
in.routed(8C)	S		N
in.rshd(8C)	C	ポートの範囲は、SunOS 4.x と SunOS 5.6 の間で異なる。SunOS 4.x では範囲は 512~1023 であり、SunOS 5.6 では 0~1023 。	N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
in.rwhod(8C)	S		N
in.talkd(8C)	S		N
in.telnetd(8C)	S		N
in.tftpd(8C)	S		N
in.tnamed(8C)	S		N
in.uucpd(8C)	S		N
indent(1)	N	このコマンドは別パッケージの製品として利用できる。	N
indentpro_to_defaults(1)	A	このコマンドの機能は現在 OpenWindows プロパティシートによって処理されている。	N
indxbib(1)	S		N
inetd(8C)	S		N
infocmp(8V) -SysV	C		N
infocmp(8V)	C	-s オプションの構文は、SunOS 4.x と SunOS 5.6 の間で異なる。SunOS 5.6 では、-s とその引数の間にスペースを入れるなければならない。SunOS 4.x では、スペースはオプション。	N
init(8)	C	SunOS 5.6 では SunOS 4.x とは非常に異なる。詳細については、init(1M) のマニュアルページを参照のこと。	N
inline(1)	N	このコマンドは別パッケージの製品として利用できる。	N
input_from_defaults(1)	N		N

表 A-7 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
insert_brackets(1)	A	同じ名前の OpenWindows のコマンドが OpenWindows のテキストエディタで利用できる。	N
install(1)	C	-c、-o、-s オプションの機能は、SunOS 4.x と SunOS リリース 5.6 の間で異なる。	S
installboot(8S)	C	パス名と構文が変更された。	N
installtxt(8)	A	msgfmt(1) コマンドが同様の機能を提供。	N
intr(8)	N		N
iostat(8)	S	新しいオプションは次のとおり。 -x: ディスク統計情報を与える。 -c システムのユーザモード、システムモード、およびアイドル状態の時間の割合を表示する。	N
ipallocald(8C)	N		N
ipcrm(1)	S		N
ipcs(1)	S		N
isainfo(1)	New	実行中のシステムでサポートされている Instruction Set Architecture(ISA) の情報を印刷できる新しいコマンド。	N
join(1)	C	SunOS 4.x では、-a オプションは引数として 1、2、または 3 をとる。SunOS 5.6 では、この値は 1 または 2 のみが可能。SunOS 4.x では -j への引数は 1 または 2 のみが可能。SunOS 5.6 ではそのような制約はない。	N

コマンドリファレンス (K, L, M, N, O)

表 A-8

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
kadb(8S)	S		N
keyenvoy(8C)	N		N
keylogin(1)	S		N
keylogout(1)	S		N
keyserv(8C)	S		N
kgmon(8)	S		N
kill(1)	S		N
labelit(8)	S		N
last(1)	S		N
lastcomm(1)	S		N
lastlogin(8)	S		N
ld(1)	C	SunOS 4.x ld コマンドと SunOS 5.6 コマンドの間には多くの相違がある。-align、-A、-B、-D、-M、-n、-t、-T、-Tdata、-x、-X、-y、および -z は、SunOS 4.x のオプションでは利用できない。-assert オプションは -z オプションに置き換えられた。-d、-dc、-dp オプションは SunOS 5.6 でのデフォルト。これらのオプションを無効にするには -b を使う。	S
ldconfig(8)	N		N
ldd(1)	S		N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
leave(1)	N	cron(1M) と at(1) コマンドが同様の機能を提供。	N
lex(1)	C	次の SunOS 4.x でのオプションは SunOS 5.6 では利用できない。-f: 生成するテーブルをパックしないことにより高速でコンパイルする。このオプションは小さなプログラムに限られる。	N
line(1)	S		N
link(8V)	S		N
lint(1V) -SysV	N		N
lint(1V)	N	別パッケージの SPARCworks で利用可能。	S
listen(8)	S		N
ln(1V)	C	すでにターゲットが存在する場合、SunOS 4.x ではそれを削除しない。SunOS 5.6 では、ユーザに適切なパーミッションが与えられていれば、ターゲットを削除する。SunOS 4.x では、-f オプションはディレクトリへのハードリンクを強制的に行う。	S
ln(1V) -SysV	C	SunOS 4.x の /usr/5bin/ln コマンドでは、-f オプションによりパーミッションを表示したり、ユーザへ質問を行ったり、またはエラーの報告をせずに、ファイルをリンクさせることができる。ディレクトリに強制的にハードリンクさせる -F オプションは、SunOS 5.6 では利用できない。	N
loadkeys(1)	S		N
lockd(8C)	S		N
lockscreen(1)	A	このコマンドは OpenWindows の xlock(1) で利用できる。フォアグラウンドパターンは異なるが、lockscreen コマンドの機能は xlock においても同じ。	N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
logger(1)	N		S
login(1)	S		N
logname(1)	S		N
lp(1)	S		N
lpc(8)	A	lpadmin(1M) コマンドが同様の機能を提供。	S
lpd(8)	A	lpadmin(1M) コマンドが同様の機能を提供。	S
lpq(1)	A	lpstat(1) コマンドが同様の機能を提供。	S
lpr(1)	A	lp(1) コマンドが同様の機能を提供。	S
lprm(1)	A	cancel コマンドが同様の機能を提供。	S
lpstat(1)	S		N
lptest(1)	N		S
ls(1V) -SysV	C		N
ls(1V)	S		S
lsw(1)	N		N
Mail(1)	A	mailx(1) コマンドが同様の機能を提供。	N
m4(1V)	C	式評価に関して構文上の非互換性がある。	N
m4(1V) -SysV	S		N
m68k(1)	S		N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
mach(1)	S		S
mail(1) - UCB	mailx		S
mail(1)	C	現在は /usr/bin/mailにあるが、SunOS 4.x では /usr/ucb/mail にあった。このエントリは /usr/bin/mail にインストールされたメールコマンドを参照する。SunOS 4.x の mail は以下を除く SunOS 5.6 コマンドと互換性がある。-i: -i (割り込みを無視する) オプションは利用できない。SunOS 4.x コマンドでは postmark 行の前にメールをすすめる「>」が付く。これは SunOS 5.6 コマンドでは必要とされない。	N
mailrc_to_defaults(1)	C	現在このコマンドの機能は OpenWindows プロパティウインドウで処理される。	N
mailstats(8)	S		N
mailtool(1)	C	このコマンドは OpenWindows のメールツールで利用できる。	N
make(1)	S	SVR4 と SVID の make は /usr/ccs/lib/svr4.mke で利用できる。	N
makedbm(8)	C	このコマンドの SunOS 5.6 インタフェースは SunOS 4.x インタフェースと互換性がある。SunOS 5.6 では SunOS 4.x で使用していた /usr/lib/dbm ではなく、/usr/lib/ndbm を使用する。	N
makedev(8)	N		N
makekey(8)	S		N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
man(1)	C	<p>オンラインマニュアルページの構成が変更された。すべてのセクションの説明については <code>intro(1)</code> を参照のこと。現在は、<code>man</code> コマンドでは <code>man</code> が検索するディレクトリのデフォルトの順番を指定できる。次のように、2つの新しいオプションによりマニュアルページを探し出すのがより簡単になった。</p> <p><code>-a</code> は、<i>title</i> に一致するマニュアルページを見つけた順にすべて表示する。</p> <p><code>-l</code> は、<i>title</i> に一致するすべてのマニュアルページのリストを表示する。</p> <p><code>-s</code> オプションは <i>section number</i> 引数に置き換わる。</p>	N
mc68010(8)	S		N
mc68020(8)	S		N
mc68881version(8)	N		N
mconnect(8)	S		N
mesg(1)	S		N
mkdir(1)	S		N
mkfile(8)	S		N
mkfs(8)	C	<p>インタフェースは2つのバージョン間で非常に異なる。SunOS 5.6では異なるファイルシステム形式を提供する。</p>	N
mknod(8)	S		N
mkproto(8)	C		N
mkstr(1)	N		S
modload(8)	C	<p>モジュールは通常、<code>modload</code> を使用して自動的にロードされる。</p>	N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
modstat (8)	A	modinfo(1M) コマンドが同様の機能を提供。	N
modunload (8)	C	モジュールは通常、自動的にロードされない。	N
monacct (8)	S		N
more (1)	S		N
mount (8)	C	インタフェースは 2 つのバージョン間でかなり異なる。SunOS 5.6 では、(ファイルシステムが /etc/vfstab に入っていない場合) ファイルシステム形式を指定した後で、ほとんどのオプションを指定しなければならない。	N
mount_tfs (8)	N		N
mountd (8C)	S		N
mt (1)	S		N
mv (1)	S		N
named (8C)	C	ネームデーモンは in.named に変更された。	N
nawk (1)	S		N
ncheck (8)	C	異なるファイルシステム形式の指定を許可できるように修正された。	N
ndbootd (8C)	N		N
neqn (1)	S		N
netstat (8C)	S		N
newaliases (8)	S		N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
newfs(8)	S	/usr/etc/newfs から /usr/sbin/newfs に移動された。	N
newgrp(1)	S		N
newkey(8)	S		N
nfsd(8)	S		N
nfsstat(8C)	S		N
nice(1)	C	SunOS 4.x には、csh に組み込まれているものと、/usr/bin にインストールされている、2 つの nice がある。csh に組み込まれているコマンドのデフォルトのプロセス優先順位は 4 で、/usr/bin/nice のデフォルト値は 10。SunOS 5.6 コマンドのデフォルトは 10。csh に組み込まれている SunOS 4.x コマンドでは、追加の -+ オプション (nice -+n) は、nice 値を n だけ増分するのではなく、nice 値を n に設定するという点で、/usr/bin にある SunOS 4.x コマンドとわずかに異なる構文を使用する。	N
nl(1V) -SysV	S		N
nlsadmin(8)	C	-l オプションの機能は、バージョン間で異なる。SunOS 4.x ソフトウェアでは、そのネットワークの接続相手が次に起動するまで addr を変更しても有効にならないが、SunOS 5.6 ソフトウェアでは即座に有効になる。SunOS 4.x ソフトウェアでは addr は 16 進表記で指定できるが、SunOS 5.6 ソフトウェアではできない。SunOS 4.x の -m オプションは SunOS 5.6 では利用できない。このオプションは、示された接続相手を通じて利用可能なサービスのリストに新しいサービスを追加するのに使用される。	N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
nm(1)	C	次の SunOS 4.x オプションは SunOS 5.6 では利用できない。-g、-p、-s、および -a、-n、-o、-r オプションは SunOS 4.x および SunOS 5.6 では異なる。	N
nohup(1V)	C		N
nohup(1V) -SysV	S		N
nroff(1)	S		N
nslookup(8C)	S		N
nsquery(8)	S		N
nulladm(8)	S		N
od(1V)	S		N
od(1V) -SysV	S		N
old-analyze(8)	N		N
old-ccat(1)	N		N
old-clocktool(1)	N		N
old-compact(1)	N		N
old-eyacc(1)	N		N
old-filemerge(1)	N		N
old-make(1)	N		N
old-perfmon(1)	N		N
old-prmail(1)	N		N
old-pti(1)	N		N

表 A-8 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
old-setkeys(1)	N		N
old-sun3cvt(1)	N		N
old-syslog(1)	N		N
old-uncompact(1)	N		N
old-vc(1)	N		N
on(1C)	S		N
overview(1)	N		N

コマンドリファレンス (P, Q, R)

表 A-9

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
pac(8)	N		N
pack(1V)	S		N
pack(1V) -SysV	S	SunOS 4.x の /usr/5bin/pack コマンドでは、ファイル名は 12 文字に制限される。SunOS 5.6 では、{NAME_MAX}-2 に制限される。SunOS 5.6 の pack および unpack コマンドは SunOS 4.x コマンドと互換性がある。	N
page(1)	S		N
pagesize(1)	S		S

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
passwd(1)	C	-F <i>filename</i> オプションは利用できない。 -f と -s オプションには、異なる意味がある。 -f オプションにより、ユーザは次のログインでパスワードを強制的に変更させられる。 -s オプションはユーザのログイン名のパスワード属性を表示する。	N
paste(1V) -SysV	S		N
pax(1V)	C		N
paxcpio(1V)	A	cpio(1) コマンドと pax(1) コマンドが同様の機能を提供。	N
pccat(1V) -SysV	S		N
pdp11(1)	S		N
perfmeter(1)	A	このコマンドは OpenWindows のパフォーマンスメータツールとして SunOS 5.6 で利用できる。	N
pg(1V) -SysV	S		N
pgrep(1)	New	システム上のアクティブなプロセスを監視し、コマンド行で指定された条件に合致する属性を持つプロセスのプロセス ID を表示する。	N
ping(8C)	S		N
pkill(1)	New	pgrep と同様に動作。ただし、プロセス ID が表示される代わりに、プロセス ID が合致するたびに kill(1) でシグナルが送られる点異なる。	N
plot(1G)	N		S
plottoa(1G)	N		S
portmap(8C)	A	rpcbind(1M) コマンドが同様の機能を提供。	N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
pr (1V)	C		N
pr (1V) -SysV	S		N
praudit (8)	S		N
prctmp (8)	S		N
prdaily (8)	S		N
printenv (1)	A	env (1) コマンドが同様の機能を提供。	S
prof (1)	C	SunOS 4.x -v オプションは SunOS 5.6 では利用できない。このオプションは出力をすべて抑止し、plot (1) フィルタで表示できるようにする標準出力にプロファイルのグラフィックバージョンを生成する。SunOS 4.x の -a オプションは、シンボルをすべて表示することを要求するが、SunOS 5.6 では外部シンボルのみ表示される。	N
prs (1)	C	このバージョンでは、読み取り不能な s.file を処理する方法が異なる。SunOS 4.x では、読み取り不能な s.file が出現すると、エラーを出力して続行するが、SunOS 5.6 ではエラーを表示せず、無視する。	N
prt (1)	S		N
prtacct (8)	S		N
ps (1)	C	-C、-k、-n、-r、-S、-U、-v、-w、-x の SunOS 4.x オプションは、SunOS 5.6 では利用できない。次のオプションは、2つのバージョン間では異なる意味を持つ。 -c: SunOS 4.x では、このオプションはコマンド名を表示する。SunOS 5.6 では、これは、新しいプロセススケジューラ的设计を反映したフォーマットで情報を出力する。	S

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
psrinfo(1)		ブートするカーネルとは無関係に SPARC V9 CPU と以前の SPARC CPU を識別できる UltraSPARC プラットフォームで検出された SPARC V9 CPU のみが、64 ビットの OS および アプリケーションを実行可能。	
pstat(8)	A	sar(1M) が同様の機能を提供。swap -s はシステムで利用可能な全スワップ領域を示す。	N
ptx(1)	N		N
pwck(8V)	S		N
pwd(1)	S		N
pwdauthd(8C)	N	別パッケージの製品により将来のリリースでも同様の機能が利用可能となる。本製品に関する情報については、購入先に問い合わせること。	N
quot(8)	S		N
quota(1)	S		N
quotacheck(8)	S		N
quotaoff(8)	S		N
quotaon(8)	S		N
ranlib(1)	C	ar(1) コマンドが同様の機能を提供。ranlib は NULL スクリプトとして存続。	N
rarpd(8C)	S		N
rasfilter8to1(1)	N		N
rastrepl(1)	N		N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rc(8)	N	/etc/init.d の下にある構成スクリプトが同様の機能を提供。rc ファイルの構成は SunOS 5.6 システムでは変更された。現在では実行レベルで分割される。	N
rc.boot(8)	N	/etc/init.d の下にある構成スクリプトが同様の機能を提供。	N
rc.local(8)	N	/etc/init.d の下にある構成スクリプトが同様の機能を提供。	N
rcp(1C)	S		N
rdate(8C)	S		N
rdist(1)	S		N
rdump(8)	A	ufsdump(1M) コマンドが同様の機能を提供。	N
reboot(8)	S		N
red(1)	S		N
refer(1)	S		N
rehash(1)	S		N
remove_brackets(1)	A	このコマンドは OpenWindows テキストエディタで利用できる。	N
renice(8)	A	priocntl(1) コマンドが同様の機能を提供。	S
repquota(8)	S		N
reset(1)	A	stty が同様の機能を提供。	S
restore(8)	A	SunOS 5.6 コマンド、ufsrestore は、ufsdump によって行われる媒体終了の検出を利用できるように性能が強化されている。	N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rev(1)	N		N
rexd(8C)	A	in.rexd が同様の機能を提供。	N
rexcld(8C)	A	in.rexcld が同様の機能を提供。	N
rfadmin(8)	N	RFS は利用できない。	N
rfpasswd(8)	N	RFS は利用できない。	N
rfstart(8)	N	RFS は利用できない。	N
rfstop(8)	N	RFS は利用できない。	N
rfuadmin(8)	N	RFS は利用できない。	N
rfudaemon(8)	N	RFS は利用できない。	N
ring_alarm(1)	N		N
rlogin(1C)	C	SunOS 4.x におけるエスケープのための ~dsusp シーケンスは、SunOS 5.6 では利用できない。また、-e オプションの構文は、SunOS 4.x と SunOS 5.6 の間で異なる。SunOS 4.x では、構文は -ec。SunOS リリース 5.6 では、-e c。	N
rlogind(8C)	A	in.rlogind も同様の機能を提供。	N
rm(1)	S		N
rm_client(8)	A	SunOS 5.6 システムでは admintool(1M) ユーティリティがこのコマンドに置き換わる。	N
rm_services(8)	A	swmtool(1M) が同様の機能を提供。	N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rmail(8C)	C	SunOS 4.x では、受信したリモートからのメールを uucp(1C) を使って処理する。これは、明示的に uucp(1C) および sendmail(8) と併用するものとして設計されている。SunOS 5.6 rmail は mailail(1) へのリンクであり、メールを読み出すのに使用される。	N
rmdel(1)	C	バージョン間では、読み取り不能な s.file を処理する方法が異なる。SunOS 4.x は、読み取り不能な s.file が出現すると、エラーを出力して続行するが、SunOS 5.6 では表示せずエラーを無視する。	N
rmdir(1)	S		N
rmntstat(8)	N	RFS は利用できない。	N
rmt(8C)	S		N
roffbib(1)	S		N
route(8C)	C	SunOS 4.x の route コマンドは、gethostent(3) を使用してすべてのシンボリック名およびゲートウェイを調べるが、SunOS 5.6 では gethostbyname(3) を使用する。	N
routed(8)	A	in.routed も同様の機能を提供。	N
rpc.bootparamd(8)	S		N
rpc.etherd(8C)	N	snoop(1M) ではこのデーモンは用いない。	N
rpc.lockd(8C)	A	lockd も同様の機能を提供。	N
rpc.mountd	A	mountd も同様の機能を提供。	N
rpc.rexd(8C)	S		N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rpc.rquotad(8C)	S		N
rpc.rstatd(8C)	S	現在は /usr/lib/netsvc/rstat にある。	N
rpc.rusersd(8C)	S	現在は /usr/lib/netsvc/rusers にある。	N
rpc.rwallld(8C)	S	現在は /usr/lib/netsvc/rwall にある。	N
rpc.showfhd(8C)	A	showfhd(1M) コマンドが同様の機能を提供。	N
rpc.sprayd(8C)	S	現在は /usr/lib/netsvc/spray にある。	N
rpc.statd(8C)	S	現在は /usr/lib/netsvc/rstat にある。	N
rpc.user_agentd(8C)	N		N
rpc.yppasswdd(8C)	N		N
rpc.yputdated(8C)	N		N
rpcgen(1)	S		N
rpcinfo(8)	S		N
rrestore(8)	A	ufsrestore(1M) コマンドが同様の機能を提供。	N
rsh(1C)	S		N
runacct(8)	S		N
rup(1C)	S		N
ruptime(1C)	S		N

表 A-9 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
rusage(8)	N		S
rusers(1C)	S		N
rwall(1C)	S		N
rwho(1C)	S		N

コマンドリファレンス (S, T)

表 A-10

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
sa(8)	A	acct(1M) が同様の機能を提供。	N
sact(1)	C	バージョン間では、読み取り不能な s.file を処理する方法は異なる。SunOS 4.x コマンドは、読み取り不能な s.file が出現すると、エラーを出力して続行する。SunOS 5.6 コマンドは表示せずエラーを無視する。	N
savecore(8)	S		N
sccs(1)	S		N
sccs-admin(1)	S		N
sccs-cdc(1)	S		N
sccs-comb(1)	S		N
sccs-delta(1)	S		N
sccs-get(1)	S		N

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
sccs-help(1)	S		N
sccs-prs(1)	S		N
sccs-prt(1)	S		N
sccs-rmdel(1)	S		N
sccs-sact(1)	S		N
sccs-sccsdiff(1)	S		N
sccs-unget(1)	S		N
sccs-val(1)	S		N
sccsdiff(1)	C		N
screenblank(1)	C	OpenWindows <code>xset -s -600</code> コマンドが同じ機能を提供。	N
screendump(1)	N		N
screenload(1)	N		N
script(1)	S		N
scrolldefaults(1)	C	このコマンドの機能は、現在 OpenWindows のプロパティウィンドウによって処理される。	N
sdiff(1V) -SysV	S		N
sed(1V) -SysV	S		N
sed(1V)	C	SunOS 4.x の <code>/usr/5bin/sed</code> および SunOS 5.6 の <code>sed</code> は、テキスト行から最初にある SPACE および TAB 文字を削除しない。	S
selection_svc(1)	N		N

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
sendmail(8)	S		N
set4(8)	N		N
set_alarm(1)	N		N
setkeys(1)	N		N
setsid(8V)	N		N
setup_client(8)	N		N
setup_exec(8)	N		N
sh(1)	C	SunOS 4.x では、組み込みコマンド echo および test の動作は、環境変数 PATH における usr/bin と /usr/5bin との相対的な位置に依存する。現在では /usr/ueb と /usr/bin の相対的な並び換えによって動作が決まる。	N
shelltool(1)	C	このコマンドは OpenWindows のシェルツールで利用できる。	N
shift_lines(1)	C	OpenWindows コマンドは OpenWindows のテキストエディタで利用できる。	N
showfh(8C)	N		N
showmount(8)	S		N
shutacct(8)	S		N
shutdown(8)	C	SunOS 4.x の shutdown(8) コマンドは SunOS 5.6 の shutdown(1M) コマンドとはかなり異なる。デフォルトでは、SunOS 5.6 の shutdown(1M) は、シャットダウン処理を起動する前に確認を要求するが、SunOS 4.x の shutdown(8) は確認を求めない。さらに、SunOS 5.6 のコマンドには、-f、-h、-k、-n、-r の SunOS 4.x オプションが存在しない。	S

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
size(1)	C	SunOS 4.x コマンドはサイズを 16 進数および 10 進数で出力し、ファイル名はオプション (デフォルトは a.out)。SunOS 5.6 コマンドは、-o または -x オプションを指定しないと、10 進数だけでサイズを出力する。ファイル名は必要。	N
skyversion(8)	N		N
sleep(1)	S		N
soelim(1)	S		N
sort(1V) -SysV	S		N
sort(1V)	C		N
sortbib(1)	S		N
sparc(1)	S		N
spell(1)	C	SunOS 4.x での -h <i>spellhist</i> オプションは、SunOS 5.6 コマンドで利用できない。 このオプションは、スペルを間違ったワードにユーザ/日付スタンプをつけて <i>spellhist</i> に入れる。	N
spellin(1)	S		N
spline(1G)	S		N
split(1)	S		N
spray(8C)	C	SunOS 4.x の -i <i>delay</i> オプションは SunOS 5.6 コマンドでは利用できない。このオプションは、RPC ではなく ICMP エコーパケットを使用するように指定する。	N
startup(8)	S		N
strings(1)	S		N

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
strip(1)	S		N
stty(1V) -SysV	C		N
stty(1V)	C	SunOS 5.6 の stty コマンドによってサポートされない SunOS 4.x オプションは、decctlq、tandem、cbreak、ctlecho、prterase、crtkill、cols、tab3、crt、dec、term。	S
stty_from_defaults(1)	N		N
su(1V) -SysV	S		N
su(1V)	C	SunOS 4.x の -f オプションは SunOS 4.x の /usr/5bin/suu または SunOS 5.6 の su コマンドによってサポートされない。このオプションは C シェルの高速 su に使用される。	N
sum(1V) -SysV	S		N
sum(1V)	C		S
sun(1)	S		N
sundiag(8)			N
suninstall(8)	C	SunOS 5.6 をインストールするコマンドはいまでも suninstall だが、インストール手順は完全に変更されている。『Solaris 7 インストールの手引き (SPARC 版)』を参照。	N
sunview(1)	A	SunView は、SunOS 5.6 システムでは OpenWindows に置き換わる。	N
sv_acquire(1)	N		N
sv_release(1)	N		N

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
swapon(8)	A	swap(1M) コマンドが同様の機能を提供。一般に、SunOS 5.6 の swap コマンドのオプションが SunOS 4.x システムの swapon などの個々のスワップ関連コマンドの機能を果たす。	N
swin(1)	N		N
switcher(1)	N		N
symorder(1)	S		N
sync(1)	S		N
sys-unconfig(8)	S		N
syslogd(8)	S		N
t300(1G)	N		S
t300s(1G)	N		S
t4013(1G)	N		S
t450(1G)	N		S
tabs(1V) -SysV	S		N
tail(1)	S		N
talk(1)	S		N
tar(1)	S		N
tbl(1)	S		N
tcopy(1)	S		N
tcov(1)	N	別パッケージの製品として利用できる。	N
tee(1)	S		N

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
tek (1G)	N		S
tektool (1)	N		N
telnet (1C)	S		N
test (1V) -SysV	S		N
test (1V)	C		S
textedit (1)	A	このコマンドは OpenWindows のテキストエディタで利用できる。	N
textedit_filters (1)	A	OpenWindows コマンドの 1 つが OpenWindows のテキストエディタで利用できる。	N
tfsd (8)	N		N
tftp (1C)	S		N
tic (8V)	S		N
time (1V) -SysV	S		N
time (1V)	C	SunOS 4.x では、SunOS 4.x の /usr/5bin/time および SunOS 5.6 とは異なる出力を表示する。SunOS 4.x の time は、経過時間、システムでの所用時間、コマンド実行所用時間を別々の 3 行ではなく、すべてを 1 行に出力する。	N
tip (1C)	S		N
toolplaces (1)	N		N
touch (1V) -SysV	S		N
touch (1V)	C	SunOS 4.x の -f オプションは利用できない。このオプションは、filename の読み書きパーミッションにかかわらず touch を強制しようとする。	S

表 A-10 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
tput(1V) -SysV	S		N
tr(1V) -SysV	S		N
tr(1V)	C		S
trace(1)	A	truss(1) コマンドが同様の機能を提供。	N
traffic(1C)	N		N
troff(1)	S		N
trpt(8C)	N		N
true(1)	S		N
tset(1)	N		S
tsort(1)	S		N
tty(1)	S		N
ttysoftcar(8)	N		N
tunefs(8)	S		N
turnacct(8)	S		N
tvconfig(8)	N		N
tzsetup(8)	N		N

コマンドリファレンス (U, V, W, X, Y, Z)

表 A-11

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
u370 (1)	S		N
u3b (1)	S		N
u3b15 (1)	S		N
u3b2 (1)	S		N
u3b5 (1)	S		N
ul (1)	S		N
umask (1)	S		N
umount (8)	C	インタフェースは2つのバージョン間でかなり異なる。SunOS 5.6 コマンドでは、大部分のオプションは変更されており、ファイルシステムに固有のオプションとして与える必要がある。	N
umount_tfs (8)	N		N
unadv (8)	N	RFS は利用できない。	N
uname (1)	S		N
uncompress (1)	S		N
unconfigure (8)	N		N
unexpand (1)	S		N
unget (1)	C	バージョン間で、読み取り不能な <code>s.file</code> を処理する方法が異なる。SunOS 4.x では、読み取り不能な <code>s.file</code> が出現すると、エラーを出力して続行する。SunOS 5.6 では表示せずにエラーを無視する。	N
unifdef (1)	S		N
uniq (1)	S		N

表 A-11 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
units(1)	S		N
unix2dos(1)	S		N
unlink(8V)	S		N
unpack(1V) -SysV	S		N
unpack(1V) -SysV	C	SunOS 4.x の /usr/5bin/pack コマンドでは、ファイル名は 12 文字に制限される。SunOS 5.6 では、{NAME_MAX} - 2 に制限される。SunOS 5.6 の pack および unpack コマンドは SunOS 4.x コマンドと互換性がある。	
unwhiteout(1)	N		N
update(8)	A	fsflush(1) コマンドがこの機能を提供。	N
uptime(1)	A	who -u コマンドが同様の機能を提供。	S
users(1)	A	who -q が同様の機能を提供。	S
ustar(1V)	A	tar(1) コマンドが同様の機能を提供。	N
uuccheck(8C)	S		N
uucico(8C)	S		N
uucleanup(8C)	S		N
uucp(1C)	S		N
uudecode(1C)	S		N
uuencode(1C)	S		N
uulog(1C)	C	指定ユーザ名で行われる処理に関する情報を出力できる -u オプションはサポートされない。	N

表 A-11 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
uuname(1C)	S		N
uupick(1C)	S		N
uusched(8C)	S		N
uusend(1C)	N		N
uustat(1C)	S		N
uuto(1C)	S		N
uux(1C)	S		N
uuxqt(8C)	S		N
vacation(1)	S		N
val(1)	S		N
vax(1)	S		N
vedit(1)	S		N
vfontinfo(1)	N		N
vgrind(1)	S		N
vi(1)	S		N
view(1)	S		N
vipw(8)	N		S
vmstat(8)	C	-f オプションは利用できない。	N
vplot(1)	N		S
vswap(1)	N		N

表 A-11 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
vtroff(1)	N		N
vwidth(1)	N		N
w(1)	S		N
wait(1)	S		N
wall(1)	S		N
wc(1)	S		N
what(1)	S		N
whatis(1)	C		N
whereis(1)	N		S
which(1)	S		N
who(1)	S		N
whoami(1)	A	id(1) コマンドが同様の機能を提供。id コマンドは、ユーザ名だけでなくユーザ名とユーザおよびグループ ID を出力する。	S
whois(1)	S		N
write(1)	S		N
xargs(1V) -SysV	S		N
xget(1)	N		N
xsend(1)a	N		N
xstr(1)	S		N
yacc(1)	S		N

表 A-11 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
yes (1)	N		N
ypbatchupd (8C)	N		N
ypbind (8)	S	現在は /usr/lib/netsvc/yp にある。	N
ypcat (1)	S		N
ypinit (8)	S		N
ypmatch (1)	S		N
yppasswd (1)	S	yppasswd コマンドは、NIS サーバのパスワード情報にアクセスするために SunOS 5.6 システムでまだ利用できる。NIS+ データベースの同機能のコマンドは nispasswd (1) である。passwd (1) コマンドはサポートされているすべてのデータベース (NIS、NIS+、ファイル) のパスワードを処理できる。	N
yppoll (8)	S		N
yppush (8)	N		N
ypserv (8)	N		N
ypset (8)	S		N
ypupdated (8C)	N		N
ypwhich (8)	S		N
ypxfr (8)	S	現在は /usr/lib/netsvc/yp にある。	N
ypxfrd (8)	S		N
zcat (1)	S		N

表 A-11 続く

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	BSD
zdump(8)	S		N
zic(8)	S		N

システムコールリファレンス

この付録では、表形式ですべての SunOS 4.x を示し、Solaris 7、ABI、SVID、SVR4、および SunOS/BSD ソース互換パッケージのそれぞれの環境での利用方法の違いを示します。

リファレンスの使い方

- インタフェースに C (変更) とマークされている場合、SunOS 4.x コマンドと Solaris 7 コマンドとの違いについて簡単な説明があります。
- インタフェースに S (同じ) とマークされている場合、Solaris 7 インタフェースは SunOS 4.x インタフェースの機能をすべてサポートします。場合によってはインタフェースは拡張されていますが、これは SunOS 4.x インタフェースの完全なスーパーセットと考えることができます。現在は多くのシステムコールがライブラリルーチンとして利用できます。注の欄には新しいルーチンのマニュアルページの参照先を示してあります。
- インタフェースに A (置換) とマークされている場合、注の欄を読んでください。
- インタフェースに N (利用不可) とマークされている場合、そのインタフェースは利用できません。
- 標準ではサポートされていない errno 値をインタフェースが含む場合、「#」で示します。errno の違いは必ずしも互換性を損ないません。ただし、EDQUOT、EFAULT、EIO は ABI または SVID に関して表示されていないことがあります。errno 値は必要に応じて ABI または SVID 準拠システムによりサポートされます。

SunOS 4.x には、System V 互換バージョンの多くのコマンド、システムコール、およびルーチンを提供する System V のインストールオプションがあります。System V インタフェースはこのあとに続く表に一覧表示されています。SunOS 4.x インタフェースの System V バージョンを指す場合には、「SysV」とインタフェースの横に示してあります。

すべての Solaris 7 インタフェースに関する詳細な情報については、『SunOS Reference Manual』のセクション 2「System Calls」を参照してください。

注 - システムコールは関数です。この付録では、関数名のすぐ後に空の () をつけて関数を識別しています。数字を入れた () がさらに続く場合、これは関連するマニュアルページのセクションを示します。

例

以下の表にエントリの例と、その説明を示します。

SunOS 4.x システムコール	SunOS 5.6	利用可能な代替システムコールと注	ABI	SVID	SVR4	BSD
mctl() (2)	A	memcntl() (2) が同様の機能を提供する。	A	A	A	S

mctl() システムコールは、ABI、SVID、SVR4、および SunOS 5.6 では利用できません。memcntl() コールを使用するには、このシステムコールを使用するアプリケーションをすべて書き直さなければなりません。mctl() は SunOS/BSD 互換パッケージで利用できますが、使用するアプリケーションは他の SVR4 システムと互換性がありません。

SunOS 4.x システムコール	SunOS 5.6	利用可能な代替システムコールと注	ABI	SVID	SVR4	BSD
getsockname() (2)	S#	SunOS 4.x の getsockname() が使用する errno 値 ENOBUFS は、SVR4 と SunOS 5.6 では ENOSR に変更された。	N	N	S#	N

getsockname() システムコールは、ABI または SVID では定義されていません。SunOS 5.6 と SVR4 における getsockname() は SunOS 4.x のものと同じですが、SunOS 4.x ではエラー条件について errno を ENOBUFS に設定していたのに対し、SunOS 5.6 では errno を ENOSR に設定するという点が異なります。

システムコールのリファレンス (A, B, C, D, E, F, G)

表 B-1

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
accept() (2)	S	現在は accept() (3N)	N	N	S	N
access() (2V) -SysV	S		S	S	S	N
acct() (2)	C#	シンボリック 名、ACOMPAT、ACORE、AXSIG は、SunOS 4.x の (<sys/acct.h> に定義されている) acct 構造体メンバ ac_flag には有効だ が、SunOS 5.6、ABI、SVID、SVR4 では無 効。また、アカウント記録のフォー マットも、SunOS 4.x と、SunOS 5.6、 ABI、SVID、SVR4 との間で異なる。	C#	C#	C#	N
adjtime() (2)	S		N	S	S	N
async_daemon() (2)	N		N	N	N	N
audit() (2)	N		N	N	N	N
auditon() (2)	N		N	N	N	N
auditsvc() (2)	N		N	N	N	N
bind() (2)	S	現在は bind() (3N)	N	N	S	N
brk() (2)	S		N	N	S	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
chdir() (2V) -SysV	S		S	S	S	N
chmod() (2V) -SysV	C#	シンボリックアクセスモード (<sys/stat.h>)、S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は SunOS 4.x の chmod() によってサポートさ5.6、ABI、SVID、SVR4 ではサポートされない。 ただし、同等の SunOS 5.6、または ABI、SVID、SVR4 のシンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) は同じ定義を持ち、SunOS 4.x の <sys/stat.h> でも定義されているので使用できる。	C#	C#	C#	N
chown() (2V)	C	SunOS 4.x では、chown() の owner および group 引数は int 型になる。SunOS 5.6、ABI、SVID、SVR4 では、owner は uid_t 型、group は gid_t 型になる。 SunOS 4.x では、path の最後の指定がシンボリックリンクならば、シンボリックリンクの所有権は変更された。SunOS 5.6 では、chown() はシンボリックリンクによって参照されるファイルまたはディレクトリの所有権を変更する。シンボリックリンクの所有権を変更するには SunOS 5.6 の lchown() (2) を使用する。	C	C	C	N
chown() (2V) -SysV	S		S	S	S	N
chroot() (2)	S		S	S	S	N
close() (2V) -SysV	S		S	S	S	N
connect() (2)	S#	現在はconnect() (3N)。	N	N	S#	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
creat() (2V)	C#	<p>SunOS 4.x では、creat() の mode 引数は int 型で、SunOS 5.6、ABI、SVID、SVR4 では、mode 引数は mode_t 型。また、SunOS 5.6、ABI、SVID、SVR4 では <fcntl.h> に含まれるが、SunOS 4.x では含まれない。シンボリックアクセスモード (<sys/stat.h>)、S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は SunOS 4.x の creat() によってサポートされるが、SunOS 5.6、ABI、SVID、SVR4 ではサポートされない。</p> <p>ただし、同等の SunOS 5.6、または ABI、SVID、SVR4 のシンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) は同じ定義を持ち、SunOS 4.x の <sys/stat.h> でも定義されているので使用できる。errno フラグの ENXIO、EOPNOTSUPP は、SunOS 4.x の chmod 関数には有効だが、SunOS 5.6、ABI、SVID、SVR4 では無効。</p>	C#	C#	C#	N
creat() (2V) -SysV	C#	<p>シンボリックアクセスモード (<sys/stat.h>)、S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は SunOS 4.x の creat() によってサポートされるが、SunOS 5.6、ABI、SVID、SVR4 ではサポートされない。ただし、同等の SunOS 5.6、または ABI、SVID、SVR4 のシンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) は同じ定義を持ち、SunOS 4.x の <sys/stat.h> でも定義されているので使用できる。errno フラグの ENXIO、EOPNOTSUPP は、SunOS 4.x の chmod 関数には有効だが、SunOS 5.6、ABI、SVID、SVR4 では無効。</p>	C#	C#	C#	N
dup() (2V) -SysV	S		S	S	S	N
dup2() (2V) -SysV	S	現在は dup2() (3C)。	S	S	S	N
_exit() (2V) -SysV	S		S	S	S	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
execve() (2V) -SysV	S		S	S	S	N
fchdir() (2V) -SysV	S		S	S	S	N
fchmod() (2V) -SysV	C	シンボリックアクセスモード (<sys/stat.h> の S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は SunOS 4.x の fchmod() によりサポートされるが、SunOS 5.6、ABI、SVID、SVR4 ではサポートされない。ただし、同等5.6、または ABI、SVID、SVR4 のシンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) は同じ定義を持ち、SunOS 4.x の <sys/stat.h> でも定義されているので使用できる。	C	C	C	N
fchown() (2)	S		S	S	S	N
fchroot() (2)	S		N	N	N	N
fcntl() (2V) -SysV	C	SunOS 4.x では、フラグ -O_APPEND、-O_SYNC、-O_NDELAY、また、<sys/file.h> に定義があるフラグ -FASYNC、-FNDELAY、-FNBIO が F_SETFL コマンドで有効となる。SunOS 5.6、ABI、SVID、SVR4 はフラグ -O_APPEND、-O_SYNC、-O_NDELAY、および -O_NONBLOCK のみサポートする。-O_SYNC は -FASYNC の代わりに使用できる。-O_NONBLOCK は -FNDELAY と -FNBIO の代わりに使用できる。また、-O_NONBLOCK は -O_NDELAY の代わりに使用できる。-O_NDELAY は削除されている。SunOS 4.x の F_GETOWN および F_SETOWN コマンドは SunOS 5.6、ABI、SVID、SVR4 ではサポートされない。	C	C	C	N
flock() (2)	N		N	N	N	S

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
fork() (2V)	C	SunOS 4.x では、fork() は int 型の値を返す。SunOS 5.6、ABI、SVID、SVR4 では、fork() は pid_t 型の値を返す。また、SunOS 5.6、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
fork() (2V) -SysV	S		S	S	S	N
fpathconf() (2V) -SysV	S		S	S	S	N
fstat() (2V) -SysV	S		S	S	S	N
fstatfs() (2)	A	fstatvfs() (2) に同等の機能がある。	A	A	A	S
fsync() (2)	S		S	S	S	N
ftruncate() (2)	S	現在はftruncate() (3C)	N	N	S	N
getauid() (2)	N		N	N	N	N
getdents() (2)	S		N	N	S	N
getdirentries() (2)	A	getdents() (2) に同等の機能がある。	N	N	N	N
getdomainname() (2)	A	sysinfo() (2) に同等の機能がある。	N	N	N	N
getdtablesize() (2)	A	現在は getdtablesize() (3C)。getrlimit() で resource 引数を RLIMIT_NOFILE に設定 すると同様の機能が使用できる。	A	A	A	S
getegid() (2V)	C	SunOS 4.x では、getegid() は int 型の値を返す。SunOS 5.6、ABI、SVID、SVR4 では、getegid() は gid_t 型の値を返す。また、SunOS 5.6、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getegid() (2V) -SysV	S		S	S	S	N
geteuid() (2V)	C	SunOS 4.x では、geteuid() は int 型の値を返す。SunOS 5.6、ABI、SVID、SVR4 では、geteuid() は uid_t 型の値を返す。また、SunOS 5.6、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
geteuid() (2V) -SysV	S		S	S	S	N
getgid() (2V)	C	SunOS 4.x では、getgid() は int 型の値を返す。SunOS 5.6、ABI、SVID、SVR4 では、getgid() は gid_t 型の値を返す。また、SunOS 5.6、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
getgid() (2V) -SysV	S		S	S	S	N
getgroups() (2V)	C	SunOS 4.x では、getgroups() の gidset 引数は int 型で、SunOS 5.6、ABI、SVID、SVR4 では grouplist 引数は gid_t 型である。また、SunOS 5.6、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
getgroups() (2V) -SysV	S		S	S	S	N
gethostid() (2)	A	現在は gethostid() (3C)。sysinfo() (2) で command 引数を SI_HW_SERIAL に設定すると同様の機能を使用できる。	N	N	N	S
gethostname() (2)	A	現在は gethostname() (3C)。sysinfo() (SI_HOSTNAME, name, namelen) ルーチンが同様の機能を提供する。	N	N	N	S
getitimer() (2)	S		N	S	S	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getmsg() (2)	S		S	S	S	N
getpagesize() (2)	A	現在は getpagesize() (3C)。sysconf() (3C) が同様の機能を提供する。	A	A	A	S
getpeername() (2)	S#	現在は getpeername() (3N)。errno フラ グの ENOBUFS は SunOS 4.x の getpeername() には有効だが、SVR4 と SunOS 5.6 では無効。	N	N	S#	N
getpgid() (2V)	S		S	S	S	N
getpgrp() (2V)	C	SunOS 4.x の getpgrp() には引数 <i>pid</i> があ り、getpgrp() は <i>pid</i> によって示されるプ ロセスのプロセスグループを返 す。getpgrp() の SunOS 5.6、ABI、 SVID、SVR4 は引数を受け付け ず、getpgrp() は呼び出し元のプロセスグ ループ ID を返す。また、SunOS 4.x getpgrp() は int 型の値を返すが、 SunOS 5.6、ABI、SVID、または SVR4 の getpgrp() は pid_t 型の値を返す。 SunOS 5.6、ABI、SVID、および SVR4 では <unistd.h> と <sys/types.h> に含まれ るが、SunOS 4.x では含まれない。	C	C	C	N
getpgrp() (2V) -SysV	S		S	S	S	N
getpid() (2V)	C	SunOS 4.x では、getpid() は int 型の値 を返す。SunOS 5.6、ABI、SVID、または SVR4 では、getpid() は pid_t 型の値を 返す。また、SunOS 5.6、ABI、SVID、およ び SVR4 では <unistd.h> と <sys/ types.h> に含まれるが、SunOS 4.x では 含まれない。	C	C	C	N
getpid() (2V) -SysV	S		S	S	S	N

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getppid() (2V)	C	SunOS 4.x では、getppid() は int 型の値を5.6、ABI、SVID、または SVR4 では、getppid() は pid_t 型の値を返す。また、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
getppid() (2V) -SysV	S		S	S	S	N
getpriority() (2)	A	現在は getpriority() (3C)。priosetl() (2) が同様の機能を提供する。	A	A	A	S
getrlimit() (2)	C	RLIMIT_RSS は SunOS 4.x でサポートされているリソースの 1 つで (プロセスの常駐セットサイズが増大できるバイト単位の最大サイズ)、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされていない。SunOS 5.6、ABI、SVID、または SVR4 では、さらに RLIMIT_AS リソース、つまりバイト単位で定義されるプロセスのアドレス空間の最大サイズもサポートする。また、SunOS 5.6、ABI、SVID、または SVR4 での、rlimit 構造体にある rlim_cur (現在のソフト限界) および rlim_max (ハード限界) フィールドは、SunOS 4.x では int 型ではなく rlim_t 型である。	C	C	C	N
getrusage() (2)	A	現在は getusage() (3C)。	N	N	N	C
getsockname() (2)	S#	SunOS 4.x の getsockname() が使用する errno 値 ENOBUFS は、SVR4 と SunOS 5.6 では ENOSR に変更された。	N	N	S#	N
getsockopt() (2)	S	現在は getsockopt() (3N)	N	N	S	N
gettimeofday() (2)	S	現在は gettimeofday() (3C)	N	S	S	S

表 B-1 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getuid() (2V)	C	SunOS 4.x では、getuid() は int 型の値を返す。SunOS 5.6、ABI、SVID、または SVR4 では、getuid() は type uid_t の値を返す。また、SunOS 5.6、ABI、SVID、および SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
getuid() (2V) -SysV	S		S	S	S	N

システムコールのリファレンス (I, K, L, M, O, P, Q)

表 B-2

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ioctl() (2)	C	163ページの「ioctl() 要求」の ioctl() 要求を参照のこと。	C	C	C	N
kill() (2V)	C	SunOS 4.x では、プロセスのグループにシグナルが送られ (同様に、pid が 0 または負であれば)、そしてシグナルを送るプロセスがそのグループのメンバであれば、そのシグナルは送り元のプロセスには送られない。SunOS 5.6、ABI、SVID、または SVR4 では、シグナルは送り元のプロセスにも送られる。SunOS 4.x では、pid 引数は int 型で、SunOS 5.6、ABI、SVID、または SVR4 では、pid 引数は pid_t 型である。また、SunOS 5.6、ABI、SVID、SVR4 では <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
kill() (2V) -SysV	S		S	S	S	N
killpg() (2)	A	現在では killpg() (3C)。kill() (2) が同様の機能を提供する。killpg() (pgrp, sig) を kill() (-pgrp, sig) に置き換える。	A	A	A	S
link() (2V) -SysV	C	SunOS 5.6、ABI、SVID、または SVR4 の link() では、最初の引数の最後の部分がシンボリックリンクならば、リンクをたどらず、シンボリックリンクへのハードリンクが作成される。	C	C	C	N
listen() (2)	S	現在は listen() (3N)	N	N	S	N
lseek() (2V) -SysV	S		S	S	S	N
lstat() (2V) -SysV	S		S	S	S	N
mctl() (2)	A	memcntl() (2) が同様の機能を提供する。	A	A	A	S
mincore() (2)	C	SunOS 4.x では、引数 len は int 型で、SVR4 と SunOS 5.6 では、引数 len は、unsigned int 型となるように定義される size_t 型である。また、SunOS 5.6 では <unistd.h> をインクルードする必要がある。	N	N	C	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
mkdir() (2V)	C	<p>SunOS 4.x では、モード引数は <i>int</i> 型で、SunOS 5.6、ABI、SVID、または SVR4 ではモード引数は <i>mode_t</i> 型である。また、SunOS 5.6、あるいは ABI、SVID、SVR4 では <code><sys/types.h></code> と <code><sys/stat.h></code> に含まれるが、SunOS 4.x では含まれない。</p> <p>シンボリックアクセスモード (<code><sys/stat.h></code>) の <code>S_IREAD</code> (00400)、<code>S_IWRITE</code> (00200)、<code>S_IEXEC</code> (00100) は SunOS 4.x の <code>mkdir()</code> によってサポートされるが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.6、ABI、SVID、および SVR4 のシンボリックアクセスモードである <code>S_IRUSR</code> (00400)、<code>S_IWUSR</code> (00200)、<code>S_IXUSR</code> (00100) には同じ定義があり、SunOS 4.x の <code><sys/stat.h></code> にも定義があるため、使用することができる。</p>	C	C	C	N
mkdir() (2V) -SysV	C	<p>シンボリックアクセスモード (<code><sys/stat.h></code>) の <code>S_IREAD</code> (00400)、<code>S_IWRITE</code> (00200)、<code>S_IEXEC</code> (00100) は SunOS 4.x の <code>mkdir()</code> によってサポートされるが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.6、ABI、SVID、および SVR4 のシンボリックアクセスモードである <code>S_IRUSR</code> (00400)、<code>S_IWUSR</code> (00200)、<code>S_IXUSR</code> (00100) には同じ定義があり、SunOS 4.x の <code><sys/stat.h></code> にも定義があるため、使用することができる。</p>	C	C	C	N
mkfifo() (2V) -SysV	S	現在は <code>mkfifo()</code> (3C)	S	S	S	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
mknod() (2V) -SysV	C	mknod() の <i>mode</i> 引数は、SunOS 4.x では <i>int</i> 型で、SunOS 5.6、ABI、SVID、または SVR4 では <i>mode_t</i> 型である。 <i>dev</i> 引数は、SunOS 4.x では <i>int</i> 型で、SunOS 5.6、ABI、SVID、または SVR4 では <i>dev_t</i> 型である。シンボリックアクセスモード (<sys/stat.h> の S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は、SunOS 4.x の mknod() によってサポートされるが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.6、ABI、SVID、または SVR4 のシンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) には同じ定義があり、SunOS 4.x の <sys/stat.h> にも定義があるため、使用することができる。	C	C	C	N
mmap() (2)	C	SunOS 4.x では、 <i>-mmap</i> フラグオプションの値に、<sys/mman.h> で定義されている MAP_TYPE が含まれており、この MAP_TYPE は SunOS 5.6、ABI、SVID、または SVR4 の <sys/mman.h> では定義されていない。	C	C	C	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
mount () (2)	C#	<p>SunOS 4.x の mount () と、SunOS 5.6、ABI、SVID、SVR4 の mount () は、多くの点で互換性がない。SunOS 4.x の最初の引数 <i>type</i> はファイルシステム形式の名前だが、SunOS 5.6、ABI、SVID、または SVR4 では、最初の引数は <i>fs</i> であり、ファイルシステムの名前である。SunOS 5.6、ABI、SVID、または SVR4 では、ファイルシステム形式の名前 <i>fstype</i> は、mount () の第 4 引数となる。SunOS 4.x では、形式に固有な引数を渡すのに 1 つのパラメータ (第 4 の引数である <i>caddr_t data</i>) を使用するが、SunOS 5.6、ABI、SVID、または SVR4 では 2 つのパラメータを使用する (5 と 6、つまり <i>const char *dataptr</i> と <i>int datalen</i>)。また、SunOS 5.6、ABI、SVID、SVR4 では <code><sys/mount.h></code> の前にある <code><sys/types.h></code> に含まれるが、SunOS 4.x では含まれない。</p> <p>SunOS 4.x の <code><sys/mount.h></code> は、SunOS 5.6、ABI、SVID、または SVR4 の <code><sys/mount.h></code> で定義されない mount () <i>flags</i> 引数 (<code>M_NEWTYPE</code>, <code>M_RDONLY</code>, <code>M_NOSUID</code>, <code>M_NEWTYPE</code>, <code>M_GRPID</code>, <code>M_REMOUNT</code>, <code>M_NOSUB</code>, <code>M_MULTI</code>) に対するシンボリック定数を定義する。かわりに、<code>M_RDONLY</code> を <code>MS_RDONLY</code> に、<code>M_NOSUID</code> を <code>MS_NOSUID</code> に、<code>M_REMOUNT</code> を <code>MS_REMOUNT</code> に置き換える必要がある。<code>M_NEWTYPE</code> フラグは、SunOS 4.x の mount () に固有であり、SunOS 5.6、ABI、SVID、または SVR4 では置き換える必要はない。フラグ、<code>M_NOSUB</code>、<code>M_GRPID</code>、<code>M_MULTI</code> の機能は、<code><sys/mount.h></code> に定義があり、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。</p> <p>SunOS 4.x の mount () は、SunOS 5.6、ABI、SVID、または SVR4 によって返されない <code>errno</code> 値、<code>ENODEV</code>、<code>EACCES</code>、<code>EMFILE</code>、<code>ENOMEM</code> を使用する。</p>	C#	C#	C#	N
mprotect () (2)	S		S	S	S	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
msgctl () (2)	S		S	S	S	N
msgget () (2)	S		S	S	S	N
msgrcv () (2)	S		S	S	S	N
msgsnd () (2)	S		S	S	S	N
msync () (2)	S		S#	S#	S	N
munmap () (2)	S		S	S	S	N
nfssvc () (2)	A	このインタフェース は、nfssys () (NFS_SVC,...)ルーチンによ り、SunOS 5.6 で置き換えられる。	N	N	N	N
open () (2V)	C#	open () への <i>mode</i> 引数は、SunOS 4.x ソフ トウェアでは <i>int</i> 型で、SunOS 5.6、ABI、 SVID、または SVR4 では <i>mode_t</i> 型であ る。SunOS 4.x では、 <i>path</i> 引数が空の文字列 ならば、カーネルは、この空のパス名をカ レントディレクトリ「.」にマップする。 SunOS 5.6、ABI、SVID、または SVR4 で は、 <i>path</i> が空の文字列を指す場合、エラー が発生する。SunOS 4.x では、 <i>O_NDELAY</i> または <i>O_NONBLOCK</i> フラグがオープンする コールに設定されると、open () コール自 身だけが有効にされる。SunOS 5.6、ABI、 SVID、または SVR4 では、 <i>O_NDELAY</i> また は <i>O_NONBLOCK</i> フラグが open () へのコ ールに設定されると、そのファイル記述子に 同等のフラグが設定され、その記述子への 次の読み書きはブロックされない。 また、SunOS 5.6、ABI、SVID、SVR4 では <sys/types.h> と <sys/stat.h> に含ま れるが、SunOS 4.x では含まれない。 errno 値 EOPNOTSUPP は、SunOS 4.x の open () には有効だが、SunOS 5.6、ABI、 SVID、または SVR4 では返されない。	C#	C#	C#	N

表 B-2 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
open() (2V) -SysV	S#	errno 値 EOPNOTSUPP は、SunOS 4.x の open() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では返されない。	S#	S#	S#	N
pathconf() (2V) -SysV	S		S	S	S	N
pipe() (2V) -SysV	S		S	S	S	N
poll() (2)	S		S	S	S	N
profil() (2)	S		S	S	S	N
ptrace() (2)	C#	<p>SunOS 4.x の ptrace() へのオプションの <i>addr2</i> 引数は、SunOS 5.6 ルーチンではサポートされない。ptrace() への要求引数は、SunOS 4.x では enum ptracereq 型で、SunOS 5.6 では int() 型。ptrace() への <i>pid</i> 引数は、SunOS 4.x では int() 型で、SunOS 5.6 では pid_t() 型。</p> <p>また、SunOS 4.x では <sys/types.h> に含まれるが、SunOS 5.6 では <signal.h>、<sys/ptrace.h>、および <sys/wait.h> に含まれる。</p> <p>errno フラグの EPERM は、SunOS 4.x の ptrace() には有効だが、SunOS 5.6 では無効。</p> <p>有効な <i>request</i> 値に関する情報については、166ページの「ptrace() 要求値」を参照のこと。</p>	C#	C#	C#	N
putmsg() (2)	S		S	S	S	N
quotactl() (2)	A	Q_QUOTACTL ioctl() が同様の機能を提供する。	A	A	A	N

システムコールのリファレンス (R, S, T, U, V)

表 B-3

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
read() (2V)	C#	errno フラグの EISDIR、EWOULDBLOCK は、SunOS 4.x の read() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
read() (2V) -SysV	C#	read() の nbyte 引数は、SunOS 4.x では int() 型で、SunOS 5.6 では unsigned() 型である。 SunOS 5.6 の read() は、(FIONBIO ioctl() 要求、あるいは <sys/file.h> から FNDELAY フラグまたは 4.x BSD 環境の <fcntl.h> から O_NDELAY フラグを使った fcntl(2V) () へのコールがある) BSD 4.2 スタイル非ブロック I/O を、SunOS 4.x の read() ルーチンのようにはサポートしない。 errno フラグの EISDIR、EWOULDBLOCK は、SunOS 4.x の read() には有効だが、SunOS 5.6 では無効。	C#	C#	C#	N
readlink() (2)	S		S	S	S	N
readv() (2)	C#	errno フラグの EISDIR、EWOULDBLOCK は、SunOS 4.x には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
readv() (2V) -SysV	C#	SunOS 4.x、SunOS 5.6、SVID または SVR4 の (<sys/uio.h> に定義がある) iovec 構造体はわずかに異なる。SunOS 4.x の iovec の iov_len フィールドは、integer ₁ として定義されるが、SunOS 5.6、SVID または SVR4 の iov_len は unsigned ₁ として定義される。SunOS 5.6、SVID または SVR4 の readv() システムコールは、BSD 4.2 スタイルの非ブロック I/O を、SunOS 4.x のようにはサポートしない。	C#	C#	C#	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
reboot() (2)	A	現在は reboot() (3C)。uadmin() (2) が同様の機能を提供する。	N	N	N	S
recv() (2)	S	現在は recv() (3N)	N	N	S	N
recvfrom() (2)	S	現在は recvfrom() (3N)	N	N	S	N
recvmsg() (2)	S	現在は recvmsg() (3N)	N	N	S	N
rename() (2V) -SysV	S#	SunOS 5.6、ABI、SVID、または SVR4 では <unistd.h> に含まれるが、SunOS 4.x では含まれない。errno フラグ ENOTEMPTY は、SunOS 4.x の rename() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。SunOS 5.6、ABI、SVID、または SVR4 は、そのかわりにフラグ EEXIST に対して errno を設定する。	S#	S#	S#	N
rmdir() (2V) -SysV	S#	SunOS 5.6、ABI、SVID、または SVR4 では <unistd.h> に含まれるが、SunOS 4.x では含まれない。errno フラグ ENOTEMPTY は、SunOS 4.x の rmdir() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。SunOS 5.6、ABI、SVID、または SVR4 は、そのかわりにフラグ EEXIST に対して errno を設定する。	S#	S#	S#	N
sbrk() (2)	S		N	N	S	N
select() (2)	S	現在は select() (3C)。	N	N	S	N
semctl() (2)	S		S	S	S	N
semget() (2)	S		S	S	S	N
semop() (2)	S		S	S	S	N
send() (2)	S#	現在は send() (3N)。 errno フラグの ENOBUFS は、SunOS 4.x の send() (2) には有効だが、SVR4、および SunOS 5.6 では無効。	N	N	S#	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
sendmsg(2)()	S#	現在は sendmsg() (3N)。 errno フラグの ENOBUFS は、SunOS 4.x の sendmsg() (2) には有効だが、SVR4 と SunOS 5.6 では無効。	N	N	S#	N
sendto() (2)	S#	現在は sendto() (3N)。 errno フラグの ENOBUFS は、SunOS 4.x の sendto() (2) には有効だが、SVR4 と SunOS 5.6 では無効。	N	N	S#	N
setaudit() (2)	N		N	N	N	N
setaudit() (2)	N		N	N	N	N
setdomainname() (2)	A	sysinfo() (2) が同様の機能を提供する。	N	N	N	N
setgroups() (2V)	C	SunOS 4.x では、gidset 引数は int 型で、SunOS 5.6、ABI、SVID、または SVR4 では、gidset 引数は gid_t 型になっている。また、SunOS 5.6、ABI、SVID、または SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。	C	C	C	N
setgroups() (2V) -SysV	S		S	S	S	N
sethostname() (2)	A	現在は sethostname() (3C)。sysinfo() (2) のコマンド引数を SI_SET_HOSTNAME に設定すると、同様の機能を提供する。	N	N	N	S
setitimer() (2)	S		N	S	S	N
setpgid() (2V) -SysV	S		S	S	S	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
setpgrp() (2V)	C	SunOS 4.x の setpgrp() には引数 <i>pid</i> と <i>pgrp</i> があり、setpgrp() はプロセスグループを <i>pid</i> によって示されるプロセスの <i>pgrp</i> に設定する。SunOS 5.6 の setpgrp() は引数を受け付けず、setpgrp() は新しいセッションも作成する。ただし、 <i>pgrp</i> がゼロであり、 <i>pid</i> が呼び出し元プロセスを参照すると、SunOS 4.x の setpgrp() コールは引数なしの SunOS 5.6 の setpgrp() コールと同じになる。また、SunOS 4.x の setpgrp() は int 型の値を返すが、SunOS 5.6 では、setpgrp() は pid_t 型の値を返す。さらに、SunOS 5.6 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4.x では含まれない。errno フラグの EACCES、EINVAL、ESRCH は、SunOS 4.x の setpgrp() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。 ()	C#	C#	C#	N
setpgrp() (2V) -SysV	S	errno フラグの EACCES、EINVAL、ESRCH は SunOS 4.x の setpgrp() (2V) には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。	S	S	S	N
setpriority() (2)	A	現在は setpriority() (3C)。prIOCNTL() (2) が同様の機能を提供する。	A	A	A	S
setregid() (2)	S	現在は setregid() (3C)。	N	N	N	C
setreuid() (2)	S	現在は setreuid() (3C)。	N	N	N	C
setrlimit() (2)	C	現在は setrlimit() (2)。	C	C	C	N
setsid() (2V) -SysV	S		S	S	S	N
setsockopt() (2)	S	現在は setsockopt() (2)。	N	N	S	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
settimeofday() (2)	S	現在は settimeofday() (2)。	N	S	S	S
setuseraudit() (2)	N		N	N	N	N
sgetl() (2)	S	現在は xdr_simple() (3N)。	N	S	S	N
shmat() (2)	S		S	S	S	N
shmctl() (2)	S		S	S	S	N
shmdt() (2)	S		S	S	S	N
shmget() (2)	S		S	S	S	N
shutdown() (2)	S	現在は shutdown() (3N)。	N	N	S	N
sigaction() (2)	C	Solaris 7 には SA_RESTART というフラグがある。このフラグを使用すると、このシグナルのハンドラの実行により割り込みされる関数がシステムにより透過的に再起動される。	N	C	C	S
sigblock() (2)	A	how 引数を SIG_BLOCK に設定した sigprocmask() (2) が同様の機能を提供する。	A	A	A	S
sigmask() (2)	A	sigsetops() (3C) ルーチンは、SunOS 5.6、ABI、SVID、または SVR4 が同様の機能を提供する。	A	A	A	S
sigpause() (2V) -SysV	S	SunOS 4.x の sigpause() は、その引数 (sigmask) をマスクされたシグナルのセットに代入するが、ABI と SVID の sigpause() は、その引数 (sig) を呼び出し元プロセスのシグナルマスクから削除する。SVR4 と SunOS 5.6 の sigpause() は、SunOS 4.x の sigpause() (2) と互換性がある。	C	C	S	S
sigpending() (2V) -SysV	S		S	S	S	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
sigprocmask() (2V) -SysV	S		S	S	S	N
sigsetmask() (2)	A	<i>how</i> 引数を SIG_SETMASK に設定した sigprocmask() (2) ルーチンは、SunOS 5.6、ABI、SVID、または SVR4 が同様の機能を提供する。	A	A	A	S
sigstack() (2)	A	現在は sigstack() (3C)。sigaltstack(2)() が同様の機能を提供する。	A	A	A	S
sigsuspend() (2V) -SysV	S		S	S	S	N
sigvec() (2)	A	sigaction() (2) が同様の機能を提供する。	A	A	A	S
socket() (2)	C#	現在は socket() (3N) SunOS 4.x では、PF_IMPIPKN はサポートされた <i>domain</i> だが、SVR4 と SunOS 5.6 では PF_IMPIPKN はサポートされない。errno フラグの ENOBUFS、EPROTOTYPE は、SunOS 4.x の socket() には有効だが、SVR4 と SunOS 5.6 では無効。	N	N	C#	N
socketpair() (2)	S	現在は socketpair() (3N)	N	N	S	N
sput1() (2)	S	現在は xdr_simple() (3N)	N	S	S	N
stat() (2V) -SysV	S		S	S	S	N
statfs() (2)	A	statvfs() (2) が同様の機能を提供する。	A	A	A	N
swapon() (2)	A	swapctl() (2) が同様の機能を提供する。	N	N	N	N
symlink() (2)	S		S	S	S	N
sync() (2)	S		S	S	S	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
syscall() (2)	N		N	N	N	S
sysconf() (2V) -SysV	S	現在は sysconf() (3C)	S	S	S	N
tell() (2V) -SysV	S		S	N	S	N
truncate() (2)	S	現在は truncate() (23C)	N	N	S	N
umask(2V) -SysV()	C	シンボリックアクセスモード (<sys/stat.h>) の S_IREAD (00400)、S_IWRITE (00200)、S_IEXEC (00100) は、SunOS 4.x の umask() によってサポートされるが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.6、ABI、SVID、または SVR4 シンボリックアクセスモードである S_IRUSR (00400)、S_IWUSR (00200)、S_IXUSR (00100) には同じ定義があり、SunOS 4.x の <sys/stat.h> にも定義があるため、使用することができる。	C	C	C	N
umount() (2V) -SysV	S		S	S	S	N
uname() (2V) -SysV	S		S	S	S	N
unlink() (2V) -SysV	S		S	S	S	N
unmount() (2)	A	umount() (2) が同様の機能を提供する。	A	A	A	N
ustat() (2)	S		S	S	S	N
utimes() (2)	S		N	N	N	N
vadvise() (2)	N		N	N	N	N

表 B-3 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>vfork()</code> (2)	S		N	N	S	N
<code>vhangup()</code> (2)	S		N	N	N	N

システムコールのリファレンス (W)

表 B-4

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>WEXITSTATUS()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	S	S
<code>WIFEXITED()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	C	S
<code>WIFSIGNALED()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	C	S
<code>WIFSTOPPED()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	C	S
<code>WSTOPSIG()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	C	S
<code>WTERMSIG()</code> (2)	C	<i>union wait</i> は、SunOS 4.x でサポートされていたが、SVR4 および SunOS 5.6 ではサポートされない。	N	N	C	S

表 B-4 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wait() (2V)	C	SunOS 4.x では、wait() は int 型の値を返す。SunOS 5.6、ABI、SVID、または SVR4 では、wait() は pid_t 型の値を返す。また、SunOS 5.6、ABI、SVID、SVR4 では <sys/types.h> に含まれるが、SunOS 4.x では含まれない。union wait は、以前の SunOS との下位互換性のために SunOS 4.x でサポートされていたが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。SunOS 4.x では、wait() は、SV_INTERRUPT ビットがそのシグナル用のフラグに設定されない限り、終了を待つ間にプロセスがシグナルを受け付けると自動的に再起動される。SunOS 5.6、ABI、SVID、または SVR4 では、wait() は、シグナルを受け付けられると途中で戻る。	C	C	C	N
wait() (2V) -SysV	C	union wait は、SunOS 4.x でサポートされていたが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。SunOS 4.x では、wait() (2V) は、SV_INTERRUPT ビットがそのシグナル用のフラグに設定されない限り、終了を待つ間にプロセスがシグナルを受け付けると自動的に再起動される。SunOS 5.6、ABI、SVID、または SVR4 では、wait() (2) は、シグナルを受け付けられると途中で戻る。	C	C	C	N
wait3() (2V)	A	現在は wait3() (3C)。wait() (2) と waitpid() (2) が同様の機能を提供する。	A	A	A	S
wait4() (2V)	A	現在は wait4() (3C)。wait() (2) と waitpid() (2) が同様の機能を提供する。	A	A	A	S

表 B-4 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
waitpid() (2V)	C	SunOS 4.x では、waitpid() は int 型の値を返す。SunOS 5.6、ABI、SVID、または SVR4 では、waitpid() は pid_t 型の値を返す。waitpid() の pid 引数は、SunOS 4.x では int 型で、SunOS 5.6、ABI、SVID、または SVR4 では pid_t 型となっている。また、SunOS 5.6、ABI、SVID、SVR4 では <sys/types.h> に含まれるが、SunOS 4.x では含まれない。union wait は、以前の SunOS との下位互換性のために SunOS 4.x でサポートされていたが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。SunOS 4.x では、waitpid() は、SV_INTERRUPT ビットがそのシグナル用のフラグに設定されない限り、終了を待つ間にプロセスがシグナルを受け付けると自動的に再起動される。SunOS 5.6、ABI、SVID、または SVR4 では、waitpid() は、シグナルが受け付けられると途中で戻る。	C	C	C	N
waitpid() (2V) -SysV	C	union wait は、SunOS 4.x でサポートされていたが、SunOS 5.6、ABI、SVID、または SVR4 ではサポートされない。SunOS 4.x では、waitpid() (2V) は、SV_INTERRUPT ビットがそのシグナル用のフラグに設定されない限り、終了を待つ間にプロセスがシグナルを受け付けると自動的に再起動される。SunOS リリース 5.6、ABI、SVID、または SVR4 では、waitpid() (2) は、シグナルが受け付けられると途中で戻る。	C	C	C	N

表 B-4 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
write() (2V)	C#	<p>SunOS 4.x ソフトウェアでは、記述子の参照が非ブロック I/O の場合、<code>ioctl()</code> の <code>FIONBIO</code> 要求を使用するか、<code>FNDELAY</code> や <code>O_NDELAY</code> フラグのセットのため <code>fcntl()</code> を使用することにより、<code>write()</code> は -1 を返して <code>errno</code> に <code>EWOULDBLOCK</code> を設定する。</p> <p>SunOS 5.6 ソフトウェアでは、通常ファイルへの <code>write()</code> では、<code>O_NDELAY</code> または <code>O_NONBLOCK</code> が設定されていれば、<code>write()</code> は -1 を返して <code>errno</code> に <code>EAGAIN</code> を設定する。</p> <p><code>O_NONBLOCK</code> または <code>O_NDELAY</code> が設定されているパイプまたは FIFO への <code>write()</code> 要求では、<code>write()</code> はプロセスをブロックしない。プロセスをブロックしないで書き込むことができるデータがある場合、<code>write()</code> は書き込みできるものを書き込み、書き込まれたバイト数を返す。それ以外の場合は、<code>O_NONBLOCK</code> が設定されると -1 を戻して <code>errno</code> に <code>EAGAIN</code> を設定し、<code>O_NDELAY</code> が設定されると 0 を返す。<code>O_NDELAY</code> が設定されると、<code>{PIPE_BUF}</code> またはそれ以下のバイト数の <code>write()</code> 要求は、完全に正常終了して <code>nbytes</code> を返すか、または 0 を返す。</p> <p><code>{PIPE_BUF}</code> バイトより大きな <code>write()</code> 要求は、転送できるものを転送して書き込まれたバイト数を返すか、またはデータを転送しないで 0 を返す。また、要求が <code>{PIPE_BUF}</code> バイトよりも大きく、以前にパイプに書き込まれたデータがすべて読み出されている場合、<code>write()</code> は少なくとも <code>{PIPE_BUF}</code> バイトを転送する。SunOS 5.6 の <code>write()</code> ルーチンは、4.2 BSD スタイルの非ブロック I/O をサポートしない。</p> <p><code>errno</code> フラグの <code>EWOULDBLOCK</code> は、SunOS 4.x の <code>write()</code> には有効だが、SunOS 5.6 では無効。</p>	C#	C#	C#	N

表 B-4 続く

SunOS 4.x システム コール	SunOS 5.6	注	ABI	SVID	SVR4	BSD
write() (2V) -SysV	C#	SunOS 5.6、ABI、SVID、または SVR4 の write() は、4.2 BSD スタイルの非ブロック I/O をサポートしない。errno フラグの EWOULDBLOCK は、SunOS 4.x の write() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
writev() (2V)	C#	SunOS 5.6、ABI、SVID、または SVR4 の writev() は、4.2 BSD スタイルの非ブロック I/O をサポートしない。errno フラグの EWOULDBLOCK は、SunOS 4.x の writev() には有効だが、SunOS 5.6、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N

ライブラリルーチンリファレンス

この付録はライブラリルーチンの情報を表形式で収録しています。すべての SunOS 4.x ライブラリルーチンを示し、Solaris 7、ABI、SVID、SVR4、および SunOS/BSD ソース互換パッケージにおける使用方法の違いを示します。

リファレンスの使い方

- インタフェースに C (変更) とマークされている場合、SunOS 4.x コマンドと Solaris 7 コマンドとの違いについて簡単な説明があります。
- インタフェースに S (同じ) とマークされている場合、Solaris 7 インタフェースは SunOS 4.x インタフェースの機能をすべてサポートします。場合によってはインタフェースは拡張されていますが、これは SunOS 4.x インタフェースの完全なスーパーセットと考えることができます。
- インタフェースに A (置換) とマークされている場合、その置換については注を読んでください。
- インタフェースに N (利用不可) とマークされている場合、その置換については注を読んでください。利用できる別のコマンドがある場合、SunOS 5.6 の欄に示されます。

SunOS 4.x には、System V の多くのルーチンの互換バージョンを提供する System V のインストールオプションがあります。System V インタフェースはこのあとの表に記載されています。SunOS 4.x インタフェースの System V バージョンを指す場合には、「SysV」とインタフェースの横に示してあります。

/usr/lib および /usr/5lib の両方に存在するルーチンにはエントリがこのリファレンス表に 2 つあります。はじめのエントリは /usr/lib ルーチンについて説明し、第 2 のエントリは /usr/5lib ルーチンについて説明します。

すべての Solaris 7 インタフェースに関する詳細な情報については、『SunOS Reference Manual』のセクション 3 「Library Routines」を参照してください。

例

以下の表に、エントリのサンプルをその説明とともにいくつか示します。

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	ABI	SVID	SVR4	BSD
clntraw _create() (3N)	S	利用可能。ただし SunOS 5.6 および SVR4 ではなるべく clnt_raw_create() (3N) を使用する。	A	A	S	N

このリリースには、clntraw_create() ルーチン、および置換ルーチンである clnt_raw_create() があります。clntraw_create() を使用するアプリケーションは、このリリースおよび他の SVR4 準拠のシステムで引き続き利用できますが、できるだけこれらのアプリケーションを更新し clnt_raw_create() を使用してください。clntraw_create() はもう使用されないとみなされ、将来利用できなくなる可能性があります。アプリケーションを ABI または SVID 準拠としたい場合は clnt_raw_create() を使用してください。

SunOS 4.x コマンド	SunOS 5.6	利用可能な代替コマンドと注	ABI	SVID	SVR4	BSD
putpwent() (3)	S		S	S	S	N

SunOS 4.x と SunOS 5.6 において、putpwent() ルーチンは同じです。このルーチンを使用するアプリケーションは SunOS 4.x リリースと同様に動作します。

ライブラリルーチンのリファレンス (A, B)

表 C-1

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
a64l () (3)	S		S	S	S	N
abort () (3)	S		S	S	S	N
abs () (3)	S		S	S	S	N
acos () (3M)	C	SunOS 4.x では、acos () 引数の絶対値が 1 より大きい場合、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID、または SVR4 では EDOM エラーおよび DOMAIN 数値演算エラーの結果として 0 (ゼロ) を返す。	N	C	C	N
acosh () (3M)	S		N	S	S	N
addch () (3V) -SysV	S		N	S	S	N
addexportent () (3)	A	/etc/dfs/sharetab ファイルは、/etc/exports を置換。詳細についてはマニュアルページの share (1M)、unshare (1M)、sharetab (4) を参照。	N	N	N	N
addmntent () (3)	A	putmntent () ルーチンが同様の機能を提供。getmntent () (3C) を参照。	N	N	N	N
addstr () (3V) -SysV	S		N	S	S	N
agt_create () (3L)	N		N	N	N	N
agt_enumerate () (3L)	N		N	N	N	N

表 C-1 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
agt_trap() (3L)	N		N	N	N	N
aint() (3M)	N		N	N	N	N
aiocancel() (3)	S		N	N	N	N
aioread() (3)	S		N	N	N	N
aiowait() (3)	S		N	N	N	N
aiowrite() (3)	S		N	N	N	N
alarm() (3V)	S		S	S	S	N
alloca() (3)	S		N	N	N	N
alphasort() (3)	N		N	N	N	S
anint() (3M)	N		N	N	N	N
annuity() (3M)	N		N	N	N	N
arc() (3X)	S		N	N	N	N
asctime() (3V)	C	ctime() (3V) を参照。	C	C	C	N
asin() (3M)	C	SunOS 4.x では、asin() の引数の絶対値が 1 より大きい場合、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID、または SVR4 では EDOM エラーおよび DOMAIN 数値演算エラーの結果として 0 (ゼロ) を返す。	N	C	C	N
asinh() (3M)	S		N	S	S	N

表 C-1 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
assert() (3V)	C	SunOS 4.x の assert() では exit() (3C)、SunOS 5.6、あるいは ABI、SVID または SVR4 では abort() (3C) を呼び出す。	C	C	C	N
assert() (3V) -SysV	S		S	S	S	N
atan() (3M)	S		N	S	S	N
atan2() (3M)	C	atan2(0.0,0.0)() と指定すると、SunOS 5.6、SVID、または SVR4 では 0 (ゼロ) を返し errno に EDOM を設定。SunOS 4.x では、ANSI/IEEE Std754-1985 に基づく 4.3BSD に準拠し +/-0.0 または +/-PI を返す。	N	C	C	N
atanh() (3M)	S		N	S	S	N
atof() (3)	C	strtod() (3) を参照。	C	C	C	N
atoi() (3)	S		S	S	S	N
atol() (3)	S		S	S	S	N
attroff() (3V) -SysV	S		N	S	S	N
attron() (3V) -SysV	S		N	S	S	N
attrset() (3V) -SysV	S		N	S	S	N
audit_args() (3)	N		N	N	N	N
audit_text() (3)	N		N	N	N	N
authdes_create() (3N)	A	利用可能。ただし SunOS 5.6、ABI、SVID、SVR4 ではなるべく authdes_seccreate() (3N) を使用する。	A	A	A	N

表 C-1 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
authdes_getucred() (3N)	S		S	S	S	N
auth_destroy() (3N)	S		S	S	S	N
authnone_create() (3N)	S		S	S	S	N
authunix_create() (3N)	A	利用可能。ただし、なるべく authsys_seccreate() (3N) を使用する。	A	A	A	N
authunix_create_default() (3N)	A	利用可能。ただし、なるべく authsys_create_default() (3N) を使用する。	A	A	A	N
baudrate() (3V) -SysV	S		N	S	S	N
bcmp() (3)	S	現在は bcmp() (3C)	A	A	A	S
bcopy() (3)	S	現在は bcopy() (3C)	A	A	A	S
beep(3V) -SysV()	S		N	S	S	N
bindresvport() (3N)	S		N	N	S	N
bootparam() (3R)	S		N	N	N	N

表 C-1 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
box() (3V)	C	<p>SunOS 4.x の box() は関数。</p> <p>SunOS 5.6 の box()、または SVID と SVR4 の box(win, verch, horch)() は wborder(win, verch, verch, horch, horch, 0, 0, 0, 0)() を呼び出すマクロ。</p> <p>SunOS 5.6、SVID、または SVR4 の環境で < curses.h > で定義されている下記のデフォルト値 ACS_ULCORNER、ACS_URCORNER、ACS_BLCORNER、ACS_BRCORNER を使用し、ウィンドウを囲む左上、右上、左下、右下の四隅を指定したボックスを描画。また SunOS 4.x ソフトウェアの引数 verch および horch は char 型だが、SunOS 5.6、SVID、または SVR4 では ch 型。</p>	N	C	C	S
box() (3V) -SysV	S		N	S	S	N
bsearch() (3)	S		S	S	S	N
byteorder() (3N)	S		N	N	S	N
bzero() (3)	S	現在は bzero() (3C)。	A	A	A	S

ライブラリルーチンのリファレンス (C, D)

表 C-2

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>_crypt()</code> (3)	<code>crypt()</code>	<code>crypt()</code> (3C) ルーチンが同様の機能を提供。	N	A	A	N
<code>CHECK()</code> (3L)	N		N	N	N	N
<code>calloc()</code> (3)	S		S	S	S	N
<code>callrpc()</code> (3N)	A	利用可能。ただし、なるべく <code>rpc_call()</code> (3N) を使用する。	N	N	S	N
<code>catclose()</code> (3C)	S		S	S	S	N
<code>catgetmsg()</code> (3C)	A	SunOS 5.6、ABI、SVID、または SVR4 では、 <code>catgets()</code> (3C) と <code>strncpy()</code> (3N) を使用し、カタログメッセージを内部バッファエリアからプログラムバッファにコピー。	A	A	A	N
<code>catgets()</code> (3C)	S		S	S	S	N
<code>catopen()</code> (3C)	S		S	S	S	N
<code>cbc_crypt()</code> (3)	S		N	N	N	N
<code>cbreak()</code> (3V) -SysV	S		N	S	S	S
<code>cbrt()</code> (3M)	S		N	S	S	N
<code>ceil()</code> (3M)	S		N	S	S	N
<code>cfgetispeed()</code> (3V)	S		S	S	S	N
<code>cfgetospeed()</code> (3V)	S		S	S	S	N
<code>cfree()</code> (3)	A	<code>void free(void*ptr)()</code> (<code>malloc()</code> (3C) を参照) に置換。	A	A	A	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
cfsetispeed() (3V)	S		S	S	S	N
cfsetospeed() (3V)	S		S	S	S	N
circle() (3X)	S		N	N	N	N
clear() (3V) -SysV	S		N	S	S	S
clearerr() (3V) -SysV	S		S	S	S	N
clearok() (3V) -SysV	S		N	S	S	S
clnt_broadcast() (3N)	S	利用可能。ただし、なるべく rpc_broadcast() (3N) を使 用する。	A	A	A	N
clnt_call() (3N)	S		S	S	S	N
clnt_control() (3N)	S		S	S	S	N
clnt_create() (3N)	S		S	S	S	N
clnt_destroy() (3N)	S		S	S	S	N
clnt_freeres() (3N)	S		S	S	S	N
clnt_geterr() (3N)	S		S	S	S	N
clnt_pcreateerror() (3N)	S		S	S	S	N
clnt_perrno() (3N)	S		S	S	S	N
clnt_perror() (3N)	S		S	S	S	N
clnt_screateerror() (3N)	S		S	S	S	N
clnt_serrno() (3N)	S		S	S	S	N
clnt_serror() (3N)	S		S	S	S	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
clntraw_create() (3N)	S	利用可能。ただし SunOS 5.6 および SVR4 ではなるべく clnt_raw_create() (3N) を使用する。	A	A	S	N
clnttcp_create() (3N)	S	利用可能。ただし SunOS 5.6、ABI、SVID、または SVR4 ではなるべく clnt_create() (3N)、clnt_tli_create() (3N)、および clnt_vc_create() (3N) の各ルーチンを使用する。	N	N	S	S
clntudp_bufcreate() (3N)	S	利用可能。 ただし SunOS 5.6、ABI、SVID、または SVR4 ではなるべく clnt_create() (3N)、clnt_tli_create() (3N)、および clnt_dg_create() (3N) の各ルーチンを使用する。	N	N	S	
clntudp_create() (3N)	S	利用可能。ただし、なるべく clnt_create() (3N)、clnt_tli_create() (3N)、および clnt_dg_create() (3N) の各ルーチンを使用する。	N	N	S	S
clock() (3C)	S		S	S	S	N
closedir() (3V)	S		S	S	S	N
closedir() (3V) -SysV	S		S	S	S	N
closelog() (3)	S		N	N	S	N
closepl() (3X)	S		N	N	N	N
clrtobot() (3V) -SysV	S		N	S	S	S
clrtoeol() (3V) -SysV	S		N	S	S	S

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>compound()</code> (3M)	N		N	N	N	N
<code>cont()</code> (3X)	S		N	N	N	N
<code>copysign()</code> (3M)	N		N	N	S	N
<code>copywin()</code> (3V) -SysV	S		N	S	S	N
<code>cos()</code> (3M)	C	<p>SunOS 5.6、あるいは SVID または SVR4 では、引数が 0 (ゼロ) よりかなり小さい場合、有効桁が失われるため 0 (ゼロ) を返す。この場合、標準出力に <code>TLOSS (matherr())</code> (3M) を参照) を示すメッセージが出力される。有効桁の一部が失われると <code>PLOSS</code> エラーが生成されるが、出力は行われない。いずれの場合も <code>errno</code> に <code>ERANGE</code> が設定される。</p> <p>SunOS 4.x では、引数の絶対値が $\pi/4$ を超えると、ソフトウェアまたはハードウェアにおいて引数が減少する。</p> <p><code><math.h></code> で定義されている <code>fp_pi</code> 変数により、精度は実行時に変更可能。例外エラーはどちらのバージョンでも IEEE 754 に基づき生成される。</p>	N	C	C	N
<code>cosh()</code> (3M)	S		N	S	S	N
<code>crmode()</code> (3X)	A	<code>cbreak()</code> (<code>curs_inopts()</code> (3X) を参照) に置換。	A	A	A	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>crypt()</code> (3)	C	SunOS 4.x では、引数 <i>salt</i> の先頭の 2 文字が、追加された認証ルーチン (<code>pwdauth()</code> (3) および <code>grpauth()</code> (3)) を呼び出すための特殊な文字 (<code>##</code> および <code>#\$</code>) かどうかを解析。それらの関数 (<code>pwdauth()</code> (3) および <code>grpauth()</code> (3)) が TRUE を返せば <code>crypt()</code> は <i>salt</i> を返し、そうでなければ NULL を返す。SunOS 5.6、SVID、または SVR4 ではサポートされていない。	N	C	C	N
<code>ctermid()</code> (3V) -SysV	S		S	S	S	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>ctime()</code> (3V)	C	<p>SunOS 4.x の <code>tm</code> 構造体には、SunOS 5.6、ABI、SVID、または SVR4 で存在しない 2 つのフィールド <code>tm_zone</code> および <code>tm_gmtoff</code> が含まれる。SunOS 5.6、ABI、SVID、または SVR4 では、GMT と地方標準時間との差 (単位は秒) が外部変数 <code>timezone</code>、夏時間を適用するかどうかが外部変数 <code>daylight</code> で示される。さらに SunOS 5.6、ABI、SVID、または SVR4 では、標準時間および夏時間の時間帯名が外部変数 <code>tzname</code> に格納される。これらの外部変数 (<code>timezone</code>、<code>daylight</code>、および <code>tzname</code>) は、SunOS 4.x System V の <code>ctime()</code> (3V) ライブラリルーチンによりサポートされる。</p> <p>環境変数 <code>TZ</code> の使用方法は SunOS 4.x と SunOS 5.6、ABI、SVID、または SVR4 とでは異なる。SunOS 4.x では、<code>TZ</code> は時間の変換情報が読み出される <code>tzfile-format</code> ファイルのパス名。SunOS 5.6、ABI、SVID、または SVR4 では、<code>TZ</code> 自体が (<code>tzfile-format</code> とは違うフォーマットの) 時間変換情報を含む。</p>	C	C	C	N
<code>curs_set()</code> (3V) -SysV	S		N	S	S	N
<code>cuserid()</code> (3V)	S		S	S	S	N
<code>cv_broadcast()</code> (3L)	N		N	N	N	N
<code>cv_create()</code> (3L)	N		N	N	N	N
<code>cv_destroy()</code> (3L)	N		N	N	N	N
<code>cv_enumerate()</code> (3L)	N		N	N	N	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
cv_notify() (3L)	N		N	N	N	N
cv_send() (3L)	N		N	N	N	N
cv_wait() (3L)	N		N	N	N	N
cv_waiters() (3L)	N		N	N	N	N
dbm_clearerr() (3)	S		N	N	N	N
dbm_close() (3X)	S	dbm_close() (3) ルーチンが同様の機能を提供。	N	N	N	N
dbm_delete() (3)	S	dbm_delete() (3) ルーチンが同様の機能を提供。	N	N	N	N
dbm_error() (3)	S		N	N	N	N
dbm_fetch() (3)	S		N	N	N	N
dbm_firstkey() (3)	S		N	N	N	N
dbm_nextkey() (3)	S		N	N	N	N
dbm_open() (3)	S		N	N	N	N
dbm_store() (3)	S		N	N	N	N
dbmclose() (3X)	N		N	N	N	S
dbminit() (3X)	S		N	N	N	S
decimal_to_double() (3)	S		N	N	N	N
decimal_to_extended() (3)	S		N	N	N	N
decimal_to_floating() (3)	S		N	N	N	N
decimal_to_single() (3)	S		N	N	N	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
def_prog_mode() (3V) -SysV	S		N	S	S	N
def_shell_mode() (3V) -SysV	S		N	S	S	N
del_curterm() (3V) -SysV	S		N	S	S	N
delay_output() (3V) -SysV	S		N	S	S	N
delch() (3V) -SysV	S		N	S	S	S
delete() (3X)	A		N	N	N	S
deleteln() (3V) -SysV	S		N	S	S	S
des_crypt() (3)	N		N	N	N	N
des_setparity() (3)	S		N	N	N	N
delwin() (3V) -SysV	S		N	S	S	S
dlclose() (3X)	S		N	N	S	N
dLError() (3X)	S		N	N	S	N
dlopen() (3X)	S		N	N	S	N
dlsym() (3X)	S		N	N	S	N
dn_comp() (3)	S		N	N	S	N
dn_expand() (3)	S		N	N	S	N
double_to_decimal() (3)	S		N	N	N	N
doupdate() (3V) -SysV	S		N	S	S	N
draino() (3V) -SysV	S		N	N	N	N

表 C-2 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
drand48() (3)	S		N	S	S	N
dysize() (3V)	N		N	N	N	N

ライブラリルーチンのリファレンス (E, F)

表 C-3

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ecb_crypt() (3)	S		N	N	N	N
echo() (3V) -SysV	S		N	S	S	S
echochar() (3V) -SysV	S		N	S	S	N
econvert() (3)	S		N	N	N	N
ecvt() (3)	S		N	N	S	N
edata() (3)	S		N	N	S	N
encrypt() (3)	S		N	S	S	N
end() (3)	S		N	N	S	N
endac() (3)	N		N	N	N	N
endexportent() (3)	A	/etc/dfs/sharetab ファイルが /etc/exports を置換。詳細については share(1M)、unshare(1M)、sharetab(4) を参照。	A	A	A	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
endfsent() (3)	A	fclose() (3) に置換。	A	A	A	N
endgraent() (3)	N		N	N	N	N
endgrent() (3V)	S		S	S	S	N
endhostent() (3N)	S		N	N	S	N
endmntent() (3)	A	fclose() (3) に置換。	A	A	A	N
endnetent() (3N)	S		N	N	S	N
endnetgrent() (3N)	S		N	N	N	N
endprotoent() (3N)	S		N	N	S	N
endpwaent() (3)	N		N	N	N	N
endpwent() (3V)	S		S	S	S	N
endrpcent() (3N)	S		N	N	S	N
endservent() (3N)	S		N	N	S	N
endttyent() (3)	N	SunOS 5.6 の tty システムに関する情報は ttymon(1) および ttydefs(4) を参照。	N	N	N	N
endusershell() (3)	S		N	N	N	N
endwin() (3V)	C	SunOS 4.x での endwin() 戻り値は不定。SunOS 5.6、SVID または SVR4 では、正常終了なら OK、そうでない場合は ERR を返す。	N	C	C	S
endwin() (3V) -SysV	S		N	S	S	N
erand48() (3)	S		N	S	S	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
erase() (3V) -SysV	S		N	S	S	S
erasechar() (3V) -SysV	S		N	S	S	N
erf() (3M)	S		N	S	S	N
erfc() (3M)	S		N	S	S	N
errno() (3)	S		N	N	N	N
etext() (3)	S		N	N	S	N
ether() (3R)	N		N	N	N	N
ether_aton() (3N)	S		N	N	S	N
ether_hostton() (3N)	S		N	N	S	N
ether_line() (3N)	S		N	N	S	N
ether_ntoa() (3N)	S		N	N	S	N
ether_ntohost() (3N)	S		N	N	S	N
exc_bound() (3L)	N		N	N	N	N
exc_handle() (3L)	N		N	N	N	N
exc_notify() (3L)	N		N	N	N	N
exc_on_exit() (3L)	N		N	N	N	N
exc_raise() (3L)	N		N	N	N	N
exc_unhandle() (3L)	N		N	N	N	N
exc_uniqpatt() (3L)	N		N	N	N	N
execl() (3V)	C		C	C	C	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
execl(3V) () (3V) -SysV	S		S	S	S	N
execle() (3V)	C		C	C	C	N
execle() (3V) -SysV	S		S	S	S	N
execlp() (3V)	C		C	C	C	N
execlp() (3V) -SysV	S		S	S	S	N
execv() (3V)	C		C	C	C	N
execv() (3V) -SysV	S		S	S	S	N
execvp() (3V)	C		C	C	C	N
execvp() (3V) -SysV	S		S	S	S	N
exit() (3)	C	SunOS 4.x および SunOS 5.6、ABI、SVID、または SVR4 のどちらでも、プロセスが終了する前に追加処理を行う。SunOS 4.x では on_exit() (3) ルーチンによって登録されたすべての関数を呼び出すが、SunOS 5.6、ABI、SVID、または SVR4 では atexit() ルーチンによって登録されたすべての関数を呼び出す。on_exit() (3) ルーチンを使用して追加された関数がない場合は、SunOS 4.x の exit() (3) と SunOS 5.6、ABI、SVID、または SVR4 の exit() とは互換性がある。	C	C	C	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>exp()</code> (3M)	C	SunOS 5.6、SVID または SVR4 では、オーバーフローの場合は HUGE、アンダフローの場合は 0 を返す。SunOS 4.x では、戻り値は IEEE のオーバーフローおよびアンダフロー (インプリメンテーションで定義)。SunOS 4.x では HUGE が + (プラス) の無限大として定義されるため、 <code>exp()</code> (HUGE) および <code>exp()</code> (-HUGE) と指定した場合でもオーバーフローやアンダフローは起こらず、 <code>errno</code> は生成されない。SunOS 5.6、SVID または SVR4 では <code>errno</code> に ERANGE が設定される。	N	C	C	N
<code>exp10()</code> (3M)	N		N	N	N	N
<code>exp2()</code> (3M)	N		N	N	N	N
<code>expm1()</code> (3)	N		N	N	N	N
<code>exportent()</code> (3)	A	<code>/etc/dfs/sharetab</code> ファイルが <code>/etc/exports</code> を置換。詳細については <code>share(1M)</code> 、 <code>unshare(1M)</code> 、 <code>sharetab(4)</code> を参照。	A	A	A	N
<code>extended_to_decimal()</code> (3)	S		N	N	N	N
<code>fabs()</code> (3M)	S		N	S	S	N
<code>fclose()</code> (3S)	S		S	S	S	N
<code>fconvert()</code> (3)	S		N	N	N	N
<code>fcvt()</code> (3)	S		N	N	S	N
<code>fdopen()</code> (3V)	S		S	S	S	N
<code>feof()</code> (3V)	S		S	S	S	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ferror() (3V)	S		S	S	S	N
fetch() (3X)	A	SunOS 5.6 では dbm_fetch() (3) に置換。	N	N	N	S
fflush() (3S)	S		S	S	S	N
ffs() (3)	S		N	N	S	N
fgetc() (3V)	S		S	S	S	N
fgetgraent() (3)	N		N	N	N	N
fgetgrent() (3V)	S		N	S	S	N
fgetpwaent() (3)	N		N	N	N	N
fgetpwent() (3V)	S		N	S	S	N
fgets() (3S)	S		S	S	S	N
fileno() (3V)	S		S	S	S	N
file_to_decimal() (3)	N		N	N	N	N
filter() (3V) -SysV	S		N	S	S	N
finite() (3M)	N		N	N	N	N
firstkey() (3X)	A	SunOS 5.6 では dbm_firstkey() (3) に置換。	N	N	N	S
fixterm() (3V)	A	reset_prog_mode() (3X) ルーチンが同様の機能を提供。	N	A	A	N
flash() (3V) -SysV	S		N	S	S	N
floatingpoint() (3)	S		N	N	N	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
floor() (3M)	S		N	S	S	N
flushinp() (3V) -SysV	S		N	S	S	N
flusok() (3X)	N		N	N	N	S
fmod() (3M)	C	fmod(x,0.0)() と指定すると、SunOS 5.6、SVID、または SVR4 では x を返し errno に EDOM を設定する。SunOS 4.x では、ANSI/IEEE Std 754-1985 に基づき、また 4.3 BSD に準拠し NaN を返す。	N	C	C	N
fopen() (3V)	S		S	S	S	S
fp_class() (3M)	N		N	N	N	N
fprintf() (3V)	S		S	S	S	S
fputc() (3S)	S		S	S	S	N
fputs() (3S)	S		S	S	S	N
fread() (3S)	S		S	S	S	N
free() (3)	S		S	S	S	N
freopen() (3V)	S		S	S	S	S
frexp() (3M)	S		N	S	S	N
fscanf() (3V)	S		S	S	S	N
fseek() (3S)	S		S	S	S	N
ftell() (3S)	S		S	S	S	N
ftime() (3V)	S	現在は ftime() (3C)。	A	A	A	S
ftok() (3)	S		S	S	S	N

表 C-3 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ftw() (3)	S		S	S	S	N
func_to_decimal() (3)	N		N	N	N	N
fwrite() (3S)	S		S	S	S	N

ライブラリルーチンのリファレンス (G, H)

表 C-4

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
gamma() (3M)	S		N	S	S	N
garbagedlines() (3V) -SysV	S		N	N	N	N
gcd() (3X)	S		N	N	N	N
gconvert() (3)	S		N	N	N	N
gcvt() (3)	S		N	N	S	N
getacdir() (3)	N		N	N	N	N
getacflg() (3)	N		N	N	N	N
getacinfo() (3)	N		N	N	N	N
getacmin() (3)	N		N	N	N	N
getauditflagsbin() (3)	N		N	N	N	N
getauditflagschar() (3)	N		N	N	N	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getbegyx() (3V) -SysV	S		N	S	S	N
getc() (3V)	S		S	S	S	N
getcap() (3X)	N		N	N	N	S
getch() (3V)	C	SunOS 5.6、SVID または SVR4 ソフトウェアにおいて、ウィンドウがパッドでなく wrefresh() を最後に呼び出した後で移動または修正された場合、別の文字が読み込まれる前に wrefresh() が呼び出される。SunOS 4.x では、このような状況で wrefresh() は呼び出されない。	N	C	C	S
getch() (3V) -SysV	C		N	C	C	S
getchar() (3V)	S		S	S	S	N
getcwd() (3V)	S	SVR4 および SunOS 5.6 では SunOS 4.x と互換性がある。SunOS 4.x では、buf が NULL ポインタのとき malloc() (3) を使用して size バイトの空きスペースを得る。この機能は ABI および SVID ではサポートされていない。	C	C	S	N
getenv() (3V)	S		S	S	S	N
getexportent() (3)	A	/etc/dfs/sharetab ファイルが /etc/exports を置換。詳細については share(1M)、unshare(1M)、sharetab(4) を参照。	A	A	A	N
getexportopt() (3)	A	/etc/dfs/sharetab ファイルが /etc/exports を置換。詳細については share(1M)、unshare(1M)、sharetab(4) を参照。	A	A	A	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getfauditflags() (3)	N		N	N	N	N
getfsent() (3)	A	getvfsent() (3) に置換。	N	N	N	N
getfsfile() (3)	A	getvfsfile() (3) に置換。	N	N	N	N
getfsspec() (3)	A	getvfsfile() (3) に置換。	N	N	N	N
getfstype() (3)	A	getvfsany() (3) に置換。	N	N	N	N
getgraent() (3)	N		N	N	N	N
getgranam() (3)	N		N	N	N	N
getgrent() (3V)	S		S	S	S	N
getgrgid() (3V)	S		S	S	S	N
getgrnam() (3V)	S		S	S	S	N
gethostbyaddr() (3N)	S		N	N	S	N
gethostbyname() (3N)	S		N	N	S	N
gethostent() (3N)	S		N	N	S	N
getlogin() (3V)	S		S	S	S	N
getmaxyx() (3V) -SysV	S		N	S	S	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getmntent() (3)	C	SunOS 4.x と SunOS 5.6、SVID または SVR4 の互換性はない。SunOS 4.x では mntent 型のオブジェクトへのポインタを返すが、SunOS 5.6、SVID または SVR4 の getmntent() は int 型を返す。また SunOS 5.6、SVID または SVR4 の getmntent() は、互換性がない異なる構造体の型 (mnttab) を使用してファイルエントリの型を返す。さらに、/etc/vfstab の対応する「-」エントリに対し NULL ポインタを返す。	N	C	C	N
get_myaddress() (3N)	S	利用可能。ただし、なるべく netdir_getbyname() (3N) を使用する。	S	N	S	N
getnetbyaddr() (3N)	S		N	N	S	N
getnetbyname() (3N)	S		N	N	S	N
getnetent() (3N)	S		N	N	S	N
getnetgrent() (3N)	N		N	N	N	N
getnetname() (3N)	S		S	S	S	N
getopt() (3)	S		S	S	S	N
getpass() (3V)	S		S	S	S	N
getprotobyname() (3N)	S		N	N	S	N
getprotobynumber() (3N)	S		N	N	S	N
getprotoent() (3N)	S		N	N	S	N
getpublickey() (3R)	S		S	S	S	N
getpw() (3)	S		N	N	S	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
getpwaent() (3)	N		N	N	N	N
getpwanam() (3)	N		N	N	N	N
getpwent() (3V)	S		S	S	S	N
getpwnam() (3V)	S		S	S	S	N
getpwuid() (3V)	S		S	S	S	N
getrpcbyname() (3N)	S		S	S	S	N
getrpcbynumber() (3N)	S		S	S	S	N
getrpcport() (3N)	S		S	S	S	N
getrpcport() (3R)	A	pmap_getport() により同じ機能を提供。	N	N	N	N
gets() (3S)	S		S	S	S	N
getsecretkey() (3R)	S		S	S	S	N
getservbyname() (3N)	S		N	N	S	N
getservbyport() (3N)	S		N	N	S	N
getservent() (3N)	S		N	N	S	N
getstr() (3V) -SysV	C		N	C	C	S
getsubopt() (3)	S		S	S	S	N
getsyx(3V) -SysV()	S		N	S	S	N
gettext() (3)	C	SunOS 5.4 では、gettext(3) は LC_MESSAGES ディレクトリの位置について最初に NLSPATH を検索する。	N	N	N	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
gettmode() (3V)	C	SunOS 5.6 では、ヘッダファイル <curses.h> がヘッダファイル <stdio.h> および <unctrl.h> を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	N	N	S
gettmode() (3V) -SysV	S		N	N	N	N
getttyent() (3)	A	SunOS 5.6 の tty システムに関する情報は、ttymon(1) および ttydefs(4) を参照。	N	N	N	N
getttynam() (3)	A	SunOS 5.6 の tty システムに関する情報は、ttymon(1) および ttydefs(4) を参照。	N	N	N	N
getusershell() (3)	S		N	N	N	N
getw() (3V)	S		S	S	S	N
getwd() (3)	S	現在は getwd() (3C)。	A	A	A	S
getyx() (3V) -SysV	S		N	S	S	S
gmtime() (3V)	C	ctime() (3V) を参照。	C	C	C	N
grpauth() (3)	N		N	N	N	N
gsignal() (3)	S		N	N	S	N
gtty() (3C)	A	termio(7) インタフェースが同様の機能を提供。	A	A	A	N

表 C-4 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
HUGE() (3M)	C	HUGE は SunOS 4.x では <code>infinity()</code> (3M) として <code><math.h></code> で定義され、IEEE の無限大を生成。SunOS 5.6、SVID、または SVR4 では <code><math.h></code> で定義されるマシン依存型定数。	N	C	C	N
HUGE_VAL() (3M)	C	SunOS 4.x では <code>infinity()</code> (3M) として <code><math.h></code> で定義され、IEEE の無限大を生成。SunOS 5.6、SVID、または SVR4 では <code><math.h></code> で定義されるマシン依存型定数。	N	C	C	N
halfdelay() (3V) -SysV	S		N	S	S	N
has_ic() (3V) -SysV	S		N	S	S	N
has_il() (3V) -SysV	S		N	S	S	N
hasmntopt() (3)	N		N	N	N	N
hcreate() (3)	S		S	S	S	N
hdestroy() (3)	S		S	S	S	N
host2netname() (3N)	S		S	S	S	N
hsearch() (3)	S		S	S	S	N
hypot() (3M)	S		N	S	S	N

ライブラリルーチンのリファレンス (I, J, K)

表 C-5

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
idlok() (3V)	C	SunOS 4.x では、機能が使われていないウィンドウに対し行の挿入または削除フラグを設定。SunOS 5.6、SVID または SVR4 では、行の挿入または削除フラグの機能が実際に使用されるかどうかを制御するフラグを設定。	N	C	C	S
idlok() (3V) -SysV	S		N	S	S	N
ieee_flags() (3M)	N		N	N	N	N
ieee_functions() (3M)	S		N	N	N	N
ieee_handler() (3M)	N		N	N	N	N
ieee_retrospective() (3M)	N		N	N	N	N
ilogb() (3M)	N		N	N	N	N
inch() (3V) -SysV	S		N	S	S	S
index() (3)	S	現在は index() (3C)。	A	A	A	S
inet_lnaof() (3N)	S		N	N	S	N
inet_makeaddr() (3N)	S		N	N	S	N
inet_netof() (3N)	S		N	N	S	N
inet_network() (3N)	S		N	N	S	N
inet_ntoa() (3N)	S		N	N	S	N
infinity() (3M)	N		N	N	N	N
initgroups() (3)	S		S	S	S	N

表 C-5 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>initscr()</code> (3V)	C	SunOS 4.x では関数。一方、SunOS 5.6、SVID または SVR4 では <code>initscr32()</code> を呼び出すマクロ。エラーが発生すると、SunOS 4.x では ERR を返すが、SunOS 5.6、SVID または SVR4 では適切なエラーメッセージを標準エラー出力へ書き込み終了する。	N	C	C	S
<code>initscr()</code> (3V) -SysV	S		N	S	S	N
<code>initstate()</code> (3)	S	現在は <code>initstate()</code> (3)。	N	A	A	S
<code>innetgr()</code> (3N)	S		N	N	N	N
<code>insch()</code> (3V) -SysV	S		N	S	S	S
<code>insertln()</code> (3V) -SysV	S		N	S	S	S
<code>insque()</code> (3)	S		N	N	S	N
<code>intrflush()</code> (3V) -SysV	S		N	S	S	N
<code>ipalloc()</code> (3R)	N		N	N	N	N
<code>irint()</code> (3M)	N	<code>rint()</code> に置換された。	N	N	N	N
<code>isalnum()</code> (3V)	S		S	S	S	N
<code>isalpha()</code> (3V)	S		S	S	S	N
<code>isascii()</code> (3V)	S		S	S	S	N
<code>isatty()</code> (3V)	S		S	S	S	N
<code>iscntrl()</code> (3V)	S		S	S	S	N
<code>isdigit()</code> (3V)	S		S	S	S	N

表 C-5 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
isendwin() (3V) -SysV	S		N	S	S	N
isgraph() (3V)	S		S	S	S	N
isinf() (3M)	N		N	N	N	N
islower() (3V)	S		S	S	S	N
isnan() (3M)	S		S	N	N	N
isnormal() (3M)	N		N	N	N	N
isprint() (3V)	S		S	S	S	N
ispunct() (3V)	S		S	S	S	N
issecure() (3)	N		N	N	N	N
isspace() (3V)	S		S	S	S	N
issubnormal() (3M)	N		N	N	N	N
isupper() (3V)	S		S	S	S	N
isxdigit() (3V)	S		S	S	S	N
iszero() (3M)	N		N	N	N	N
itom() (3X)	S		N	N	N	N
j0() (3M)	C	j0 (HUGE) ()、j1 (HUGE) ()、jn(4, HUGE) () と指定した場合、SunOS 4.x ではエラーを表示せずに 0(ゼロ) を返す。SunOS 5.6、SVID または SVR4 ソフトウェアでは、0(ゼロ) を返して errno に ERANGE を設定し、TLOSS 数値演算エラーを示すメッセージを標準エラー出力へ出力。	N	C	C	N

表 C-5 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
j1() (3M)	C		N	C	C	N
jn() (3M)	C		N	C	C	N
jrand48() (3)	S		N	S	S	N
key_decryptsession() (3N)	S		S	S	S	N
key_encryptsession() (3N)	S		S	S	S	N
key_gendes() (3N)	S		S	S	S	N
key_setsecret() (3N)	S		S	S	S	N
keyname() (3V) -SysV	S		N	S	S	N
keypad() (3V) -SysV	S		N	S	S	N
killchar() (3V) -SysV	S		N	S	S	N
klm_prot() (3R)	S		N	N	N	N
kvm_close() (3K)	S		N	N	N	N
kvm_getcmd() (3K)	S		N	N	N	N
kvm_getproc() (3K)	S		N	N	N	N
kvm_getu() (3K)	S		N	N	N	N
kvm_nextproc() (3K)	S		N	N	N	N
kvm_nlist() (3K)	S		N	N	N	N
kvm_open() (3K)	S		N	N	N	N
kvm_read() (3K)	S		N	N	N	N

表 C-5 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
kvm_setproc() (3K)	S		N	N	S	N
kvm_write() (3K)	S		N	N	N	N

ライブラリルーチンのリファレンス (L)

表 C-6

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
_longjmp() (3)	siglongjmp()	現在は_longjmp() (3C)。siglongjmp() (3) ルーチンが同様の機能を提供。	A	A	A	S
l3tol() (3C)	N		N	N	N	N
l64a() (3)	S		S	S	S	N
label() (3X)	S		N	N	N	N
lcong48() (3)	S		N	S	S	N
ldaclose() (3X)	N		N	N	N	N
ldahread() (3X)	N		N	N	N	N
ldaopen() (3X)	N		N	N	N	N
ldclose() (3X)	N		N	N	N	N

表 C-6 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ldexp() (3M)	C	SunOS 4.x と、SunOS 5.6、ABI、SVID、または SVR4 とはオーバーフロー処理のみ異なる。SunOS 4.x において正しい値がオーバーフローすると (+/-) 1.0e999 を返すのに対し、SunOS 5.6、ABI、SVID、または SVR4 では (+/-) HUGE (値の符号に従う) を返す。どちらも、errno には ERANGE が設定される。	C	C	C	S
ldfcn() (3)	N		N	N	N	N
ldfhread() (3X)	N		N	N	N	N
ldgetname() (3X)	N		N	N	N	N
ldlinit() (3X)	N		N	N	N	N
ldlitem() (3X)	N		N	N	N	N
ldlread() (3X)	N		N	N	N	N
ldlseek() (3X)	N		N	N	N	N
ldnlseek() (3X)	N		N	N	N	N
ldnrseek() (3X)	N		N	N	N	N
ldnshread() (3X)	N		N	N	N	N
ldnsseek() (3X)	N		N	N	N	N
ldohseek() (3X)	N		N	N	N	N
ldopen() (3X)	N		N	N	N	N
ldrseek() (3X)	N		N	N	N	N
ldshread() (3X)	N		N	N	N	N

表 C-6 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ldsseek() (3X)	N		N	N	N	N
ldtbindx() (3X)	N		N	N	N	N
ldtbread() (3X)	N		N	N	N	N
ldtbseek() (3X)	N		N	N	N	N
leaveok() (3V) -SysV	S		N	S	S	S
lfind() (3)	S		S	S	S	N
lgamma() (3M)	S		N	S	S	N
line() (3X)	S		N	N	N	N
linemod() (3X)	S		N	N	N	N
localdtconv() (3)	N		N	N	N	N
localeconv() (3)	S		S	S	S	N
localtime() (3V)	C	ctime() (3V) を参照。	C	C	C	N
lockf() (3)	S		S	S	S	N
log() (3M)	C	SunOS 4.x では、関数の結果が未定義の値になると (log(-1.0)() など)、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID または SVR4 では、EDOM エラーおよび DOMAIN 数値演算エラーの結果として -HUGE を返す。	N	C	C	N

表 C-6 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
log10() (3M)	C	SunOS 4.x では、関数の結果が未定義の値になると (log10(0)() など)、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID または SVR4 では、EDOM エラーおよび DOMAIN 数値演算エラーの結果として -HUGE を返す。	N	C	C	N
log1p() (3M)	N		N	N	N	N
log2() (3M)	N		N	N	N	N
logb() (3M)	S		N	C	C	N
longjmp() (3V)	S		S	S	S	S
longname() (3V)	C	SunOS 4.x では、 <i>termbuf</i> および <i>name</i> の 2 つの引数が必要。これらは SunOS 5.6、SVID または SVR4 では不要。 <i>termbuf</i> は、 <i>termcap</i> から端末エントリへのポインタで、SunOS 5.6、SVID または SVR4 では <i>terminfo</i> に置換。 <i>name</i> は、結果を保持するバッファへのポインタ。どちらも同じ情報を返すため、SunOS 5.6、SVID または SVR4 へ移植するには、SunOS 4.x の呼び出しから 2 つの引数を削除するだけでよい。	N	C	C	S
longname() (3V) -SysV	S		N	S	S	N
lrand48() (3)	S		N	S	S	N
lsearch() (3)	S		S	S	S	N
ltol3() (3C)	N		N	N	N	N
lwp_checkstkset() (3L)	N		N	N	N	N

表 C-6 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>lwp_create()</code> (3L)	N		N	N	N	N
<code>lwp_ctxinit()</code> (3L)	N		N	N	N	N
<code>lwp_ctxmemget()</code> (3L)	N		N	N	N	N
<code>lwp_ctxmemset()</code> (3L)	N		N	N	N	N
<code>lwp_ctxremove()</code> (3L)	N		N	N	N	N
<code>lwp_ctxset()</code> (3L)	N		N	N	N	N
<code>lwp_datastk()</code> (3L)	N		N	N	N	N
<code>lwp_destroy()</code> (3L)	N		N	N	N	N
<code>lwp_enumerate()</code> (3L)	N		N	N	N	N
<code>lwp_errstr()</code> (3L)	N		N	N	N	N
<code>lwp_fpset()</code> (3L)	N		N	N	N	N
<code>lwp_geterr()</code> (3L)	N		N	N	N	N
<code>lwp_getregs()</code> (3L)	N		N	N	N	N
<code>lwp_getstate()</code> (3L)	N		N	N	N	N
<code>lwp_join()</code> (3L)	N		N	N	N	N
<code>lwp_libcset()</code> (3L)	N		N	N	N	N
<code>lwp_newstk()</code> (3L)	N		N	N	N	N
<code>lwp_perror()</code> (3L)	N		N	N	N	N
<code>lwp_ping()</code> (3L)	N		N	N	N	N
<code>lwp_resched()</code> (3L)	N		N	N	N	N

表 C-6 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
lwp_resume() (3L)	N		N	N	N	N
lwp_self() (3L)	N		N	N	N	N
lwp_setpri() (3L)	N		N	N	N	N
lwp_setregs() (3L)	N		N	N	N	N
lwp_setstkcache() (3L)	N		N	N	N	N
lwp_sleep() (3L)	N		N	N	N	N
lwp_stkcswset() (3L)	N		N	N	N	N
lwp_suspend() (3L)	N		N	N	N	N
lwp_yield() (3L)	N		N	N	N	N

ライブラリルーチンのリファレンス (M, N)

表 C-7

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
MONITOR() (3L)	N		N	N	N	N
MSG_RECVALL() (3L)	N		N	N	N	N
madd() (3X)	S		N	N	N	N
madvise() (3)	S		N	N	N	N
malloc() (3)	S		S	S	S	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
malloc_debug() (3)	S		N	N	N	N
malloc_verify() (3)	S		N	N	N	N
mallocmap() (3)	S		N	N	N	N
matherr() (3M)	S		N	S	S	N
max_normal() (3M)	N		N	N	N	N
max_subnormal() (3M)	N		N	N	N	N
mblen() (3)	S		S	S	S	N
mbstowcs() (3)	S		S	S	S	N
mbtowc() (3)	S		S	S	S	N
mcmp() (3X)	S		N	N	N	N
mdiv() (3X)	S		N	N	N	N
memalign() (3)	S		N	N	S	N
memccpy() (3)	S		S	S	S	N
memchr() (3)	S		S	S	S	N
memcmp() (3)	S		S	S	S	N
memcpy() (3)	S		S	S	S	N
memset() (3)	S		S	S	S	N
meta() (3V) -SysV	S		N	S	S	N
mfree() (3X)	S		N	N	N	N
min() (3X)	S		N	N	N	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
min_normal() (3M)	N		N	N	N	N
min_subnormal() (3M)	N		N	N	N	N
mkstemp() (3)	S	mktemp() (3C) ルーチンが同様の機能を提供。	A	A	A	N
mktemp() (3)	C	SunOS 4.x mktemp() では、テンプレートの末尾の X 文字を 1 文字の英字と現在のプロセス ID に置換。SunOS 5.6、ABI、SVID、または SVR4 では、一意のファイル名を作成するために使用する末尾の文字列 (XXXXXX) を置換するのみ。アプリケーションがファイル名を特定しなければ (つまりアプリケーションがファイル名の一意性のみをチェックするのならば)、SunOS 4.x と、SunOS 5.6、ABI、SVID、または SVR4 のこの関数とは互換性がある。	C	C	C	N
mlock() (3)	S		S	S	S	N
mlockall() (3)	S		S	S	S	N
modf() (3M)	S		N	S	S	N
mon_break() (3L)	N		N	N	N	N
mon_cond_enter() (3L)	N		N	N	N	N
mon_create() (3L)	N		N	N	N	N
mon_destroy() (3L)	N		N	N	N	N
mon_enter() (3L)	N		N	N	N	N
mon_enumerate() (3L)	N		N	N	N	N
mon_exit() (3L)	N		N	N	N	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>mon_waiters()</code> (3L)	N		N	N	N	N
<code>moncontrol()</code> (3)	A	<code>profil()</code> (2) に置換。	A	A	A	N
<code>monitor()</code> (3)	C	<p>SunOS 4.x と SunOS 5.6、ABI、SVID、または SVR4 との相違点は以下のとおり。SunOS 4.x ソフトウェアではプログラム全体をプロファイルするため、次のように指定する。</p> <pre>extern etext () monitor(N_TXTOFF(0), etext, buf, bufsize, nfunc) ()</pre> <p>SunOS 5.6、ABI、SVID、または SVR4 では次のように指定する。</p> <pre>extern int etext (); monitor((int (*) ())2, etext, buf, bufsize, nfunc) ()</pre> <p>SunOS 4.x において、以前に定義済みの <code>buf</code> に対するモニタリングを中止し結果を書き込むには、次のように指定する。</p> <pre>monitor(0) ()</pre> <p>SunOS 5.6、ABI、SVID、または SVR4 <code>monitor</code> ルーチンでは次のように指定する。</p> <pre>monitor((int (*) ())0, (int (*) ())0, (WORD*) 0, 0, 0) ();</pre> <p>この後で <code>prof(1)</code> コマンドを使用すれば、結果が確認できる。</p>	C	C	C	N
<code>monstartup()</code> (3)	A	<code>profil()</code> (2) に置換。	A	A	A	N
<code>mout()</code> (3X)	S		N	N	N	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
move() (3V) -SysV	S		N	S	S	S
mrnd48() (3)	S		N	S	S	N
msg_enumrecv() (3L)	N		N	N	N	N
msg_enumsend() (3L)	N		N	N	N	N
msg_recv() (3L)	N		N	N	N	N
msg_reply() (3L)	N		N	N	N	N
msg_send() (3L)	N		N	N	N	N
msub() (3X)	S		N	N	N	N
msync() (3)	C	errno フラグの EIO は、SunOS 4.x では有効、SunOS 5.6、ABI、SVID、または SVR4 では無効。MS_INVALIDATE が指定されており、さらに 1 ページまたは複数のページがメモリにロックされている場合、errno フラグに設定される値は、SunOS 4.x では EPERM、SunOS 5.6、ABI、SVID、または SVR4 では EBUSY。	C	C	C	N
mtox() (3X)	S		N	N	N	N
mult() (3X)	S		N	N	N	N
munlock() (3)	S		S	S	S	N
munlockall() (3)	S		S	S	S	N
mvaddch() (3V) -SysV	S		N	S	S	N
mvaddstr() (3V) -SysV	S		N	S	S	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>mvcur() (3V)</code>	C	SunOS 4.x での戻り値は不定。 SunOS 5.6、SVID または SVR4 では、正常終了なら OK、そう でない場合は ERR を返す。	N	C	C	S
<code>mvcur() (3V) -SysV</code>	S		N	S	S	N
<code>mvdelch() (3V) -SysV</code>	S		N	S	S	N
<code>mvgetch() (3V) -SysV</code>	C	SunOS 5.6、SVID または SVR4 において、ウィンドウがパッド でなく <code>wrefresh()</code> を最後に 呼び出した後で移動または修正 された場合、別の文字が読み込 まれる前に <code>wrefresh()</code> が呼 び出される。SunOS 4.x では、 このような状況で <code>wrefresh()</code> は呼び出され ない。	N	C	C	N
<code>mvgetstr() (3V) -SysV</code>	C	<code>getstr() (3V) -SysV</code> を参 照。	N	C	C	N
<code>mvinch() (3V) -SysV</code>	S		N	S	S	N
<code>mvinsch() (3V) -SysV</code>	S		N	S	S	N
<code>mvprintw() (3V)</code>	C	<code>wprintw() (3V)</code> を参照。	N	C	C	S
<code>mvprintw() (3V) -SysV</code>	S		N	S	S	N
<code>mvscanw() (3V)</code>	C	<code>wscanw() (3V)</code> を参照。	N	C	C	S
<code>mvscanw() (3V) -SysV</code>	S		N	S	S	N
<code>mvwaddch() (3V) -SysV</code>	S		N	S	S	N
<code>mvwaddstr() (3V) -SysV</code>	S		N	S	S	N
<code>mvwdelch() (3V) -SysV</code>	S		N	S	S	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>mvwgetch()</code> (3V) -SysV	C	SunOS 5.6、SVID または SVR4 において、ウィンドウがパッドでなく <code>wrefresh()</code> を最後に呼び出した後で移動または修正された場合、別の文字が読み込まれる前に <code>wrefresh()</code> が呼び出される。SunOS 4.x では、このような状況で <code>wrefresh()</code> は呼び出されない。	N	C	C	N
<code>mvwgetstr()</code> (3V) -SysV	C	<code>getstr()</code> (3V) -SysV を参照。	N	C	C	N
<code>mvwin()</code> (3V)	C	SunOS 4.x を使用すれば、サブウィンドウは移動可能。SunOS 5.6、SVID または SVR4 では、親ウィンドウ内部のサブウィンドウ (または子ウィンドウ) を移動するには <code>mvderwin()</code> を使用すること。	N	C	C	S
<code>mvwin()</code> (3V) -SysV	S		N	S	S	N
<code>mvwinch()</code> (3V) -SysV	S		N	S	S	N
<code>mvwinsch()</code> (3V) -SysV	S		N	S	S	N
<code>mvwprintw()</code> (3V)	C		N	C	C	S
<code>mvwprintw()</code> (3V) -SysV	S		N	S	S	N
<code>mvwscanw()</code> (3V)	C	<code>wscanw()</code> (3V) を参照。	N	C	C	S
<code>mvwscanw()</code> (3V) -SysV	S		N	S	S	N
<code>napms</code> (3V) () (3V) -SysV	S		N	S	S	N
<code>net_addr()</code> (3N)	S		N	N	S	N

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
netname2host() (3N)	S		S	S	S	N
netname2user() (3N)	S		S	S	S	N
newpad() (3V) -SysV	S		N	S	S	N
newterm() (3V) -SysV	S		N	S	S	N
newwin() (3V) -SysV	S		N	S	S	S
nextafter() (3M)	S		N	S	S	N
nextkey() (3X)	A	dbm_nextkey() (3) に置換。	N	N	N	S
nice() (3V)	S		S	S	S	S
nint() (3M)	N		N	N	N	N
nl() (3V) -SysV	S		N	S	S	S
nl_init() (3C)	N		N	N	N	N
nl_langinfo() (3C)	S		S	S	S	N
nlist() (3V)	C	SunOS 4.x では、正常終了時は見つからないシンボルの数を返し、エラー発生時には -1 を返す。SunOS 5.6、SVID または SVR4 では、正常終了時には 0 (ゼロ)、エラー発生時には -1 を返す。SunOS 5.6 では ELF フォーマットのファイル、4.1 では a.out フォーマットのファイルにのみ対応することに注意。	N	C	C	S
nlm_prot() (3R)	S		N	N	N	N
nocbreak() (3V) -SysV	S		N	S	S	S

表 C-7 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>nocrmode()</code> (3X)	S		N	N	S	N
<code>nodelay()</code> (3V) -SysV	S		N	S	S	N
<code>noecho()</code> (3V) -SysV	S		N	S	S	S
<code>nonl()</code> (3V) -SysV	S		N	S	S	S
<code>nonstandard _arithmetic()</code> (3M)	N		N	N	N	N
<code>noraw()</code> (3V) -SysV	S		N	S	S	S
<code>notimeout()</code> (3V) -SysV	S		N	S	S	N
<code>nrand48()</code> (3)	S		N	S	S	N
<code>ntohl()</code> (3N)	S		N	N	S	N
<code>ntohs()</code> (3N)	S		N	N	S	N

ライブラリルーチンのリファレンス (O, P, Q, R)

表 C-8

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
on_exit() (3)		atexit() (3C) に置換。atexit() を使用して登録された関数は、引数なしで呼び出されることに注意。	A	A	A	N
opendir() (3V)	C	SunOS 5.6、ABI、SVID、または SVR4 の DIR 構造体には、SunOS 4.x の dd_bsize および dd_off フィールドが含まれない。また、SunOS 5.6、ABI、SVID または SVR4 の dd_loc および dd_size フィールドは、SunOS 4.x の場合の long 型と違い int 型。 SunOS 5.6、ABI、SVID または SVR4 では、<sys/types.h> に含まれるが、SunOS 4.x では含まれない。SunOS 5.6、ABI、SVID または SVR4 では、ディレクトリ名の引数が空の文字列を指す場合 errno に ENOENT を設定。	C	C	C	N
opendir() (3V) -SysV	S		S	S	S	N
openlog() (3)	S		N	N	N	N
openpl() (3X)	N		N	N	N	N
optarg() (3)	S		N	N	N	N
optind() (3)	S		N	N	N	N
overlay() (3V)	C	SunOS 4.x の overlay() は関数。SunOS 5.6、SVID または SVR4 の overlay(srcwin, dstwin)() は _overlap((srcwin), (dstwin), TRUE)() を呼び出すマクロ。 SunOS 4.x では戻り値は不定。SunOS 5.6、SVID または SVR4 では、正常終了なら OK、そうでない場合は ERR を返す。	N	C	C	S

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
overlay() (3V) -SysV	S		N	S	S	N
overwrite() (3V)	C	SunOS 4.x の overwrite() は関数。SunOS 5.6、SVID または SVR4 の overwrite(srcwin, dstwin)() は _overlap((srcwin), (dstwin), FALSE)() を呼び出すマクロ。 SunOS 4.x では戻り値は不定。SunOS 5.6、SVID または SVR4 では、正常終了なら OK、そうでない場合は ERR を返す。	N	C	C	S
overwrite() (3V) -SysV	S		N	S	S	N
passwd2des() (3R)	S		N	N	N	N
pause() (3V)	S		S	S	S	N
pclose() (3S)	S		S	S	S	N
pechochar() (3V) -SysV	S		N	S	S	N
perror() (3)	S		S	S	S	N
plock() (3)	S		S	S	S	N
plot() (3X)	S		N	N	N	N
pmap_getmaps() (3N)	S	利用可能。ただし、なるべく rpcb_getmaps() (3N) を使用する。	A	A	S	N
pmap_getport() (3N)	S	利用可能。ただし、なるべく rpcb_getaddr() (3N) を使用する。	A	A	S	N
pmap_rmtcall() (3N)	S	利用可能。ただし、なるべく rpcb_rmtcall() (3N) を使用する。	A	A	S	N

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>pmap_set()</code> (3N)	S	利用可能。ただし、なるべく <code>rpcb_set()</code> (3N) を使用する。	A	A	S	N
<code>pmap_unset()</code> (3N)	S	利用可能。ただし、なるべく <code>rpcb_unset()</code> (3N) を使用する。	A	A	S	N
<code>pnoutrefresh()</code> (3V) -SysV	S		N	S	S	N
<code>pnv()</code> (3R)	N		N	N	N	N
<code>pod_getexit()</code> (3L)	N		N	N	N	N
<code>pod_getmaxpri()</code> (3L)	N		N	N	N	N
<code>pod_getmaxsize()</code> (3L)	N		N	N	N	N
<code>pod_setexit()</code> (3L)	N		N	N	N	N
<code>pod_setmaxpri()</code> (3L)	N		N	N	N	N
<code>point()</code> (3X)	S		N	N	N	N
<code>popen()</code> (3S)	S		S	S	S	N

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>pow()</code> (3M)	C	SunOS 5.6、SVID または SVR4 では、 $x == 0$ で y が正でないとき、または $x < 0$ で y が整数でないときに 0 を返す。オーバーフローまたはアンダフローの場合は、それぞれ +HUGE または -HUGE または 0 (ゼロ) を返す。いずれの場合にも <code>errno</code> は設定される。 <code>pow(x, 0.0)()</code> と指定すると、SunOS 4.x では 1 (SunOS 5.6、SVID または SVR4 では記述なし)、 $x < 0$ で y が整数でないときは NaN、 $x == 0$ で $y < 0$ のときは $+\infty$ (無限大) または $-\infty$ を返す。オーバーフローおよびアンダフローの場合は IEEE のインプリメンテーションに依存する値を返す。SunOS 4.x では HUGE が +oo と定義されるため、 <code>pow(10.0, HUGE)()</code> と <code>pow(10.0, -HUGE)()</code> はアンダフローやオーバーフローにはならず <code>errno</code> も設定されない。SunOS 5.6、SVID または SVR4 ソフトウェアでは <code>errno</code> に ERANGE が設定される。	N	C	C	N
<code>prefresh()</code> (3V) -SysV	S		N	S	S	N
<code>printf()</code> (3V)	S	<code>fprintf(3)</code> を参照。	S	S	S	S

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
printw() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると整数 ERR、正常終了すると ERR 以外の整数値を返す。SunOS 4.x では void を返す。SunOS 5.6、SVID または SVR4 では、ヘッダファイル < curses.h> がヘッダファイル < stdio.h> および < unctrl.h> を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
printw() (3V) -SysV	S		N	S	S	N
prof() (3)	A	profil() (2) ルーチンが同様の機能を提供。	A	A	A	N
psignal() (3)	C	sig 引数は、SunOS 4.x では unsigned int 型、SVR4 および SunOS 5.6 では int 型として定義される。	N	N	C	S
putc() (3S)	S		S	S	S	N
putchar() (3S)	S		S	S	S	N
putenv() (3)	S		S	S	S	N
putp() (3V) -SysV	S		N	S	S	N
putpwent() (3)	S		S	S	S	N
puts() (3S)	S		S	S	S	N
putw() (3S)	S		S	S	S	N
pwdauth() (3)	N		N	N	N	N

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
qsort() (3)	S		S	S	S	N
quiet_nan() (3M)	N		N	N	N	N
rand() (3V)	S		S	S	S	S
random() (3)	A	現在は random() (3C)。SunOS 5.6、SVID または SVR4 ソフトウェア用の drand48() (3C)、または rand() (3C) ルーチンが同様の機能を提供。	A	A	A	S
raw() (3V) -SysV	S		N	S	S	S
rcmd() (3N)	S		N	N	S	N
readdir() (3V)	C	SunOS 4.x、および ABI と SVID での dirent 構造体のみ、共通に d_name フィールドがある。SunOS 4.x の readdir() は、<sys/dir.h> で定義された古いデータ構造体 dirent をサポートする。この構造体は SunOS 5.6、ABI、SVID、または SVR4 ソフトウェアではサポートされない。<sys/dir.h> で定義された古いデータ構造体 direct を使用しているアプリケーションは、<dirent.h> で定義されている dirent 構造体に移行する必要がある。SunOS 5.6、ABI、SVID、または SVR4 の readdir() は、アクセスの最後にディレクトリを更新する。SunOS 4.x、SVR4、および SunOS 5.6 の dirent 構造体のみ、共通に d_name および d_reclen フィールドがある。また、SunOS 5.6 の dd_loc および dd_size フィールドは、SunOS 4.x の場合の long 型と違い int 型。	C	C	C	S

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
readdir() (3V) -SysV	C	SunOS 4.x、SVR4、および SunOS 5.6 の dirent 構造体のみ、共通に d_name および d_reclen フィールドがある。また、SunOS 5.6 の dd_loc および dd_size フィールドは、SunOS 4.x ソフトウェアの場合の long 型と違い int 型。SunOS 5.6、ABI、SVID、または SVR4 の readdir() は、アクセスの最後にディレクトリを更新する。SunOS 4.x、および ABI と SVID の dirent 構造体のみ、共通に d_name フィールドがある。	C	C	C	N
realloc() (3)	C	SunOS 4.x では、malloc()、calloc()、および realloc() を最後に呼び出した後で解放されたブロックへのポインタを受け付ける。SunOS 5.6、ABI、SVID、または SVR4 では同様のポインタを受け付けない。	C	C	C	N
realpath() (3N)	S		N	N	S	N
re_comp() (3)	A	現在は re_comp() (3C)。ABI および SVID では、汎用の正規表現マッチングルーチン regexp() (3) が同様の機能を提供。SunOS 5.6 では re_comp() (3G) に置換。	A	A	A	S
re_exec() (3)	A	現在は re_exec() (3C)。ABI および SVID では、汎用の正規表現マッチングルーチン regexp() (3) が同様の機能を提供。SunOS 5.6 では regex() (3G) に置換。	A	A	A	S
refresh() (3V) -SysV	S		N	S	S	S
registerrpc() (3N)	S	利用可能。ただし、なるべく rpc_reg() (3C) を使用する。	N	N	S	N

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
remainder() (3M)	S		S	S	S	N
remexportent() (3)	N	/etc/dfs/sharetab ファイルが /etc/exports を置換。詳細については share(1M)、unshare(1M)、sharetab(4) を参照。	N	N	N	N
remque() (3)	S		N	N	S	N
resetterm() (3V)	A	reset_shell_mode() (3) に置換。	N	A	A	N
res_init() (3)	S		N	N	S	N
res_mkquery() (3)	S		N	N	S	N
res_send() (3)	S		N	N	S	N
reset_prog_mode() (3V) -SysV	S		N	S	S	N
reset_shell_mode() (3V) -SysV	S		N	S	S	N
resetty() (3V) -SysV	S		N	S	S	S
restartterm() (3V) -SysV	S		N	S	S	N
rewind() (3S)	S		S	S	S	N
rewinddir() (3V)	S		S	S	S	N
rex() (3R)	S		N	N	N	N
rexec() (3N)	S		N	N	S	N
rindex() (3)	S	現在は rindex() (3)。	A	A	A	S
rint() (3M)	S		N	N	S	N

表 C-8 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
ripoffline() (3V) -SysV	S		N	S	S	N
rnusers() (3R)	N		N	N	N	N
rpc_createerr() (3N)	S		S	S	S	N
rpow() (3X)	S		N	N	N	N
rquota() (3R)	N		N	N	N	N
rresvport() (3N)	S		N	N	S	N
rstat() (3R)	N		N	N	N	N
rtime() (3N)	N		N	N	N	S
ruserok() (3N)	S		N	N	N	N
rusers() (3R)	S		N	N	S	N
rwall() (3R)	S		N	N	S	N

ライブラリルーチンのリファレンス (S)

表 C-9

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>_setjmp()</code> (3)	<code>sigsetjmp()</code>	現在は <code>setjmp()</code> (3C)。 <code>sigsetjmp()</code> (3) <code>savemask</code> 引数がゼロのとき <code>sigsetjmp(3)</code> ルーチンが 同じ機能を提供。このルーチ ンは、呼び出し元プロセスの レジスタおよびスタック環境 を保存するが、 <code>signalmask</code> は 保存しない。	A	A	A	S
<code>SAMECV()</code> (3L)	N		N	N	N	N
<code>SAMEMON()</code> (3L)	N		N	N	N	N
<code>SAMETHREAD()</code> (3L)	N		N	N	N	N
<code>STKTOP()</code> (3L)	N		N	N	N	N
<code>saveterm()</code> (3V)	A	<code>def_prog_mode()</code> (3X) に 置換。	N	A	A	N
<code>savetty()</code> (3V) <code>-SysV</code>	S		N	S	S	S
<code>scalb()</code> (3M)	C	SunOS 5.6、SVID または SVR4 では、 $x * (r^{**n})$ の値を 計算。ここで r はマシンの浮 動小数点演算の基数。 $r == 2$ のとき、 <code>ldexp()</code> (3M) ルー チンと同等。オーバーフローの 場合は <code>+HUGE</code> または <code>-HUGE</code> (x の符号による) を返す。アン ダフローの場合は 0 (ゼロ) を返し <code>errno</code> を設定する。 SunOS 4.x では、常に $x *$ (2^{**n}) の値を計算。 y が整数 でないとき戻り値は不定。	N	C	C	N
<code>scalbn()</code> (3M)	S		N	N	N	N
<code>scandir()</code> (3)	N		N	N	N	S
<code>scanf()</code> (3V)	S		S	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
scanw() (3V)	C	SunOS 5.6、SVID または SVR4 では、ヘッダファイル < curses.h > がヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルード し、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
scanw() (3V) -SysV	S		N	S	S	N
scr_dump() (3V) -SysV	S		N	S	S	N
scr_init() (3V) -SysV	S		N	S	S	N
scr_restore() (3V) -SysV	S		N	S	S	N
scroll() (3V)	C	異常終了すると ERR、正常終了すると不定値を返す。 SunOS 4.x では、異常終了すると ERR、正常終了すると OK (0) を返す。SunOS 5.6、SVID または SVR4 では、ヘッダファイル < curses.h > がヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルード し、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
scroll() (3V) -SysV	S		N	S	S	N
scrollok() (3V) -SysV	S		N	S	S	S
seconvert() (3)	S		N	N	N	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
seed48() (3)	S		N	S	S	N
seekdir() (3V)	S		S	S	S	N
setac() (3)	N		N	N	N	N
setbuf() (3V)	S		S	S	S	S
setbuffer() (3V)	S		N	N	N	S
set_curterm() (3V) -SysV	S		N	S	S	N
setegid() (3V)	S		N	N	N	N
seteuid() (3V)	S		N	N	N	N
setexportent() (3)	A	/etc/dfs/sharetab ファイルが /etc/exports を置換。詳細については share(1M)、unshare(1M)、 sharetab(4) のマニュアル ページを参照。	N	N	N	N
setfsent() (3)	A	fopen() (3) に置換。	A	A	A	N
setgid() (3V)	S		S	S	S	N
setgraent() (3)	N		N	N	N	N
setgrent() (3V)	S		S	S	S	N
sethostent() (3N)	S		N	N	S	N
setjmp() (3V)	S		S	S	S	S
setkey() (3)	S		N	S	S	N
setlinebuf() (3V)	S		N	N	N	S
setlocale() (3V)	C		S	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
setlogmask() (3)	S		N	N	N	N
setmntent() (3)	A	fopen() (3) に続けて lockf() (3) ルーチンを使用 するのと同様の機能を提供。	A	A	A	N
setnetent() (3N)	S		N	N	S	N
setnetgrent() (3N)	S		N	N	N	N
setprotoent() (3N)	S		N	N	S	N
setpwaent() (3)	N		N	N	N	N
setpwent() (3V)	S		S	S	S	N
setpwfile() (3V)	N		N	N	N	N
setrgid() (3V)	A	setgid() (2) に置換。	A	A	A	N
setrpcent() (3N)	S		N	N	S	N
setruid() (3V)	A	setuid() (2) に置換。	A	A	A	N
setscrreg() (3V) -SysV	S		N	S	S	N
setservernt() (3N)	S		N	N	S	N
setstate() (3)	S	現在は setstate() (3C)	N	A	A	S
setsyx() (3V) -SysV	S		N	S	S	N
set_term() (3V) -SysV	S		N	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
setterm() (3V)	C	SunOS 4.x および SunOS 5.6 の両方のリリースで <code>setupterm()</code> に置換される、古い呼び出し形式。 <code>cursor_tinfo()</code> (3X) を参照すること。 <code>setupterm(term, 1, (int *) 0)()</code> は <code>setterm(term)()</code> と同様の機能を提供。	N	C	C	S
setterm() (3V) -SysV	S		N	S	S	N
setttyent() (3)	N	SunOS 5.6 の tty システムに関する情報は、 <code>ttymon(1)</code> および <code>ttydefs(4)</code> のマニュアルページを参照。	N	N	N	N
setuid() (3V)	S		S	S	S	N
setupterm() (3V) -SysV	S		N	S	S	N
setusershell() (3)	S		N	N	N	N
setvbuf() (3V)	S		S	S	S	S
sfconvert() (3)	S		N	N	N	N
sgconvert() (3)	S		N	N	N	N
sigaction() (3V)	S		S	S	S	N
sigaddset() (3V)	S		S	S	S	N
sigdelset() (3V)	S		S	S	S	N
sigemptyset() (3V)	S		S	S	S	N
sigfillset() (3V)	S		S	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
sigfpe() (3)	S		N	N	N	N
siginterrupt() (3V)	A	sigaction() (2) ルーチンが同様の機能を提供。	A	A	A	S
sigismember() (3V)	S		S	S	S	N
siglongjmp() (3V)	S		S	S	S	N
signal() (3V)	C	SunOS 4.x のシグナルである SIGLOST は、SVR4 および SunOS 5.6 の signal() (2) ルーチンでは未定義。 SunOS 4.x のシグナルである SIGIO、SIGURG、SIGFSZ、SIGVTALRM、SIGPROF、SIGLOST は、ABI および SVID の signal ルーチンでは未定義。	C	C	C	S
ssignal() (3V)	C		C	C	C	N
signaling_nan() (3M)	N		N	N	N	N
signbit() (3M)	N		N	N	N	N
significand() (3M)	N		N	N	N	N
sigsetjmp() (3V)	S		S	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>sin()</code> (3M)	C	SunOS 5.6、あるいは SVID または SVR4 では、引数が 0 (ゼロ) よりかなり小さい場合、有効桁が失われるため 0 (ゼロ) を返す。この場合、標準出力に TLOSS (<code>matherr()</code> (3M) を参照) を示すメッセージが出力される。有効桁の一部が失われると PLOSS エラーが生成されるが、出力は行われない。いずれの場合も <code>errno</code> に ERANGE が設定される。SunOS 4.x では、引数の絶対値が $\pi/4$ を超えると、ソフトウェアまたはハードウェアにおいて引数が減少する。 <code><math.h></code> で定義されている <code>fp_pi</code> 変数により、精度は実行時に変更可能。例外エラーはどちらも IEEE 754 に基づき生成される。	N	C	C	N
<code>sinh()</code> (3M)	S		N	S	S	N
<code>single_precision()</code> (3M)	N		N	N	N	N
<code>single_to_decimal()</code> (3)	S		N	N	N	N
<code>sleep()</code> (3V)	S		S	S	S	S
<code>slk_clear()</code> (3V) -SysV	S		N	S	S	N
<code>slk_init()</code> (3V) -SysV	S		N	S	S	N
<code>slk_label()</code> (3V) -SysV	S		N	S	S	N
<code>slk_noutrefresh()</code> (3V) -SysV	S		N	S	S	N
<code>slk_refresh()</code> (3V) -SysV	S		N	S	S	N
<code>slk_restore()</code> (3V) -SysV	S		N	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
slk_set() (3V) -SysV	S		N	S	S	N
slk_touch() (3V) -SysV	S		N	S	S	N
sm_inter() (3R)	S		N	N	N	N
space() (3X)	S		N	N	N	N
spray() (3R)	S		N	N	S	N
sprintf() (3V)	S	fprintf(3) を参照。	S	S	S	S
sqrt() (3M)	C	SunOS 4.x ソフトウェアでは、関数の結果が未定義の値になると (sqrt(-3.0)() など)、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。 SunOS 5.6、SVID または SVR4 では EDOM エラーおよび DOMAIN 数値演算エラーの結果として 0 (ゼロ) を返す。	N	C	C	N
srand() (3V)	C	引数 seed は、SunOS 4.x ソフトウェアでは int 型、SunOS 5.6、ABI、SVID、または SVR4 では unsigned int 型として定義される。	C	C	C	S
srand48() (3)	S		N	S	S	N
srandom() (3)	S	現在は srandom() (3C)。 srand48() (3C) ルーチン (SunOS 5.6、SVID または SVR4 の場合)、または srand() (3C) が同様の機能を提供。	A	A	A	S
sscanf() (3V)	S		S	S	S	N
ssignal() (3)	S		N	N	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
standard_arithmetic() (3M)	N		N	N	N	N
standend() (3V) -SysV	S		N	S	S	S
standout() (3V) -SysV	S		N	S	S	S
store() (3M)	A	dbm_store() (3) に置換。	N	N	N	S
strcasecmp() (3)	S		N	N	N	N
strcat() (3)	S		S	S	S	N
strchr() (3)	S		S	S	S	N
strcmp() (3)	S		S	S	S	N
strcoll() (3)	S		S	S	S	N
strcpy() (3)	S		S	S	S	N
strcspn() (3)	S		S	S	S	N
strdup() (3)	S		S	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>strftime()</code> (3V)	C	<p>次のフォーマットで指定される指令語における相違点は下記のとおりに。</p> <p><code>%k</code> と <code>%l</code> は SunOS 5.6 ではサポートされない。<code>%s</code> は、秒を SunOS 4.x では 0~59 の範囲内で指定。SunOS 5.6 では 0~61 の範囲内 (うるう秒を許容する) で定義。<code>%U</code> と <code>%W</code> は、週番号を指定する場合、SunOS 4.x における 01 は 4 日以上を含む 1 月の最初の週を示す。SunOS 5.6 における 01 は、<code>%U</code> は日曜日から、<code>%W</code> は月曜日からはまる 1 月の最初の週を示す。</p> <p>SunOS 4.1 の <code>tm</code> 構造体には、SunOS 5.6 の <code>tm</code> 構造体に存在しない 2 つのフィールド、<code>tm_zone</code> および <code>tm_gmtoff</code> がある。そのかわり SunOS 5.6 では、GMT と地方標準時間との差 (単位は秒) が外部変数 <code>timezone</code>、夏時間を適用するかどうか外部変数 <code>daylight</code> で示される。</p> <p>さらに SunOS では、標準時間および夏時間の時間帯名が外部変数 <code>tzname</code> に格納される。これらの外部変数 (<code>timezone</code>、<code>daylight</code>、および <code>tzname</code>) は、SunOS 4.x の System V インストールオプションである <code>ctime()</code> (3V) ライブラリルーチンによりサポートされる。</p>	C	C	C	N
<code>string_to_decimal()</code> (3)	N		N	N	N	N
<code>strlen()</code> (3)	S		S	S	S	N
<code>strncasecmp()</code> (3)	S		N	N	N	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>strncat()</code> (3)	S		S	S	S	N
<code>strncmp()</code> (3)	S		S	S	S	N
<code>strncpy()</code> (3)	S		S	S	S	N
<code>strpbrk()</code> (3)	S		S	S	S	N
<code>strptime()</code> (3V)	S		A	A	A	N
<code>strrchr()</code> (3)	S		S	S	S	N
<code>strspn()</code> (3)	S		S	S	S	N
<code>strtod()</code> (3)	C	SunOS 4.x の <code>strtod()</code> および <code>atof()</code> ルーチンは、 <i>inf_form</i> 、 <i>infinity_form</i> 、 <i>nan_form</i> 、および <i>nanstring_form</i> を受け付けるが、SunOS 5.6、ABI、SVID、または SVR4 ではこれらの形式を受け付けない。	C	C	C	N
<code>strtok()</code> (3)	S		S	S	S	N
<code>strtol()</code> (3)	S		S	S	S	N
<code>strxfrm()</code> (3)	S		S	S	S	N
<code>stty()</code> (3C)	A	<code>termio(7)</code> インタフェースが同様の機能を提供。	A	A	A	N
<code>subpad()</code> (3V) -SysV	S		N	S	S	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
subwin() (3V)	C	SunOS 5.6、SVID または SVR4 ルーチンは、異常終了すると NULL ポインタを返す。SunOS 5.6、SVID または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
subwin() (3V) -SysV	S		N	S	S	N
svc_destroy() (3N)	S		S	S	S	N
svc_fds() (3N)	S	利用可能。ただし、なるべく svc_fdset() (3N) を使用する。	N	S	S	N
svc_fdset() (3N)	S		S	S	S	N
svc_freeargs() (3N)	S		S	S	S	N
svc_getargs() (3N)	S		S	S	S	N
svc_getcaller() (3N)	S	利用可能。ただし、なるべく svc_getrpcaller() (3N) を使用する。	A	A	A	N
svc_getreq() (3N)	S	利用可能。ただし、なるべく svc_getreqset() (3N) を使用する。	S	S	S	N
svc_getreqset() (3N)	S		S	S	S	N
svc_register() (3N)	A	利用可能。ただし、なるべく svc_reg() (3N) を使用する。	A	A	A	N

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
svc_run() (3N)	S		S	S	S	N
svc_sendreply() (3N)	S		S	S	S	N
svc_unregister() (3N)	A	利用可能。ただし、なるべく svc_unreg() (3N) を使用する。	A	A	A	N
svcerr_auth() (3N)	S		S	S	S	N
svcerr_decode() (3N)	S		S	S	S	N
svcerr_noproc() (3N)	S		S	S	S	N
svcerr_noprogram() (3N)	S		S	S	S	N
svcerr_progvers() (3N)	S		S	S	S	N
svcerr_systemerr() (3N)	S		S	S	S	N
svcerr_weakauth() (3N)	S		S	S	S	N
svcfld_create() (3N)	A	利用可能。ただし、なるべく svc_fd_create() (3N) を使用する。	A	A	A	S
svccraw_create() (3N)	S	利用可能。ただし、なるべく svc_raw_create() (3N) を使用する。	N	N	S	N
svctcp_create() (3N)	S	利用可能。 ただし、なるべく svc_create() (3N)、 svc_tli_create() (3N) および svc_vc_create() (3N) を使用する。	N	N	S	S
svcudp_bufcreate() (3N)	S	利用可能。ただし、なるべく svc_tli_create() (3N) と svc_dg_create() (3N) を使用する。	N	N	S	S

表 C-9 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
svcdp_create() (3N)	S	利用可能。ただし、なるべく svc_create() (3N)、 svc_tli_create() (3N) および svc_dg_create() (3N) を 使用する。	N	N	S	S
swab() (3)	S		S	S	S	N
sys_siglist() (3)	N	psignal() (3C) を使用。	N	N	N	S
syslog() (3)	S		N	N	S	N
system() (3)	S		S	S	S	N

ライブラリルーチンのリファレンス (T)

表 C-10

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
_tolower() (3V) -SysV	S		S	S	S	N
_toupper() (3V) -SysV	S		S	S	S	N
t_accept() (3N)	S		S	S	S	N
t_alloc() (3N)	S		S	S	S	N
t_bind() (3N)	S		S	S	S	N
t_close() (3N)	S		S	S	S	N

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
t_connect() (3N)	S		S	S	S	N
t_error() (3N)	S		S	S	S	N
t_free() (3N)	S		S	S	S	N
t_getinfo() (3N)	S		S	S	S	N
t_getstate() (3N)	S		S	S	S	N
t_listen() (3N)	S		S	S	S	N
t_look() (3N)	S		S	S	S	N
t_open() (3N)	S		S	S	S	N
t_optmgmt() (3N)	S		S	S	S	N
t_rcv() (3N)	S		S	S	S	N
t_rcvconnect() (3N)	S		S	S	S	N
t_rcvdis() (3N)	S		S	S	S	N
t_rcvrel() (3N)	S		S	S	S	N
t_rcvudata() (3N)	S		S	S	S	N
t_rcvuderr() (3N)	S		S	S	S	N
t_snd() (3N)	S		S	S	S	N
t_snddis() (3N)	S		S	S	S	N
t_sndrel() (3N)	S		S	S	S	N
t_sndudata() (3N)	S		S	S	S	N
t_sync() (3N)	S		S	S	S	N

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
t_unbind() (3N)	S		S	S	S	N
tan() (3M)	S		N	S	S	N
tanh() (3M)	S		N	S	S	N
tcdrain() (3V)	S		S	S	S	N
tcflow() (3V)	S		S	S	S	N
tcflush() (3V)	S		S	S	S	N
tcgetattr() (3V)	S		S	S	S	N
tcgetpgrp() (3V)	S		S	S	S	N
tcsendbreak() (3V)	S		S	S	S	N
tcsetattr() (3V)	S		S	S	S	N
tcsetpgrp() (3V)	S		S	S	S	N
tdelete() (3)	S		S	S	S	N
telldir() (3V)	S		S	S	S	N
tempnam() (3S)	S		S	S	S	N
textdomain() (3)	N		N	N	N	N
tfind() (3)	S		S	S	S	N
tgetent() (3X)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。 SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
tgetent() (3V) -SysV	S		N	S	S	N
tgetflag() (3X)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。 SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
tgetflag() (3V) -SysV	S		N	S	S	N
tgetnum() (3X)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。 SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
tgetnum() (3V) -SysV	S		N	S	S	N
tgetstr() (3X)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。 SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
tgetstr() (3V) -SysV	S		N	S	S	N

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
tgoto() (3X)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
tgoto() (3V) -SysV	S		N	S	S	N
tigetflag() (3V) -SysV	S		N	S	S	N
tigetnum() (3V) -SysV	S		N	S	S	N
tigetstr() (3V) -SysV	S		N	S	S	N
time() (3V)	S		S	S	S	N
timegm() (3V)	A	mktime() (3C) に置換。	A	A	A	N
timelocal() (3V)	S	localtime() (3C) に置換。	A	A	A	N
times() (3V)	C	SunOS 4.x は、1/HZ 秒の単位で時間値を返す (HZ は 60)。SunOS 5.6、ABI、SVID、または SVR4 は、1/CLK_TCK 秒の単位で時間値を返す。	C	C	C	S
timezone() (3C)	S		N	N	N	N
tmpfile() (3S)	C		C	C	C	N
tmpnam() (3S)	S		S	S	S	N
toascii() (3V)	S		S	S	S	N
toascii() (3V) -SysV	S		S	S	S	N
tolower() (3V)	S		S	S	S	N

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>tolower()</code> (3V) -SysV	C	SunOS 5.6、ABI、SVID、または SVR4 では、 <code>LC_CTYPE</code> で指定されるプログラムのロケールの影響を受ける。SunOS 4.x では影響なし。	C	C	C	N
<code>touchline()</code> (3V)	C	SunOS 5.6、ABI、SVID、または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
<code>touchline()</code> (3V) -SysV	S		N	S	S	N
<code>touchoverlap()</code> (3X)	N		N	N	N	S
<code>touchwin()</code> (3V)	C	SunOS 5.6、ABI、SVID、または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
<code>touchwin()</code> (3V) -SysV	S		N	S	S	N
<code>toupper()</code> (3V)	S		S	S	S	N
<code>toupper()</code> (3V) -SysV	C	SunOS 5.6、ABI、SVID、または SVR4 では、 <code>LC_CTYPE</code> で指定されるプログラムのロケールの影響を受ける。SunOS 4.x では影響なし。	C	C	C	N
<code>tparam()</code> (3V) -SysV	S		N	S	S	N
<code>tputs()</code> (3V)	C	SunOS 5.6、SVID または SVR4 では、変換のための補助用としてこのルーチンをサポートしているため、新規のアプリケーションでは使用しないこと。SunOS 5.6 または、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
<code>tputs()</code> (3V) -SysV	S		N	S	S	N

表 C-10 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
traceoff() (3V) -SysV	S		N	N	S	N
traceon() (3V) -SysV	S		N	N	S	N
tsearch() (3)	S		S	S	S	N
ttyname() (3V)	S		S	S	S	N
ttyslot() (3V)	S		N	N	S	N
twalk() (3)	S		S	S	S	N
typeahead() (3V) -SysV	S		N	S	S	N
tzset() (3V)	C	ctime() (3V) を参照。	C	C	C	N
tzsetwall() (3V)	A	tzset() (3C) に置換。	A	A	A	N

ライブラリルーチンのリファレンス (U, V)

表 C-11

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>ualarm()</code> (3)	S	現在は <code>ualarm()</code> (3C)。 <code>ITIMER_REAL</code> に引数 <i>which</i> を設定した <code>setitimer()</code> (2) システムコールが同様の機能を提供。	N	A	A	S
<code>ulimit()</code> (3C)	S	SVR4 および SunOS 5.6 の <code>ulimit()</code> は、SunOS 4.x の <code>ulimit()</code> と互換性がある。SunOS 4.x の <code>ulimit()</code> ルーチンの整数 <i>cmd</i> の値 1 および 2 は同等な SVID の <code>ulimit()</code> ルーチンのシンボリック定数 <i>cmd</i> の値 <code>UL_GETFSIZE</code> および <code>UL_SETFSIZE</code> と互換性のない場合がある。また、SVID の <code>ulimit()</code> ルーチンは、3 (最大のブレイク値を取得) および 4 (プロセスのファイル記述子テーブルのサイズを取得) の機能をサポートしない。	C	C	S	N
<code>unctrl()</code> (3V) -SysV	S		N	S	S	S
<code>ungetc()</code> (3S)	S	SVR4 および SunOS 5.6 の <code>ungetc()</code> は必ず 4 文字をプッシュバックするため、SunOS 4.x の <code>ungetc()</code> と互換性がある。SunOS 4.x の <code>ungetc()</code> は、前に <code>read</code> 文がなくても必ず標準入力に 1 文字をプッシュバックするが、ABI および SVID の <code>ungetc()</code> はこの属性をサポートしていない。	C	C	S	N
<code>ungetch()</code> (3V) -SysV	S		N	S	S	N
<code>user2netname()</code> (3N)	S		S	S	S	N
<code>usleep()</code> (3)	S	現在は <code>usleep()</code> (3C)。 <code>setitimer()</code> (2) または <code>select()</code> (3C) が同様の機能を提供。	N	A	A	S

表 C-11 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
utime() (3V)	C	SunOS 4.x と、SunOS 5.6、ABI、SVID、または SVR4 では、第 2 引数の型が異なる。SunOS 4.x では、引数 <i>timep</i> は 2 つの <i>time_t</i> 値の配列を指すが、SunOS 5.6、ABI、SVID、または SVR4 では、引数 <i>times</i> が <i>utimbuf</i> 構造体 (2 つの <i>time_t</i> メンバを含む) を指す。	C	C	C	N
valloc() (3)	S		N	N	S	N
varargs() (3)	S		N	N	N	N
vfprintf() (3V)	C	<i>vprintf()</i> (3V) を参照。	C	C	C	S
vidattr() (3V) -SysV	S		N	S	S	N
vidputs() (3V) -SysV	S		N	S	S	N
vlimit() (3C)	A	<i>getrlimit()</i> (2) に置換。	A	A	A	N
vprintf() (3V)	C	<i>vprintf()</i> 、 <i>vfprintf()</i> 、および <i>vsprintf()</i> ルーチンでは、可変フォーマットリストが異なるため、SunOS 4.x と SunOS 5.6、ABI、SVID、または SVR4 との互換性はない。SunOS 4.x では、関数ヘッダで <i>va_list</i> (<varargs.h> で定義されている) を使用し、可変引数リスト (たとえば <i>void function(va_alist)</i> など) を宣言する。SunOS 5.6、ABI、SVID、または SVR4 では、<stdarg.h> での定義を関数ヘッダで使用し、可変引数リスト (たとえば <i>void function(int arg1)</i> など) を宣言する。	C	C	C	S
vsprintf() (3V)	C	<i>vprintf()</i> (3V) を参照。	C	C	C	S

表 C-11 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>vsyslog()</code> (3)	S	<code>syslog()</code> (3) に置換。	N	N	N	N
<code>vtimes()</code> (3C)	A	<code>getrusage()</code> (2) に置換。	N	N	N	N
<code>vwprintw()</code> (3V) -SysV	S		N	S	S	N
<code>vwscanw()</code> (3V) -SysV	S		N	S	S	N

ライブラリルーチンのリファレンス (W)

表 C-12

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
<code>waddch()</code> (3V)	C	CHTYPE が <code><curses.h></code> で別に定義されていないならば、SunOS 5.6 における文字は <code>chtype (long)</code> 。SunOS 5.6、SVID、または SVR4 では、異常終了すると ERR を、正常終了すると ERR 以外の整数値を返す。SunO 5.6、SVID、または SVR4 のヘッダファイル <code><curses.h></code> は、ヘッダファイル <code><stdio.h></code> および <code><unctrl.h></code> を自動的にインクルードし、 <code>CURS_PERFORMANCE</code> が定義されているならば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
<code>waddch()</code> (3V) -SysV	S		N	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
waddstr() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR (-1) を返す。SunOS 5.6、SVID、または SVR4 のヘッダファイル <curses.h> は、ヘッダファイル <stdio.h> および <unctrl.h> を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
waddstr() (3V) -SysV	S		N	S	S	N
wattroff() (3V) -SysV	S		N	S	S	N
wattron() (3V) -SysV	S		N	S	S	N
wattrset() (3V) -SysV	S		N	S	S	N
wclear() (3V)	C	SunOS 5.6、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4.x は void を返す。SunOS 5.6、SVID、または SVR4 のヘッダファイル <curses.h> は、ヘッダファイル <stdio.h> および <unctrl.h> を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
wclear() (3V) -SysV	S		N	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wclrrobot() (3V)	C	SunOS 5.6、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4.x では void を返す。SunOS 5.6、SVID、または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
wclrrobot() (3V) -SysV	S		N	S	S	N
wclrtoeol() (3V)	C	SunOS 5.6、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4.x では void を返す。SunOS 5.6、SVID、または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
wclrtoeol() (3V) -SysV	S		N	S	S	N
wcstombs() (3)	S	wchar_t のサイズは SunOS 4.x は短く、SunOS 5.6 は長い。	S	S	S	N
wctomb() (3)	S	wchar_t のサイズは SunOS 4.x は短く、SunOS 5.6 は長い。	S	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wdelch() (3V)	C	SunOS 5.6、SVID または SVR4 ではマクロの場合があるが、SunOS 4.x では常にマクロ。SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
wdelch() (3V) -SysV	S		N	S	S	N
wdeleteln() (3V)	C	SunOS 5.6、SVID または SVR4 ではマクロの場合があるが、SunOS 4.x では常にマクロ。SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
wdeleteln() (3V) -SysV	S		N	S	S	N
wechochar() (3V) -SysV	S		N	S	S	N
werase() (3V)	C	SunOS 5.6、SVID または SVR4 においては、 <i>immedok</i> が設定されていれば OK(0) または負でない整数値を返す。SunOS 5.6、SVID または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
werase() (3V) -SysV	S		N	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wgetch() (3V)	C	SunOS 5.6、SVID、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。 SunOS 5.6、SVID または SVR4 では、追加でファンクションキーもサポート。	N	C	C	S
wgetch() (3V) -SysV	C	SunOS 5.6、SVID または SVR4 において、ウィンドウがパッドでなく wrefresh() を最後に呼び出した後で移動または修正された場合、別の文字が読み込まれる前に wrefresh() が呼び出される。SunOS 4.x では、このような状況で wrefresh() は呼び出されない。	N	C	C	N
wgetstr() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
wgetstr() (3V) -SysV	C	getstr(3V) -Sys V() を参照。	N	C	C	N
winch() (3V) -SysV	S		N	S	S	S
winsch() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
winsch() (3V) -SysV	S		N	S	S	N
winsertln() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。SunOS 5.6、SVID または SVR4 ではマクロである場合がある。	N	C	C	S
winsertln() (3V) -SysV	S		N	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wmove() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。	N	C	C	S
wmove() (3V) -SysV	S		N	S	S	N
wnoutrefresh() (3V) -SysV	S		N	S	S	N
wprintw() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると ERR を返し、正常終了すると ERR 以外の整数値を返す。SunOS 4.x では void を返す。SunOS 5.6、SVID または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
wprintw() (3V) -SysV	S		N	S	S	N
wrefresh() (3V)	C	SunOS 5.6、SVID または SVR4 では、異常終了すると (ERR = -1) を返し、正常終了するとその他の整数を返す。一方 SunOS 4.x では void を返す。SunOS 5.6、SVID または SVR4 のヘッダファイル < curses.h > は、ヘッダファイル < stdio.h > および < unctrl.h > を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wrefresh() (3V) -SysV	S		N	S	S	N
wscanw() (3V)	C	SunOS 5.6、SVID または SVR4 では、呼び出しによりマッピングされたフィールドの数を含む int 型の値を返す。一方 SunOS 4.x では void を返す。SunOS 5.6、SVID または SVR4 のヘッダファイル < curses.h> は、ヘッダファイル < stdio.h> および < unctrl.h> を自動的にインクルードし、CURS_PERFORMANCE が定義されていれば、性能を上げるために最も一般的に使用されるルーチンをマクロとして定義する。	N	C	C	S
wscanw() (3V) -SysV	S		N	S	S	N
wsetscrreg() (3V) -SysV	S		N	S	S	N
wstandend() (3V)	C	このルーチンは、attrset(0)() を使用しウィンドウ属性をすべてクリアする curses() (3V) 関数。SunOS 4.x では常に不定値を返す。一方、SunOS 5.6、SVID または SVR4 の standout() ルーチンは、常に 1 (正常終了) を返す。	N	C	C	S
wstandend() (3V) -SysV	S		N	S	S	N

表 C-12 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
wstandout () (3V)	C	このルーチンは、ウィンドウに対して最適な <code>standout</code> モードの端末を有効にするため <code>A_STANDOUT</code> 属性を設定する <code>curses () (3V)</code> 関数。SunOS 4.x では、この関数に <code>attron(A_STANDOUT) ()</code> を使用し、不定値を返す。SunOS 5.6、SVID または SVR4 の <code>standout ()</code> ルーチンは <code>attron(A_STANDOUT) ()</code> と同等で、常に 1 (正常終了) を返す。	N	C	C	S
wstandout () (3V) -SysV	S		N	S	S	N

ライブラリルーチンのリファレンス (X, Y)

表 C-13

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
xcrypt () (3R)	N		N	N	N	N
xdecrypt () (3R)	N		N	N	N	N
xdr_accepted_reply () (3N)	S		S	S	S	N
xdr_array () (3N)	S		S	S	S	N
xdr_authunix_parms () (3N)	S	利用可能。 ただし SunOS 5.6、ABI、SVID または SVR4 ではなるべく <code>xdr_authsys_parms () (3N)</code> を使用する	A	A	A	N

表 C-13 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
xdr_bool() (3N)	S		S	S	S	N
xdr_bytes() (3N)	S		S	S	S	N
xdr_callhdr() (3N)	S		S	S	S	N
xdr_callmsg() (3N)	S		S	S	S	N
xdr_enum() (3N)	S		S	S	S	N
xdr_float() (3N)	S		S	S	S	N
xdr_free() (3N)	S		S	S	S	N
xdr_getpos() (3N)	S		S	S	S	N
xdr_inline() (3N)	S		S	S	S	N
xdr_int() (3N)	S		S	S	S	N
xdr_long() (3V)	S		S	S	S	N
xdr_opaque() (3N)	S		S	S	S	N
xdr_pointer() (3N)	S		S	S	S	N
xdr_reference() (3N)	S		S	S	S	N
xdr_setpos() (3N)	S		S	S	S	N
xdr_short() (3N)	S		S	S	S	N
xdr_string() (3N)	S		S	S	S	N
xdr_u_char() (3N)	S		S	S	S	N
xdr_u_int() (3N)	S		S	N	S	N
xdr_u_long() (3N)	S		S	S	S	N

表 C-13 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
xdr_u_short() (3N)	S		S	S	S	N
xdr_union() (3N)	S		S	S	S	N
xdr_vector() (3N)	S		S	S	S	N
xdr_void() (3N)	S		S	S	S	N
xdr_wrapstring() (3N)	S		S	S	S	N
xdrmem_create() (3N)	S		S	S	S	N
xdrrec_create() (3N)	S		S	S	S	N
xdrrec_endofrecord() (3N)	S		S	N	S	N
xdrrec_eof() (3N)	S		S	S	S	N
xdrrec_skiprecord() (3N)	S		S	N	S	N
xdrstdio_create() (3N)	S		S	S	S	N
xtom() (3X)	S		N	N	N	N
y0() (3M)	C	SunOS 4.x では、関数の結果が未定義の値になると、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID または SVR4 では、EDOM エラーおよび DOMAIN 数値演算エラーの結果として -HUGE を返す。SunOS 4.x において y0(HUGE)(), y1(HUGE)(), yn(9,HUGE)() と指定すると、エラーは出力されずに 0 (ゼロ) を返す。	N	C	C	N

表 C-13 続く

SunOS 4.x	SunOS 5.6	注	ABI	SVID	SVR4	BSD
y1() (3M)	C	SunOS 4.x では、関数の結果が未定義の値になると、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID または SVR4 では、EDOM エラーおよび DOMAIN 数値演算エラーの結果として -HUGE を返す。SunOS 4.x において y0 (HUGE) (), y1 (HUGE) (), yn (9, HUGE) () と指定すると、エラーは出力されずに 0(ゼロ) を返す。	N	C	C	N
yn() (3M)	C	SunOS 4.x では、関数の結果が未定義の値になると、EDOM エラーおよび DOMAIN 数値演算エラーの結果として NaN を返す。SunOS 5.6、SVID または SVR4 では、EDOM エラーおよび DOMAIN 数値演算エラーの結果として -HUGE を返す。SunOS 4.x において y0 (HUGE) (), y1 (HUGE) (), yn (9, HUGE) () と指定すると、エラーは出力されずに 0(ゼロ) を返す。	N	C	C	N

システムファイルリファレンス

この付録は、システムファイルの情報を表形式で収録しています。この表は SunOS 4.x システムファイルと Solaris 7 での利用方法の違いを示します。

リファレンスの使い方

- インタフェースに C (変更) とマークされている場合、SunOS 4.x と Solaris 7 との違いについて簡単な説明があります。
- インタフェースに S (同じ) とマークされている場合、Solaris 7 インタフェースは SunOS 4.x インタフェースの機能をすべてサポートします。場合によってはインタフェースは拡張されていますが、これは SunOS 4.x インタフェースの完全なスーパーセットと考えることができます。
- インタフェースに N (使用不可) とマークされている場合、その置換については注を読んでください。

すべての Solaris 7 インタフェースに関する詳細な情報については、『*SunOS Reference Manual*』のセクション 4 の「File Formats」を参照してください。

システムファイル

表 D-1 ファイルフォーマットのリファレンス

SunOS 4.x	SunOS 5.6	注
a.out (5)	C	アセンブラとリンクエディタの出力フォーマット
acct (5)	S	実行アカウントリングファイル
aliases (5)	S	sendmail 用のアドレスと別名
ar (5)	S	アーカイブ (ライブラリ) ファイルフォーマット
audit.log (5)	N	セキュリティ監査トレイルファイル
audit_control (5)	N	システム監査デーモンの制御情報
audit_data (5)	N	監査デーモンの現在の情報
auto.home (5)	C	ホームディレクトリ用の自動マウントマップ
auto.master (5)	C	ホームディレクトリ用の自動マウントマップ
bar (5)	N	テープアーカイブのファイルフォーマット
boards.pc (5)	N	DOS ウィンドウ用の ATN と XTN 互換ボード
bootparams (5)	S	ブートパラメータデータベース
cpio (5)	S	cpio アーカイブのフォーマット
crontab (5)	S	定期的なジョブの実行用タイムテーブル
dir (5)	A	ディレクトリのフォーマット
dump (5)	C	増分ダンプフォーマット
environ (5V)	C	ユーザ環境
ethers (5)	S	hostname データベースまたは NIS ドメインへのイーサネットアドレス
exports (5)	A	NFS クライアントにエクスポートするディレクトリ
fbtab (5)	C	フレームバッファテーブル

表 D-1 ファイルフォーマットのリファレンス 続く

SunOS 4.x	SunOS 5.6	注
fcntl(5)	C	ファイル制御オプション
fs(5)	C	4.2 (ufs) ファイルシステムボリュームのフォーマット
fspec(5)	S	テキストファイルのフォーマット指定
fstab(5)	A	静的なファイルシステムのマウントテーブル、マウントされたファイルシステムテーブル
ftputers(5)	S	FTP によって禁止されたユーザのリスト
gettytab(5)	N	端末構成データベース
group(5)	S	グループファイル
group.adjunct(5)	N	グループセキュリティのデータファイル
holidays(5)	C	System V アカウンティングのプライム/非プライムテーブル
hosts(5)	S	ホスト名データベース
hosts.equiv(5)	S	システムとユーザにより信頼されるホスト
indent.pro(5)	N	字下げ用のデフォルトオプション
inetd.conf(5)	S	インターネットサーバのデータベース
internat(5)	N	国際化のためのキーマッピングテーブル
keytables(5)	S	ロードキーとダンプキーのキーボードテーブルの記述
link(5)	N	リンクエディタインタフェース
locale(5)		ロケールデータベース
magic(5)	S	ファイルコマンドのマジックナンバーファイル
mtab(5)	A	マウントされたファイルシステムのテーブル
netgroup(5)	S	ネットワークグループのリスト

表 D-1 ファイルフォーマットのリファレンス 続く

SunOS 4.x	SunOS 5.6	注
netmasks (5)	S	ネットワークマスクデータベース
netrc (5)	S	ftp リモートログインデータ用のファイル
networks (5)	S	ネットワーク名データベース
passwd (5)	C	パスワードファイル
passwd.adjunct (5)	N	ユーザセキュリティデータファイル。shadow (4) を参照
phones (5)	S	リモートホスト電話番号データベース
plot (5)	N	グラフィックスインタフェース
printcap (5)	A	プリンタ情報データベース
proto (5)	S	at 用のプロトタイプジョブファイル
protocols (5)	S	プロトコル名データベース
publickey (5)	S	公開鍵データベース
queuedefs (5)	S	at、batch、および cron 用の待ち行列記述ファイル
rasterfile (5)	S	ラスタイメージ用の Sun のファイルフォーマット
remote (5)	S	リモートホスト記述ファイル
resolv.conf (5)	S	ドメインネームシステム解決用の構成ファイル
rfmaster (5)	N	リモートファイル共有ネームサーバマスタファイル
rgb (5)	N	coloredit 用に利用可能なカラー (名前で分類)
rhosts (5)	S	システムとユーザにより信頼されるホスト
rmtab (5)	S	リモートマウントされたファイルシステムテーブル
rootmenu (5)	A	SunView 用のルートメニュー指定
rpc (5)	S	rpc プログラム番号データベース

表 D-1 ファイルフォーマットのリファレンス 続く

SunOS 4.x	SunOS 5.6	注
sccsfile(5)	S	SCCS 履歴ファイルのフォーマット
services(5)	S	インターネットのサービスと別名
sm(5)	S	in.statd ディレクトリとファイル構造
statmon(5)	S	statd ディレクトリとファイル構造
sunview(5)	A	SunView の初期設定ファイル
svdtab(5)	N	SunView デバイステーブル
syslog.conf(5)	S	syslogd システムログデーモン用の構成ファイル
systems(5)	C	NIS システムファイル
tar(5)	S	テープアーカイブのファイルフォーマット
term(5)	S	nroff 用の端末駆動テーブル
term(5V)	S	コンパイルされた端末ファイルのフォーマット
termcap(5)	S	端末データベース
terminfo(5V)	S	端末データベース
toc(5)	N	オプションクラスタの内容テーブル
translate(5)	N	システムメッセージ翻訳のための入出力ファイル
ttytab(5)	N	端末初期設定データ
types(5)	S	プリミティブシステムのデータ型
tzfile(5)	S	時間帯情報
updaters(5)	S	NIS 更新用の構成ファイル
utmp(5V)	C	ログインレコード

表 D-1 ファイルフォーマットのリファレンス 続く

SunOS 4.x	SunOS 5.6	注
uuencode(5)	S	コード化された uuencode ファイルのフォーマット
vfont(5)	N	フォントフォーマット
vgrindefs(5)	N	vgrind の言語定義データベース
xtab(5)	N	NFS クライアントへエクスポートするディレクトリ
ypaliases(5)	N	sendmail 用の NIS 別名
ypfiles(5)	S	NIS データベースとディレクトリ構造
ypgroup(5)	N	NIS グループファイル
yppasswd(5)	N	NIS パスワードファイル
ypprintcap(5)	N	NIS プリンタ情報データベース

/ と /usr ファイルシステムの変更

この付録では、/ と /usr ファイルシステムのディレクトリの配置を示します。これらのディレクトリにあるコマンドの違いについては、付録 A で説明しています。

/ ファイルシステムの配置

表 E-1 に、SunOS 5.6 の / ファイルシステムの配置を示します。/ ファイルシステムのディレクトリ構成は各システムによって異なります。

表 E-1 / ファイルシステムのディレクトリ

ディレクトリ	説明
/	ファイルシステムの名前空間全体のルート
/dev	特殊ファイルの一次格納位置
/dev/dsk	ブロックディスクデバイス
/dev/rdsk	raw ディスクデバイス
/dev/pts	擬似端末スレーブデバイス
/dev/rmt	raw テープデバイス
/dev/sad	STREAMS 管理ドライバのエントリポイント

表 E-1 / ファイルシステムのディレクトリ 続く

ディレクトリ	説明
/dev/term	端末デバイス
/etc	ホスト固有のシステム管理構成ファイルとデータベース
/etc/acct	システム構成情報のアカウント
/etc/cron.d	cron の構成情報と FIFO
/etc/default	各種プログラムのデフォルト情報
/etc/dfs	エクスポートするファイルシステムの構成情報
/etc/fs	/usr をマウントする前に必要な処理のために、ファイルシステム形式により分類したバイナリファイル
/etc/inet	インターネットサービスの構成ファイル
/etc/init.d	各実行レベル間の移行スクリプト
/etc/lib	ブートに必要な共用ライブラリ
/etc/lp	プリンタサブシステムの構成情報
/etc/mail	メールサブシステムの構成
/etc/net	ti (トランスポート独立) ネットワークサービスの構成情報
/etc/opt	オプションパッケージの構成情報
/etc/rc0.d	実行レベル 0 の開始または終了スクリプト
/etc/rc1.d	実行レベル 1 の開始または終了スクリプト
/etc/rc2.d	実行レベル 2 の開始または終了スクリプト
/etc/rc3.d	実行レベル 3 の開始または終了スクリプト
/etc/rcS.d	実行レベル S の開始または終了スクリプト

表 E-1 / ファイルシステムのディレクトリ 続く

ディレクトリ	説明
/etc/saf	Service Access Facility (SAF) ファイル (FIFO を含む)
/etc/skel	新規ユーザアカウント用のデフォルトのプロファイルスクリプト
/etc/sm	状態監視情報
/etc/sm.bak	状態監視情報のバックアップコピー
/etc/tm	ブート時に表示される商標ファイル
/etc/uucp	uucp の構成情報
/export	エクスポートされるファイルシステムツリーのデフォルトのルート
/home	ユーザディレクトリ用サブツリーのデフォルトのルート
/kernel	ロード可能なカーネルモジュールのサブツリー (/kernel/unix などのベースカーネル自身も含む)
/mnt	ファイルシステムの一時マウントポイント
/opt	アドオンアプリケーションパッケージ用サブツリーのルート
/opt/SUNWspro	アンバンドル言語製品のマウントポイントとインストールポイント
/platform	ロード可能なカーネルモジュールのサブツリー
/sbin	ブートプロセスと手作業のシステム障害が回復するのに必須の実行可能ファイル
/tmp	一時ファイル。ブート時に消去される
/usr	/usr ファイルシステムのマウントポイント
/var	さまざまなファイル用サブツリーのルート
/var/adm	システムのログファイルとアカウント用ファイル
/var/crash	カーネルクラッシュダンプ用のデフォルト格納位置

表 E-1 / ファイルシステムのディレクトリ 続く

ディレクトリ	説明
/var/cron	cron 用ログファイル
/var/lp	ラインプリンタサブシステムのログ情報
/var/mail	ユーザメール保存用ディレクトリ
/var/news	コミュニティサービスメッセージ (USENET 形式の news と区別する)
/var/nis	NIS+ データベース
/var/opt	オプションのソフトウェアパッケージに関連するさまざまなファイル用サブツリーのルート
/var/options	SunOS 5.0 以前のパッケージとの互換性維持オプション
/var/preserve	vi と ex エディタ用のバックアップファイル
/var/sadm	ソフトウェアパッケージ管理ユーティリティが保守するデータベース
/var/saf	System Access Facility (SAF) のログファイルとアカウントファイル
/var/spool	スプールされた一時ファイル用のディレクトリ
/var/spool/cron	cron と at
/var/spool/locks	ロックファイルのスプーリング
/var/spool/lp	ラインプリンタ用スプールファイル
/var/spool/mqueue	送信用メール待ち行列
/var/spool/pkg	スプールされるパッケージ
/var/spool/uucp	待ち行列に登録された uucp ジョブ
/var/spool/uucppublic	uucp により配達されたファイル
/var/tmp	ブート処理中に消去されない一時ファイル用のディレクトリ

表 E-1 / ファイルシステムのディレクトリ 続く

ディレクトリ	説明
/var/uucp	uucp 用のログファイルと状態ファイル
/var/yp	yp 用データベース (NIS と ypbind との下位互換用)

/usr ファイルシステムの配置

表 E-2 に、/usr ファイルシステムの配置を示します。/usr ファイルシステムには、アーキテクチャ依存型と、アーキテクチャ独立型の共用ファイルが格納されます。

表 E-2 /usr ファイルシステム内のディレクトリ

ディレクトリ	説明
/usr/4lib	バイナリ互換 a.out パッケージ (BCP) 用ライブラリ
/usr/bin	標準システムコマンドの格納位置
/usr/bin/sunview1	SunView 実行可能ファイル、BCP の一部
/usr/ccs	C コンパイルシステム
/usr/ccs/bin	バイナリファイル
/usr/ccs/lib	ライブラリと補助ファイル
/usr/demo	デモ用プログラムとデータ
/usr/games	ゲームのバイナリとデータ
/usr/include	インクルードヘッダファイル (C プログラム用など)
/usr/kernel	追加モジュール
/usr/kvm	アーキテクチャ固有のバイナリとライブラリ

表 E-2 /usr ファイルシステム内のディレクトリ 続く

ディレクトリ	説明
/usr/lib	プログラムライブラリ、アーキテクチャ依存のデータベース、ユーザが直接呼び出すことのないバイナリファイル
/usr/lib/acct	アカウント用スクリプトとバイナリ
/usr/lib/dict	spell コマンド用データベースファイル
/usr/lib/class	priocntl と dispadm コマンドの実行可能ファイルを含むクラス固有のスケジュール用ディレクトリ
/usr/lib/font	troff 用のフォント記述ファイル
/usr/lib/fs	ファイルシステム形式依存型モジュール (ユーザが直接呼び出すことはない)
/usr/lib/iconv	iconv 用変換テーブル
/usr/lib/libp	プロファイルライブラリ
/usr/lib/locale	国際化および言語対応データベース
/usr/lib/localedef	localedef 用のロケールソースファイル
/usr/lib/lp	ラインプリンタサブシステムデータベースとバックエンドの実行可能ファイル
/usr/lib/mail	メールサブシステム用補助プログラム
/usr/lib/netsvc	インターネットネットワークサービス
/usr/lib/nfs	NFS 関連の補助プログラムとデーモン
/usr/lib/pics	実行時リンカの構築に必要な PIC アーカイブ
/usr/lib/refer	nroff/troff 用のプリプロセッサ
/usr/lib/sa	システム動作レポートパッケージ用のスクリプトとコマンド
/usr/lib/saf	Service Access Facility (SAF) 関連の補助プログラムとデーモン

表 E-2 /usr ファイルシステム内のディレクトリ 続く

ディレクトリ	説明
/usr/lib/spell	spell 関連の補助プログラムとデータベース
/usr/lib/uucp	uucp 関連の補助プログラムとデーモン
/usr/local	サイト固有のコマンド
/usr/net/servers	接続相手関連の外部ネームサービス要求用エントリポイント
/usr/oasys	オプションの FACE パッケージに属するファイル
/usr/old	段階的に削除されるプログラム
/usr/openwin	OpenWindows ソフトウェア用のマウントまたはインストールポイント
/usr/sadm	システム管理関連のファイルとディレクトリ
/usr/sadm/bin	FMLI スクリプトが使用するバイナリ
/usr/sadm/install	パッケージ管理用の実行可能ファイルとスクリプト
/usr/sbin	システム管理用の実行可能ファイル
/usr/sbin/static	/usr/bin と /usr/sbin にある特定のプログラムを静的にリンクしたもの。動的リンクが破壊されたときの回復に使用する。
/usr/share	アーキテクチャ独立型データベース
/usr/share/lib	アーキテクチャ独立型データベース
/usr/share/lib/keytables	キーボード配置の記述テーブル
/usr/share/lib/mailx	mailx のヘルプファイル
/usr/share/lib/nroff	nroff 用端末テーブル
/usr/share/lib/pub	さまざまなデータファイル
/usr/share/lib/spell	spell 関連の補助データベースとスクリプト

表 E-2 /usr ファイルシステム内のディレクトリ 続く

ディレクトリ	説明
/usr/share/lib/tabset	タブ設定用エスケープシーケンス
/usr/share/lib/terminfo	端末記述ファイル
/usr/share/lib/tmac	nroff と troff 用のマクロパッケージ
/usr/share/lib/zoneinfo	時間帯情報
/usr/share/src	カーネル、ライブラリ、ユーティリティのソースコード
/usr/snadm	管理ツール (admintool) 関連ファイル
/usr/ucb	Berkeley 互換パッケージバイナリ
/usr/ucbinclude	Berkeley 互換パッケージヘッダファイル
/usr/ucblib	Berkeley 互換パッケージライブラリ
/usr/vmsys	オプションの FACE パッケージに属するファイル

基本的な変更についてのクイックリファレンス

この付録は、よく使用されるコマンド、ファイルとディレクトリ、デーモンと標準プロセスの変更点のクイックリファレンスです。

変更一覧表

表 F-1 基本コマンド

SunOS 4.x	Solaris 7	説明
<code>lpr</code>	<code>lp</code>	基本的なデフォルトの印刷コマンド
<code>lpr -P printer</code>	<code>lp -d printer</code>	印刷コマンドでプリンタを指定する
<code>lpq</code>	<code>lpstat -o</code>	デフォルトのプリンタの印刷待ち行列をチェックする
<code>lpq -P printer</code>	<code>lpstat -o printer</code>	特定のプリンタの状態をチェックし印刷 ID を表示する
	<code>lpstat -a</code>	使用可能なプリンタを確認する (SunOS 4.x では <code>/etc/printcap</code> ファイルをチェックしていた)
<code>lprm print job#</code>	<code>cancel request ID</code>	印刷ジョブをキャンセルする
	<code>cancel printer</code>	現在アクティブな印刷ジョブをキャンセルする別の方法

表 F-1 基本コマンド 続く

SunOS 4.x	Solaris 7	説明
ps -ax	ps -ef	プロセス状態表示コマンドは同じだが、いくつかのオプションが変更された
pstat -s	swap -s	スワップ領域関連情報を表示する

表 F-2 高度なコマンド

SunOS 4.x	Solaris 7	説明
dump	ufsdump	ファイルシステムまたは指定されたファイルのバックアップをとる
exportfs	share <i>resources</i>	/etc/dfs/dfstab 内に登録された特定の資源のリモートマウントを可能にする
exportfs -a	shareall	/etc/dfs/dfstab 内に登録されたすべての資源のマウントを可能にするオプション
exportfs -u	unshare <i>resource</i>	資源を利用できないにする
mount -a	mountall	mountall オプションが設定された /etc/vfstab 内のすべてのファイルシステムをマウントする
restore	ufsrestore	バックアップ媒体にダンプしたファイルを復元する
showmount -d	dfmounts <i>option</i>	マウントされた NFS ファイルシステムを一覧表示する。オプションにマシン名を指定する
showmount -e	dfshares <i>option</i>	共用 (エクスポートされた) NFS ファイルシステムを一覧表示する
umount -a	umountall	/etc/vfstab 内のすべてのファイルシステム (ルート、/proc、/var、/usr を除く) のマウントを解除する

表 F-3 ファイルとディレクトリ

SunOS 4.x	Solaris 7	説明
/var/spool/mail	/var/mail	受信 mail の格納位置
/etc/fstab	/etc/vfstab	ファイルシステムのマウントテーブル
/etc/exports	/etc/dfs/dfstab	エクスポートされたファイルシステムをリスト表示する
/etc/mstab	/etc/mnttab	/etc/mount コマンドが読み取る現在マウントされている資源のリスト
/etc/xtab	/etc/dfs/sharetab	共用可能な資源のリスト
/usr/bin	/usr/bin および /usr/sbin	/usr/sbin は Solaris の実行可能ファイルで利用できる
/etc/aliases	/etc/mail/aliases	ローカルな電子 mail 別名ファイルの新しい格納位置
/etc/printcap	削除された	/usr/share/lib/terminfo と /etc/lp 内のファイルで置換された機能
/etc/passwd	/etc/passwd /etc/shadow	/etc/passwd ファイルと /etc/shadow ファイル (ユーザの暗号化されたパスワードとその他の情報を格納するファイル) は機能を共有する。

表 F-4 デーモンと標準プロセス

SunOS 4.x	Solaris 7	説明
/usr/lib/lpd	/usr/lib/lp/lpsched	印刷デーモン
/usr/etc/rpc.lockd	/usr/lib/nfs/lockd	ネットワークロックデーモン
/usr/etc/rpc.mountd	/usr/lib/nfs/mountd	NFS マウント要求サーバ
/usr/etc/ypbind	/usr/lib/netsvc/yp/ypbind	NIS バインダプロセス
/usr/etc/nfsd	/usr/lib/nfs/nfsd	NFS デーモン

表 F-4 デーモンと標準プロセス 続く

SunOS 4.x	Solaris 7	説明
/usr/etc/biod	削除された	ブロック I/O デーモン (カーネル内部に実装された)
/etc/rc と /etc/ rc.local	/etc/rc[012356S].d	システム初期設定スクリプト

表 F-5 ファイルおよびコマンドの相違

SunOS 4.x	Solaris 7
ac	sar
add_services	pkgadd
arch	uname -m
bar	ファイルの取り出しには cpio -H を使用
biff -n	chmod -o-x /dev/tty
biff -y	chmod -o+x /dev/tty
cc	利用できない
dbxtool	debugger
df	df -k
dketl	利用できない
dkinfo	prtvto
du	du -k
dump	ufsdump

表 F-5 ファイルおよびコマンドの相違 続く

SunOS 4.x	Solaris 7
dumpfs	利用できない
etherfind	snoop
exportfs	share
extract_files	利用できない
extract_patch	利用できない
extract_unbundled	pkgadd
fastboot	reboot または init -6
fasthalt	init -0
hostid	sysdef -h
hostname	uname -n
intr	利用できない
leave	cron と at を使用
lint	利用できない
load	pkgadd
loadc	pkgadd
load_package	利用できない
lpc	lpadmin
lpd	lpsched
lpq	lpstat

表 F-5 ファイルおよびコマンドの相違 続く

SunOS 4.x	Solaris 7
lpr	lp
lprm	cancel
lptest	利用できない
mach	uname -p
modstat	mount -a
mount	mount -F <i>fstype</i> [<i>options</i>]
mountall	modinfo
mount_tfs	mount -F <i>fstype</i>
pax	cpio
paxcpio	cpio
portmap	rpcbind
printenv	env
ps -a	ps -e
ps -aux	ps -el
pstat	sar
pstat -s	swap -s
rdump	ufsdump
restore	ufsrestore
rm_client	admintool

表 F-5 ファイルおよびコマンドの相違 続く

SunOS 4.x	Solaris 7
rm_services	利用できない
rpc.etherd	利用できない
rpc.lockd	lockd
rpc.mountd	mountd
rpc.read	利用できない
rpc.rquotad	利用できない
rpc.showfhd	showfhd
rpc.statd	statd
rpc.user_agentd	利用できない
rpc.yppasswdd	利用できない
rpc.yppupdated	ypupdated
rrestore	ufsrestore
rusage	利用できない
startup	利用できない
swapon	swap -a
sys-config	admintool
umountall	umount -a
umount-tfs	umount -F <i>fstype</i>
unload	pkgrm

表 F-5 ファイルおよびコマンドの相違 続く

SunOS 4.x	Solaris 7
update	fsflush
uptime	who -b
users	who -q
vipw	利用できない
wall	利用できない
whereis	利用できない
whoami	id
ypbatchupd	利用できない
yppasswd	NIS+ の nispasswd を使用
ypserv	利用できない

用語集

DDI	デバイスドライバインタフェース。DDI を使用すれば、特定のシステム上でオペレーティングシステムの連続するリリース間で、ソースとバイナリの両方を容易に移植できる。
DKI	ドライバカーネルインタフェース。ドライバとカーネルとの間の通信に指定されるエントリポイントルーチンとユーティリティ機能が使用するサービスインタフェース。ドライバとハードウェアとの間、またはドライバとブートソフトウェアとの間のインタフェースは含まれない。
DNS	ドメインネームシステム。インターネットで使用されている分散ネーム/ドレスメカニズム。
ELF	実行可能なリンクフォーマット。Solaris 7 実行可能ファイルのオブジェクト形式。
IP アドレス	ネットワーク上の各ホストを識別する一意な番号。ネットワークアドレスとホストアドレスという 2 つの異なるアドレスからなる。
NIS	ネットワーク情報サービス。LAN 内のマシンとサービスに関する情報を提供する。
NIS+	ネットワーク情報サービスプラス。安全なアップデート、より高い性能、階層ネームサービスを実現する。
OLIT	OPEN LOOK Intrinsic Toolkit の略。

SAC	Service Access Control。モデムや端末など、ローカルとネットワーク上のシステムサービスへのアクセスを管理する SunOS 5.6 のツール
SAF	Service Access Facility。サービスの設定と管理に使用するコマンド。
アーキテクチャ	コンピュータシステムを構成する各要素とそれらが相互に動作する方法。Solaris 7 のカーネルレベルからみた場合、「アーキテクチャ」とはシステム内の CPU チップの種類を意味する。このマニュアルのアーキテクチャの説明は、カーネルアーキテクチャ (たとえば、sun4、sun4c、sun4m) だけを対象にしている。
アンバンドル製品	SunOS 5.6 ソフトウェア配布に含まれない製品。たとえば、SunPro™ のコンパイラなど。
異機種サーバ	ディスクレスクライアントのサーバで、自分と同じカーネルアーキテクチャと、自分とは異なるアーキテクチャのクライアントを持つもの。
インストールサーバ	Solaris 7 の配布ソフトウェアにブートサービスとネットワークアクセスを提供するマシン。ローカルの CD-ROM 上、または、配布ソフトウェアのコピーを含むファイルシステムのどちらかにある。
カーネルアーキテクチャ	Solaris 7 カーネルのハードウェア部分。同じ Solaris 7 のカーネルが稼働していれば、その 2 つのシステムは同じカーネルアーキテクチャを持っている。すべての Sun-4 システムが同じカーネルアーキテクチャを持っているとはかぎらない。
クライアント	他のシステムが提供する NIS、NFS、その他のサービスを使用するシステム。
クラスタ	ソフトウェアパッケージの機能的集合体。
構成クラスタ	典型的なソフトウェアのグループを表すクラスタのデフォルトの選択。
サーバ	ネットワークにサービスを提供するシステム。NFS サーバや NIS データベースサーバなどがある。

時間帯	縦に 24 分割された地球の表面の各部分に設定された標準時間。
スタンドアロン	root、swap、/usr の各ディスクパーティションがすべてローカルのディスク上にある、サーバに依存しないシステム。
ソース互換パッケージ	SunOS 4.x と BSD のコマンド、ライブラリルーチン、ヘッダファイルなど、Solaris 7 では使用できない部分を含むオプションのパッケージ。
データレス	/usr と /usr/kvm ファイルシステムがファイルサーバ上にあり、ルートとスワップ用のディスクパーティションだけがローカルのディスク上にあるシステム。
ディスクスライス	分割されたディスクの各部分。インストール時に構成される。SunOS 4.1.x と Sytem V リリース 3 におけるパーティションと同義。
ディスクパーティション	「ディスクスライス」を参照。
ディスクレス	/(ルート)、swap、/usr の各ファイルシステム (ディスクパーティション) がローカルのディスク上ではなく、NFS サーバ (または、ファイルサーバ) 上にあるシステム。
同機種サーバ	ディスクレスクライアントのサーバで、自分と同じカーネルアーキテクチャのクライアントだけを持つもの。
ネットマスク	サブネット情報を IP アドレスのホスト部分から分離するために使用される数字。サブネットマスクと呼ぶこともある。
バイナリ互換モード	既存の SunOS 4.x アプリケーション (静的にリンクされたものと動的にリンクされたもの) を、修正したり再コンパイルしたりせずに SunOS 5.6 の制御下で実行できるようにするオプションのパッケージ。
パッケージ	ソフトウェアを機能グループにわけたもの。SunOS 5.6 のすべてのソフトウェアは、パッケージにグループ化され配布される。パッケージは、Sun とサードパーティ製のアンバンドルソフトウェアを配布する標準的な方法でもある。

マルチ OS オペレーション

SPARC サーバが、SunOS 4.1.x クライアントにサービスを提供しながら Solaris 7 も稼働できるようにするオペレーション。このオペレーションでは、異機種サーバが同じカーネルアーキテクチャを持つクライアントにサービスを提供することもある。

索引

数字

/4lib ディレクトリ, 381
/5bin ディレクトリ, 96
/5include ディレクトリ, 96
/5lib ディレクトリ, 96, 169 - 171
32 ビットの実行環境, 10
64 ビットの実行環境, 10
64 ビットライブラリ, 152

A

a.out システムファイル, 97, 372
a641 ライブラリルーチン, 283
abort ライブラリルーチン, 283
abs ライブラリルーチン, 283
accept システムコール, 253
access システムコール, 253
accounting システム
 構成情報ディレクトリ, 378
accountingファイル
 Service Access Facility (SAF), 380
 システム, 379
 スクリプトとバイナリ, 382
acctcms コマンド, 198
acctcom コマンド, 198
acctcon1 コマンド, 198
acctcon2 コマンド, 198
acctdisk コマンド, 198
acctdusg コマンド, 198, 209
acctmerg コマンド, 198
accton コマンド, 198
acctprc1 コマンド, 198
acctprc2 コマンド, 198

acctwtmpt コマンド, 198
acct コマンド, 236
acct システムコール, 253
acct システムファイル, 372
/acct ディレクトリ, 378, 382
acosh ライブラリルーチン, 283
acos ライブラリルーチン, 283
ac コマンド, 198
adbgen コマンド, 199
adb コマンド, xv, 160, 162, 198, 199
addbib コマンド, 199
addch ライブラリルーチン, 283
addexportent ライブラリルーチン, 283
addmntent ライブラリルーチン, 283
addstr ライブラリルーチン, 283
add_client コマンド, 199
add_drv コマンド, 74, 191, 192
add_services コマンド, 199
adjacentscreens コマンド, 199
adjtime システムコール, 253
Administration ツール
 files ディレクトリ, 384
admintool
 Serial Port Manager, 125
Admintool
 描写, 9, 24
admintool
 ユーザとグループの管理, 55
Admintool
 ユーザアカウント, 56
admintool コマンド, xv
Admintool コマンド, 24

admintool コマンド
 add_client command vs., 199
 files ディレクトリ, 384
 rm_client command vs., 233
 admin コマンド, 176, 199
 /.Admin ディレクトリ, 134
 /adm ディレクトリ, 379
 adv コマンド, 199
 aedplot コマンド, 199
 aging パスワード, 47
 agt_create ライブラリルーチン, 283
 agt_enumerate ライブラリルーチン, 283
 agt_trap ライブラリルーチン, 284
 aint ライブラリルーチン, 284
 aiocancel ライブラリルーチン, 284
 aioread ライブラリルーチン, 284
 aiowait ライブラリルーチン, 284
 aiowrite ライブラリルーチン, 284
 alarm ライブラリルーチン, 284
 aliasadm コマンド, 23
 aliases システムファイル, 372, 387
 align_equals コマンド, 199
 alloca ライブラリルーチン, 284
 alphasort ライブラリルーチン, 284
 analyze コマンド, 199
 anint ライブラリルーチン, 284
 annuity ライブラリルーチン, 284
 ANSI C コンパイラ, 151
 AnswerBook2, 9
 apropos コマンド, 199
 architecture-specific information, directories
 for, 381
 arch コマンド, 200
 arc ライブラリルーチン, 284
 arp コマンド, 200
 ar コマンド, 176, 199, 231
 ar システムファイル, 372
 asctime ライブラリルーチン, 284
 ASET (自動セキュリティ拡張ツール), 9, 49
 asinh ライブラリルーチン, 284
 asin ライブラリルーチン, 284
 assert ライブラリルーチン, 285
 async_daemon システムコール, 253
 as コマンド, 176, 200
 AT&T システム
 SVR4 features excluded from Solaris 動作
 環境, 11
 sysadm メニューユーティリティ, 11
 デバイスの命名, 17
 atan2 ライブラリルーチン, 285
 atanh ライブラリルーチン, 285
 atan ライブラリルーチン, 285
 atexit ライブラリルーチン, 328
 atof ライブラリルーチン, 285
 atoi ライブラリルーチン, 285
 atol ライブラリルーチン, 285
 atoplot コマンド, 200
 atq コマンド, 200
 atrm コマンド, 200
 attroff ライブラリルーチン, 285
 attron ライブラリルーチン, 285
 attrset ライブラリルーチン, 285
 at コマンド, 200, 221, 380
 audioio ioctl, 166
 audit.log システムファイル, 372
 auditd コマンド, 201
 auditon システムコール, 253
 auditsvc system call, 253
 audit_args ライブラリルーチン, 285
 audit_control システムファイル, 372
 audit_data システムファイル, 372
 audit_text ライブラリルーチン, 285
 audit_warn コマンド(8), 201
 audit コマンド, 201
 audit システムコール, 253
 authdes_create ライブラリルーチン, 285
 authdes_getucred ライブラリルーチン, 286
 authdes_seccreate ライブラリルーチン, 285
 authnone_create ライブラリルーチン, 286
 authsys_create_default ライブラリルーチ
 ン, 286
 authsys_seccreate ライブラリルーチン, 286
 authunix_create_default ライブラリルーチ
 ン, 286
 authunix_create ライブラリルーチン, 286
 auth_destroy ライブラリルーチン, 286
 auto.home システムファイル, 201, 372
 auto.master システムファイル, 201, 372
 AutoFS, 22
 autofs プログラム, 21, 99, 100
 automounting, 201

automount コマンド, 201
auto_home システムファイル, 201
auto_master システムファイル, 201
awk コマンド, 201
a自動セキュリティ拡張ツール(ASET), 49

B

Backup CoPilot, 31, 105
banner コマンド, 201
bar コマンド, 105, 201
bar システムファイル, 372
basename コマンド, 201
batch コマンド, 201
baudrate ライブラリルーチン, 286
bcmp ライブラリルーチン, 286
bcopy ライブラリルーチン, 286
bc コマンド, 202
beep ライブラリルーチン, 286
Berkeley 互換パッケージ, 384
BFS (ブートファイルシステム), 86
bgplot コマンド, 202
biff コマンド, 202
bin-mail コマンド, 202
binaries, directories for, 378, 381
bindresvport ライブラリルーチン, 286
bindtextdomain ライブラリルーチン, 184
bind システムコール, 253
biod コマンド, 202, 388
block I/O デモン, 388
boards.pc システムファイル, 372
boot.sun4c.sunos.4.1 コマンド, 74, 190
bootblk コマンド, 190
bootparamd コマンド, 202
bootparams データベース, 372
bootparam ライブラリルーチン, 286
bootsd コマンド, 74
boot コマンド
 changes in, 202
 再構成ブート, 191
 デバイスの管理, 68
 デバイスの追加と, 72
 デバイスを追加, 67
 の変更, 72, 73, 189, 190
Bourne シェル, 51 - 53, 238
 restricted, 235
 制限付きシェル, 47

box ライブラリルーチン, 287
brk システムコール, 253
BSD ソース互換パッケージ, 41, 42
bsearch ライブラリルーチン, 287
byteorder ライブラリルーチン, 287
bzero ライブラリルーチン, 287

C

C2conv コマンド, 202
C2unconv コマンド, 202
CACHEFS (キャッシュファイルシステム), 81, 85
calendar コマンド, 202
calloc ライブラリルーチン, 288
callrpc ライブラリルーチン, 288
cal コマンド, 202
cancel コマンド, 26, 123, 202, 222, 385
capitalize コマンド, 203
captaininfo コマンド, 203
catclose ライブラリルーチン, 288
catgetmsg ライブラリルーチン, 185, 288
catgets ライブラリルーチン, 184, 288
catman コマンド, 203
catopen ライブラリルーチン, 288
cat コマンド, 203
cbc_crypt ライブラリルーチン, 288
cbreak ライブラリルーチン, 288, 291
cbrt ライブラリルーチン, 288
cb コマンド, 179, 203
cc コマンド, 179, 203
CD-ROM デバイス
 install4x プログラムと, 113, 115
 Solaris 2.6 要求, 30
 管理, 68, 70
 性能の改善, 80, 85
 マネージャ, 28
 命名規則, 65
CD-ROM ファイルシステム (HSFS), 68, 84, 101
cdc コマンド, 176, 203
/cdrom ディレクトリ, 69, 101
cd コマンド, 203
ceil ライブラリルーチン, 288
cfgetispeed ライブラリルーチン, 288
cfgetospeed ライブラリルーチン, 288
cflow コマンド, 179, 203

cfree ライブラリルーチン, 288
 cfsetispeed ライブラリルーチン, 289
 cfsetospeed ライブラリルーチン, 289
 chargefee コマンド, 203
 chdir システムコール, 254
 checkeq コマンド, 203
 checking, xv
 checknr コマンド, 203
 CHECK ライブラリルーチン, 288
 chfn コマンド, 204
 chgrp コマンド, 204
 chkey コマンド, 204
 chmod system call, 254
 chmod コマンド, 202, 204
 chown コマンド, 204
 chown システムコール, 254
 chroot システムコール, 254
 chroot ライブラリルーチン, 184, 204
 chrtbl コマンド, 204
 chsh コマンド, 204
 circle ライブラリルーチン, 289
 ckpacct コマンド, 204
 /class ディレクトリ, 382
 clearerr ライブラリルーチン, 289
 clearok ライブラリルーチン, 289
 clear_colormap コマンド, 204
 clear_function コマンド, 205
 clear コマンド, 204
 clear ライブラリルーチン, 289
 click コマンド, 205
 clntraw_create ライブラリルーチン, 290
 clnttcp_create ライブラリルーチン, 290
 clntudp_bufcreate ライブラリルーチン, 290
 clntudp_create ライブラリルーチン, 290
 clnt_broadcast ライブラリルーチン, 289
 clnt_call ライブラリルーチン, 289
 clnt_control ライブラリルーチン, 289
 clnt_create ライブラリルーチン, 289, 290
 clnt_destroy ライブラリルーチン, 289
 clnt_dg_create ライブラリルーチン, 290
 clnt_freeres ライブラリルーチン, 289
 clnt_geterr ライブラリルーチン, 289
 clnt_pcreateerror ライブラリルーチン, 289
 clnt_perrno ライブラリルーチン, 289
 clnt_perror ライブラリルーチン, 289
 clnt_raw_create ライブラリルーチン, 290
 clnt_spccreateerror ライブラリルーチン, 289
 clnt_sperrno ライブラリルーチン, 289
 clnt_sperror ライブラリルーチン, 289
 clnt_tli_create ライブラリルーチン, 290
 clnt_vc_create ライブラリルーチン, 290
 clock コマンド, 205
 clock ライブラリルーチン, 290
 closedir ライブラリルーチン, 290
 closelog ライブラリルーチン, 290
 closepl ライブラリルーチン, 290
 close システムコール, 254
 clri コマンド, 86, 205
 clrtobot ライブラリルーチン, 290
 clrtoeol ライブラリルーチン, 290
 cmdtool コマンド, 205
 cmp コマンド, 205
 COFF, 178, 179
 colcrt コマンド, 205
 colldef コマンド, 205
 colltbl コマンド, 205
 coloredit コマンド, 205
 colrm コマンド, 205
 col コマンド, 205
 comb コマンド, 176, 205
 comm コマンド, 205
 compound ライブラリルーチン, 291
 compress コマンド, 205
 config コマンド, 74, 188, 191, 205
 Config ファイル, 134
 connect system call, 254
 cont ライブラリルーチン, 291
 convert4x プログラム, 113, 118
 copyright ファイル, 174
 copysign ライブラリルーチン, 291
 copywin ライブラリルーチン, 291
 cosh ライブラリルーチン, 291
 cos ライブラリルーチン, 291
 cpio コマンド, 107
 bar command vs., 201
 changes in, 206
 pax command vs., 229
 paxcpio command vs., 229
 説明, 106, 107, 109
 のサポート, 105
 の変更, 108, 109
 cpio システムファイル, 372
 cpp コマンド, 176, 206

cp コマンド, 206
 crash コマンド, 206
 /crash ディレクトリ, 379
 creat システムコール, 255
 crmode ライブラリルーチン, 291
 /cron.d ディレクトリ, 378
 crontab コマンド, 206
 crontab システムファイル, 372
 cron コマンド, 206, 221, 380
 /cron ディレクトリ, 380
 crtplot コマンド, 206
 crypt コマンド, 206
 crypt ライブラリルーチン, 288, 292
 _crypt ライブラリルーチン, 288
 cscope コマンド, 179
 .cshrc ファイル, 53
 csh コマンド, 51 - 53, 206
 csplit コマンド, 206
 ctags コマンド, 206
 ctermid ライブラリルーチン, 292
 ctime ライブラリルーチン, 293
 ctrace コマンド, 179, 206
 curs_set ライブラリルーチン, 293
 cuserid ライブラリルーチン, 293
 cut コマンド, 207
 cu コマンド, 206
 cv_broadcast ライブラリルーチン, 293
 cv_create ライブラリルーチン, 293
 cv_destroy ライブラリルーチン, 293
 cv_enumerate ライブラリルーチン, 293
 cv_notify ライブラリルーチン, 294
 cv_send ライブラリルーチン, 294
 cv_waiters ライブラリルーチン, 294
 cv_wait ライブラリルーチン, 294
 cxref コマンド, 179, 207
 C コンパイラ, 151, 179
 C コンパイルシステム, 381
 C コンパイルシステム, ディレクトリ, 96
 C シェル, xv, 51 - 53, 206
 C プログラムツール, 179

D

Data Link Provider Interface (DLPI), 182
 date コマンド, 184, 207
 dbconfig コマンド, 207
 dbmclose ライブラリルーチン, 294

dbmopen ライブラリルーチン, 294
 dbm_clearerr ライブラリルーチン, 294
 dbm_close ライブラリルーチン, 294
 dbm_delete ライブラリルーチン, 294
 dbm_error ライブラリルーチン, 294
 dbm_fetch ライブラリルーチン, 294, 301
 dbm_firstkey ライブラリルーチン, 294, 301
 dbm_nextkey ライブラリルーチン, 294, 326
 dbm_open ライブラリルーチン, 294
 dbm_store ライブラリルーチン, 294, 345
 dbxtool コマンド, xv, 160, 179, 207
 dbx コマンド, 160, 179, 207
 -dcheck コマンド, 207
 dc コマンド, 207
 DDI/DKI (デバイスドライバインタフェース/ドライバカーネルインタフェース), 10
 ddi_create_minor_node コマンド, 192
 DDI (デバイスドライバインタフェース), 10
 dd コマンド, xv, 104, 106, 207
 decimal_to_double ライブラリルーチン, 294
 decimal_to_extended ライブラリルーチン, 294
 decimal_to_floating ライブラリルーチン, 294
 decimal_to_single ライブラリルーチン, 294
 defaultsedit コマンド, 208
 defaults_from_input コマンド, 207
 defaults_merge コマンド, 207
 defaults_to_indentpro コマンド, 208
 defaults_to_mailrc コマンド, 208
 def_prog_mode ライブラリルーチン, 295, 337
 def_shell_mode ライブラリルーチン, 295
 delay_output ライブラリルーチン, 295
 delch ライブラリルーチン, 295
 deleteln ライブラリルーチン, 295
 delete ライブラリルーチン, 295
 delta コマンド, 176, 208
 delwin ライブラリルーチン, 295
 del_curterm ライブラリルーチン, 295
 /demo ディレクトリ, 381
 deroff コマンド, 208
 DeskSet, 4, 7
 des_crypt ライブラリルーチン, 295
 des_setparity ライブラリルーチン, 295
 des コマンド, 208
 /dev/dsk ディレクトリ, 90, 193, 377
 /dev/ksyms file, 162

/dev/pts ディレクトリ, 90, 377
/dev/rdisk ディレクトリ, 90, 193, 377
/dev/rmt ディレクトリ, 90, 377
/dev/sad ディレクトリ, 90, 377
/dev/sd1g コマンド, 74
/dev/term ディレクトリ, 90, 378
/devices ディレクトリ, 192
devinfo コマンド, 66, 67, 208
devnm コマンド, 208
/dev ディレクトリ
 changes in, 74, 377
 /devices directory and, 192
 reconfiguration boot and, 191
 記述, 193, 377
 の変更, 90
 変更, 18, 63
dfmounts コマンド, 100, 386
dfshares コマンド, 100, 386
DFS (分散ファイルシステム) の管理, 130
df コマンド, 66, 86, 101, 208
dgettext ライブラリルーチン, 184
/dict ディレクトリ, 382
diff3 コマンド, 209
diffmk コマンド, 209
diff コマンド, 209
dircmp コマンド, 209
directories
 クイックリファレンス, 387, 388
dirname コマンド, 209
dir システムファイル, 372
discover4x プログラム, 112, 113
diskusg コマンド, 209
dispadm コマンド, 382
dis コマンド, 178, 209
dkctl コマンド, 210
dkinfo コマンド, 32, 66, 210
dkio ioctl, 163, 165
DKI (ドライバカーネルインタフェース), 10
dlclose ライブラリルーチン, 295
dlerror ライブラリルーチン, 295
dlopen ライブラリルーチン, 295
DLPI (Data Link Provider Interface), 182
dlsym ライブラリルーチン, 295
dmesg コマンド, 210
dname コマンド, 210
DNS (ドメインネームシステム)

NIS+ (Network Information Services Plus)
 vs., 136
NIS+ (Network Information Services Plus)
 対, 137
 説明, 136
dn_comp ライブラリルーチン, 295
dn_expand ライブラリルーチン, 295
dodisk コマンド, 210
domainname コマンド, 210
dorfs コマンド, 210
dos2unix コマンド, 210
double_to_decimal ライブラリルーチン, 295
doupdate ライブラリルーチン, 295
draino ライブラリルーチン, 295
drand48 ライブラリルーチン, 296, 311, 333,
 340
Driver Kernel Interface (DKI), xv
drvconfig プログラム, 192
/drv ディレクトリ, 97
/dsk ディレクトリ, 90, 193, 377
dumbplot コマンド, 210
dumpadm コマンド, 211
dumpfs コマンド, 211
dumpkeys コマンド, 211
dump コマンド, xv
 changes to old, 211
 new, 178
 new 218, xv
 oldへの変更, 105
 クイックリファレンステーブル, 386
 リンクのチェックを使用する, 158, 159
dump システムファイル, 372
dup2 システムコール, 255
dup system call, 255
du コマンド, 66, 101, 210
dysize ライブラリルーチン, 296

E

ecb_crypt ライブラリルーチン, 296
echochar ライブラリルーチン, 296
echo コマンド, 212
echo ライブラリルーチン, 296
econvert ライブラリルーチン, 296
ecvt ライブラリルーチン, 296
edata ライブラリルーチン, 296
edit コマンド, 212

edquota コマンド, 212
 ed コマンド, 212
 eeprom コマンド, 212
 EFT (Extended Fundamental Types), 31
 egrep コマンド, 212
 eject コマンド, 212
 ELF (executable and linking format) files
 コンパイラ, 151
 ELF (executable and linking format) ファイル
 実行するためのカーネルモジュール, 97
 リンカ, 152
 encrypt ライブラリルーチン, 296
 endac ライブラリルーチン, 296
 endexportent ライブラリルーチン, 296
 endfsent ライブラリルーチン, 297
 endgraent ライブラリルーチン, 297
 endgrent ライブラリルーチン, 297
 endhostent ライブラリルーチン, 297
 endmntent ライブラリルーチン, 297
 endnetent ライブラリルーチン, 297
 endnetgrent ライブラリルーチン, 297
 endprotoent ライブラリルーチン, 297
 endpwaent ライブラリルーチン, 297
 endpwent ライブラリルーチン, 297
 endrpcent ライブラリルーチン, 297
 endservent ライブラリルーチン, 297
 endttyent ライブラリルーチン, 297
 endusershell ライブラリルーチン, 297
 endwin ライブラリルーチン, 297
 end ライブラリルーチン, 296
 enroll コマンド, 212
 environ システムファイル, 372
 env コマンド, 212, 230
 eqn コマンド, 212
 erand48 ライブラリルーチン, 297
 erasechar ライブラリルーチン, 298
 erase ライブラリルーチン, 298
 erfc ライブラリルーチン, 298
 erf ライブラリルーチン, 298
 errno 値, 252
 errno ライブラリルーチン, 298
 error コマンド, 176, 212
 /etc/.login ファイル, 53
 /etc/acct ディレクトリ, 378
 /etc/aliases ファイル, 372, 387
 /etc/config コマンド, 74, 188, 191, 205
 /etc/cron.d ディレクトリ, 378
 /etc/default/fs ファイル, 88
 /etc/default/login ファイル, 47
 /etc/default/passwd ファイル, 47
 /etc/default/su ファイル, 47
 /etc/default ディレクトリ, 46, 91, 378
 /etc/dfs/dfstab ファイル, 39, 102
 /etc/dfs/fstype ファイル, 88
 /etc/dfs/sharetab ファイル, 283, 387
 /etc/dfs ディレクトリ, 378
 /etc/exports ファイル, 39, 102, 283, 372, 387
 /etc/fstab ファイル
 described, 373
 /etc/vfstab vs., 39, 40, 92, 387
 記述, 387
 描写, 32
 /etc/fs ディレクトリ, 89, 378
 /etc/group ファイル, 39, 373
 /etc/inet ディレクトリ, 91, 378
 /etc/init.d スクリプト, 232, 378
 /etc/inittab ファイル, 73, 75
 /etc/lib ディレクトリ, 378
 /etc/lp/printers ディレクトリ, 123
 /etc/lp ディレクトリ, 26, 80, 91, 122, 378
 /etc/mail/aliases ファイル, 387
 /etc/mail/sendmail.cf ファイル, 40
 /etc/mail ディレクトリ, 378
 /etc/mnttab ファイル, 387
 /etc/mntab ファイル, 373, 387
 /etc/netgroup ファイル, 39, 373
 /etc/net ディレクトリ, 378
 /etc/opt ディレクトリ, 91, 98, 378
 /etc/passwd ファイル, 40, 46, 374, 387
 /etc/printcap database, replacement of, 122,
 123, 374
 /etc/printcap データベース, 置換, 80
 /etc/printcap データベースの変更, 387
 /etc/printcap データベース、復元, 40
 /etc/printcap データベース、変更, 26
 /etc/profile ファイル, 53
 /etc/rc.boot スクリプト, 74, 75, 90, 190, 232
 /etc/rc.d スクリプト, 91, 190, 378, 388
 /etc/rc.local スクリプト, 74, 75, 90, 190, 232,
 388
 /etc/rc.single スクリプト, 74, 90, 190
 /etc/rcS.d スクリプト, 91, 378, 388
 /etc/rcS スクリプト, 74, 91, 190

/etc/rc スクリプト, xv, 74, 75, 90, 91, 190, 232, 378, 388
/etc/rmmount.conf ファイル, 70
/etc/saf ディレクトリ, 91, 379
/etc/sendmail.cf ファイル, 40
/etc/shadow ファイル, 40, 46, 95, 387
/etc/skel ディレクトリ, 54, 379
/etc/sm.bak ディレクトリ, 379
/etc/sm ディレクトリ, 379
/etc/system ファイル
 moddir variable, 188
 moddir 変数, 21
 記述, 74, 189, 191
 描写, 68
/etc/tm ディレクトリ, 379
/etc/ttytab ファイル, 40, 375
/etc/uucp/Config ファイル, 132, 134
/etc/uucp/Grades ファイル, 132
/etc/uucp/Limits ファイル, 132, 134
/etc/uucp ディレクトリ, 39, 132, 379
/etc/vfstab ファイル
 fields, 92, 95
 merging /etc/fstab ファイル内, 39
 記述, 387
 説明, 92
 のファイルシステムを指定する, 101
 例, 94
/etc/vold.conf ファイル, 70
/etc/xtab ファイル, 376, 387
/etc ディレクトリ, xv
 changes in, 378, 379
 記述, 378
 説明, 82, 90
 の変更, 80, 88, 90, 92
etext ライブラリルーチン, 298
etherd コマンド, 212
etherfind コマンド, 212
Ethernet ドライバ, 182
ethers システムファイル, 372
ether_aton ライブラリルーチン, 298
ether_hostton ライブラリルーチン, 298
ether_line ライブラリルーチン, 298
ether_ntoa ライブラリルーチン, 298
ether_ntohost ライブラリルーチン, 298
ether ライブラリルーチン, 298
EUC (拡張 UNIX コード), 183
exc_bound ライブラリルーチン, 298

exc_handle ライブラリルーチン, 298
exc_notify ライブラリルーチン, 298
exc_on_exit ライブラリルーチン, 298
exc_raise ライブラリルーチン, 298
exc_unhandle ライブラリルーチン, 298
exc_uniqpatt ライブラリルーチン, 298
execle ライブラリルーチン, 299
execlp ライブラリルーチン, 299
execl ライブラリルーチン, 298
execve システムコール, 256
execvp ライブラリルーチン, 299
execv ライブラリルーチン, 299
/exec ディレクトリ, 97
_exit システムコール, 255
exit ライブラリルーチン, 299
exp10 ライブラリルーチン, 300
exp2 ライブラリルーチン, 300
expand コマンド, 212
expm1 ライブラリルーチン, 300
/export/home ディレクトリ, 201
exportent ライブラリルーチン, 300
exportfs コマンド, 100, 102, 212, 386
exports ファイル, 39, 102, 283, 372, 387
/export ディレクトリ, 112, 379
expr コマンド, 213
exp ライブラリルーチン, 300
exstr コマンド, 178
Extended Fundamental Types (EFT), 31
extended_to_decimal ライブラリルーチン, 300
extract_files コマンド, 213
extract_patch コマンド, 213
extract_unbundled コマンド, 213
exエディタ用の バックアップファイル ディレ
 クトリ, 380
ex コマンド, 212
e コマンド, 212

F

fabs ライブラリルーチン, 300
FACE パッケージ, 383, 384
false コマンド, 213
fastboot コマンド, 72, 78, 213
fasthalt コマンド, 74, 76, 78, 213
fbtab システムファイル, 372
fchdir system call, 256

fchmod システムコール, 256
 fchown システムコール, 256
 fchroot system call, 256
 fclose ライブラリルーチン, 297, 300
 fcntl system call, 267
 fcntl システムコール, 256
 fcntl システムファイル, 373
 fconvert ライブラリルーチン, 300
 fevt ライブラリルーチン, 300
 fdformat コマンド, 213
 FDFS (ファイル記述子ファイルシステム), 81, 84
 fdopen ライブラリルーチン, 300
 feof ライブラリルーチン, 300
 ferror ライブラリルーチン, 301
 fetch ライブラリルーチン, 301
 fflush ライブラリルーチン, 301
 ffs ライブラリルーチン, 301
 ff コマンド, 86
 fgetc ライブラリルーチン, 301
 fgetgraent ライブラリルーチン, 301
 fgetgrent ライブラリルーチン, 301
 fgetpwaent ライブラリルーチン, 301
 fgetpwent ライブラリルーチン, 301
 fgets ライブラリルーチン, 301
 fgrep コマンド, 213
 FIFOFS (FIFO/パイプファイルシステム), 81, 84
 fileno ライブラリルーチン, 301
 file_to_decimal ライブラリルーチン, 301
 file コマンド, 213
 filio ioctl, 163, 165
 filter ライブラリルーチン, 301
 find コマンド, 102, 213
 fingerd コマンド, 213
 finger コマンド, 213
 finite ライブラリルーチン, 301
 firstkey ライブラリルーチン, 301
 fixterm ライブラリルーチン, 301
 flash ライブラリルーチン, 301
 floatingpoint ライブラリルーチン, 301
 flock システムコール, 256
 floor ライブラリルーチン, 302
 /floppy ディレクトリ, 69, 101
 flushing ライブラリルーチン, 302
 flusok ライブラリルーチン, 302
 FMLI スクリプトが使用するバイナリ, 383
 fmod ライブラリルーチン, 302
 fmt_mail コマンド, 214
 fmt コマンド, 213
 fold コマンド, 214
 fontedit コマンド, 214
 fonts
 troff用のフォント記述ファイル, 382
 /font ディレクトリ, 382
 fopen ライブラリルーチン, 302, 339, 340
 foption コマンド, 214
 fork システムコール, 257
 format コマンド, 31, 214
 fparel コマンド, 214
 fpathconf システムコール, 257
 fpaversion コマンド, 214
 fpa_download コマンド, 214
 fprintf ライブラリルーチン, 302
 fpurel コマンド, 214
 fputc ライブラリルーチン, 302
 fputs ライブラリルーチン, 302
 fpuversion コマンド, 214
 fp_class ライブラリルーチン, 302
 fread ライブラリルーチン, 302
 free ライブラリルーチン, 302
 freopen ライブラリルーチン, 302
 frexp ライブラリルーチン, 302
 from コマンド, 214
 fscanff ライブラリルーチン, 302
 fsck_cdrom コマンド, 214
 fsck コマンド, 86, 104, 207, 214
 fsdb コマンド, 86, 217
 fseek ライブラリルーチン, 302
 fsflush コマンド, 245
 fsirand コマンド, 214
 fspec システムファイル, 373
 fstatfs system call, 257
 fstatvfs system call, 257
 fstat システムコール, 257
 fstyp コマンド, 86, 211
 fsync システムコール, 257
 fs システムファイル, 373
 /fs ディレクトリ, 89, 90, 97, 378, 382
 ftell ライブラリルーチン, 302
 ftime ライブラリルーチン, 302
 ftok ライブラリルーチン, 302
 ftpd コマンド, 214

ftpusers system ファイル, 373
ftp コマンド, 214
ftruncate システムコール, 257
ftw ライブラリルーチン, 303
fumount コマンド, 214
func_to_decimal ライブラリルーチン, 303
fusage コマンド, 214
fuser コマンド, 215
fwrite ライブラリルーチン, 303
fwtmp コマンド, 215

G

/games ディレクトリ, 381
gamma ライブラリルーチン, 303
garbagedlines ライブラリルーチン, 303
gcd ライブラリルーチン, 303
gconvert ライブラリルーチン, 303
gcore コマンド, 215
gcvt ライブラリルーチン, 303
generic_args コマンド, 215
getacdir ライブラリルーチン, 303
getacflg ライブラリルーチン, 303
getacinfo ライブラリルーチン, 303
getacmin ライブラリルーチン, 303
getauditflagsbin ライブラリルーチン, 303
getauditflagschar ライブラリルーチン, 303
getaudit システムコール, 257
getbegyx ライブラリルーチン, 304
getcap ライブラリルーチン, 304
getchar ライブラリルーチン, 304
getch ライブラリルーチン, 304
getcwd ライブラリルーチン, 185, 304, 308
getc ライブラリルーチン, 304
getdate ライブラリルーチン, 347
getdents system call, 257
getdirentries システムコール, 257
getdomainname システムコール, 257
getdtablesize システムコール, 257
getegid システムコール, 257
getenv ライブラリルーチン, 304
geteuid システムコール, 258
getexportent ライブラリルーチン, 304
getexportopt ライブラリルーチン, 304
getfauditflags ライブラリルーチン, 305
getfsent ライブラリルーチン, 305

getfsfile ライブラリルーチン, 305
getfsspec ライブラリルーチン, 305
getfstype ライブラリルーチン, 305
getgid システムコール, 258
getgraent ライブラリルーチン, 305
getgranam ライブラリルーチン, 305
getgrent ライブラリルーチン, 305
getgrgid ライブラリルーチン, 305
getgrnam ライブラリルーチン, 305
getgroups システムコール, 258
gethostbyaddr ライブラリルーチン, 305
gethostbyname ライブラリルーチン, 305
gethostent ライブラリルーチン, 305
gethostid システムコール, 258
gethostname システムコール, 258
getitimer システムコール, 258
getlogin ライブラリルーチン, 305
getmaxyx ライブラリルーチン, 305
getmntent ライブラリルーチン, 306
getmsg システムコール, 259
getnetbyaddr ライブラリルーチン, 306
getnetbyname ライブラリルーチン, 306
getnetent ライブラリルーチン, 306
getnetgrent ライブラリルーチン, 306
getnetname ライブラリルーチン, 306
getoptcvt コマンド, 215
getopts コマンド, 215
getopt コマンド, 215
getopt ライブラリルーチン, 306
getpagesize システムコール, 259
getpass ライブラリルーチン, 306
getpeername システムコール, 259
getpgid システムコール, 259
getpgrp システムコール, 259
getpid システムコール, 259
getppid システムコール, 260
getpriority システムコール, 260
getprotobyname ライブラリルーチン, 306
getprotobynumber ライブラリルーチン, 306
getprotoent ライブラリルーチン, 306
getpublickey ライブラリルーチン, 306
getpwaent ライブラリルーチン, 307
getpwanam ライブラリルーチン, 307
getpwent ライブラリルーチン, 307

getpwnam ライブラリルーチン, 307
getpwuid ライブラリルーチン, 307
getpw ライブラリルーチン, 306
getrlimit システムコール, 257, 260, 271, 358
getrpcbyname ライブラリルーチン, 307
getrpcbynumber ライブラリルーチン, 307
getrpcport ライブラリルーチン, 307
getrusage システムコール, 260, 359
getsecretkey ライブラリルーチン, 307
getservbyname ライブラリルーチン, 307
getservbyport ライブラリルーチン, 307
getserverent ライブラリルーチン, 307
getsockname システムコール, 260
getsockopt システムコール, 260
getstr ライブラリルーチン, 307
getsubopt ライブラリルーチン, 307
getsyx ライブラリルーチン, 307
gets ライブラリルーチン, 307
gettable コマンド, 215
gettext ライブラリルーチン, 185, 307
gettimeofday システムコール, 260
gettmode ライブラリルーチン, 308
getttyent ライブラリルーチン, 308
getttynam ライブラリルーチン, 308
gettytab データベース, 373
getty コマンド, 215
getuid システムコール, 261
getusershell ライブラリルーチン, 308
getut ライブラリルーチン, 185
getvfsany ライブラリルーチン, 305
getvfsent ライブラリルーチン, 305
getvfsfile ライブラリルーチン, 305
getwd ライブラリルーチン, 308
getw ライブラリルーチン, 308
getyx ライブラリルーチン, 308
get_alarm コマンド, 215
get_myaddress ライブラリルーチン, 306
get_selection コマンド, 215
get コマンド, 176, 215
gfxtool コマンド, 215
gigipolot コマンド, 215
glob コマンド, 215
glossary, 394
gmtime ライブラリルーチン, 308

goto コマンド, 215
gpconfig コマンド, 215
gprof コマンド, 179, 216
graphical user interfaces (GUIs)
 インストール, 9
graph コマンド, 216
grep コマンド, 216
group.adjunct システムファイル, 373
groups コマンド, 216
group システムファイル, 39, 373
grpauth ライブラリルーチン, 308
grpck コマンド, 216
gsignal ライブラリルーチン, 308
GSS-API, 46
gtty ライブラリルーチン, 308
gxtest コマンド, 216

H

halfdelay ライブラリルーチン, 309
halt コマンド, 72, 77, 78, 216
hashcheck コマンド, 216
hashmake コマンド, 216
hashstat コマンド, 216
hasmntopt ライブラリルーチン, 309
has_ic ライブラリルーチン, 309
has_il ライブラリルーチン, 309
hcreate ライブラリルーチン, 309
hdestroy ライブラリルーチン, 309
head コマンド, 216
help_open コマンド, 216
help コマンド, 176, 216
holidays system ファイル, 373
\$HOME/.cshrc ファイル, 53
\$HOME/.login ファイル, 53
\$HOME/.profile ファイル, 53
\$HOME/ ファイル, 53
/home ディレクトリ, 379
/home ファイルシステム, 83
host2netname ライブラリルーチン, 309
hostid コマンド, 216
hostname コマンド, 216
hostrfs コマンド, 216
hosts.equiv システムファイル, 373
hosts データベース, 373
hp7221plot コマンド, 216

hpplot コマンド, 216
hsearch ライブラリルーチン, 309
HSFS (CD-ROM ファイルシステム), 68, 84,
101
htable コマンド, 217
HUGE_VAL ライブラリルーチン, 309
HUGE ライブラリルーチン, 309
hypot ライブラリルーチン, 309

I

i386 コマンド, 217
iAPX286 コマンド, 217
icheck コマンド, 217
Icon Edit tool (OpenWindows), 217
iconedit コマンド, 217
iconvコマンド用変換テーブル, 382
/iconv ディレクトリ, 382
iconv用変換テーブル, 382
idload コマンド, 217
idlok ライブラリルーチン, 310
id コマンド, 217, 247
ID データ型、拡張された, 10
ieee_flags ライブラリルーチン, 310
ieee_functions ライブラリルーチン, 310
ieee_handler ライブラリルーチン, 310
ieee_retrospective ライブラリルーチン, 310
ifconfig コマンド, 217
IIIMP (internet intranet input method
protocol), 131
ilogb ライブラリルーチン, 310
imemtest コマンド, 217
implot コマンド, 217
in.comsat コマンド, 217
in.fingerd コマンド, 217
in.ftpd コマンド, 217
in.named コマンド, 217, 225
in.rexcd コマンド, 233
in.rexd コマンド, 233
in.rexecd コマンド, 217
in.rlogind コマンド, 217
in.routed コマンド, 217
in.rshd コマンド, 217
in.rwhod コマンド, 218
in.talkd コマンド, 218
in.telnetd コマンド, 218
in.tftpd コマンド, 218
in.tnamed コマンド, 218
in.uucpd コマンド, 218
inch ライブラリルーチン, 310
/include ディレクトリ, 96, 381
indent.pro システムファイル, 373
indentpro_to_defaults コマンド, 218
indent コマンド, 179, 218
index ライブラリルーチン, 310
indxbib コマンド, 218
inetboot コマンド, 74, 189, 190
inetd.conf データベース, 373
inetd コマンド, 218
inet_lnaof ライブラリルーチン, 310
inet_makeaddr ライブラリルーチン, 310
inet_netof ライブラリルーチン, 310
inet_network ライブラリルーチン, 310
inet_ntoa ライブラリルーチン, 310
/inet ディレクトリ, 91, 378
infinity ライブラリルーチン, 310
infocmp コマンド, 218
init.d スクリプト, 232, 378
initgroups ライブラリルーチン, 310
initscr ライブラリルーチン, 311
initstate ライブラリルーチン, 311
inittab ファイル, 73, 75
init コマンド
 changes in, 218
 使用, 74
 使用する, 76
 説明, 73
 代替コマンド, 213
 によって置き換えられたコマンド, 72
 の変更, 75
inline コマンド, 179, 218
innetgr ライブラリルーチン, 311
inodes, クリアする, 86
inodesをクリアする, 86
input_from_defaults コマンド, 218
insch ライブラリルーチン, 311
insertln ライブラリルーチン, 311
insert_brackets コマンド, 219
insque ライブラリルーチン, 311
install4x プログラム, 112, 113
installboot コマンド, 73, 189, 219
installtxt コマンド, 184, 219
install コマンド, 219

internat システムファイル, 373
internet intranet input method protocol
(IIIMP), 131
intrflush ライブラリルーチン, 311
intr コマンド, 219
ioctl 要求, 163, 166, 261
iostat コマンド, 219
ipallocald コマンド, 219
ipalloc ライブラリルーチン, 311
ipcrm コマンド, 219
ipcs コマンド, 219
irint ライブラリルーチン, 311
isainfo コマンド, 219
isalnum ライブラリルーチン, 311
isalpha ライブラリルーチン, 311
isascii ライブラリルーチン, 311
isatty ライブラリルーチン, 311
iscntrl ライブラリルーチン, 311
isdigit ライブラリルーチン, 311
isendwin ライブラリルーチン, 312
isgraph ライブラリルーチン, 312
isinf ライブラリルーチン, 312
islower ライブラリルーチン, 312
isnan ライブラリルーチン, 312
isnormal ライブラリルーチン, 312
isprint ライブラリルーチン, 312
ispunct ライブラリルーチン, 312
issecure ライブラリルーチン, 312
isspace ライブラリルーチン, 312
issubnormal ライブラリルーチン, 312
isupper ライブラリルーチン, 312
isxdigit ライブラリルーチン, 312
iszero ライブラリルーチン, 312
itom ライブラリルーチン, 312

J

j0 ライブラリルーチン, 312
j1 ライブラリルーチン, 313
jn ライブラリルーチン, 313
join コマンド, 219
jrand48 ライブラリルーチン, 313

K

kadb コマンド, 160
kadb コマンド, xv, 161, 220

Kerberos セキュリティ, 49
kernel
 crash dump ディレクトリ, 379
 modules
 directory search path, 379, 381
 /kernel/drv ディレクトリ, 97
 /kernel/exec ディレクトリ, 97
 /kernel/fs ディレクトリ, 97
 /kernel/misc ディレクトリ, 97
 /kernel/sched ディレクトリ, 97
 /kernel/strmod ディレクトリ, 97
 /kernel/sys ディレクトリ, 98
 /kernel/unix ディレクトリ, 30, 74, 98, 188
 /kernel ディレクトリ, 81, 97, 188, 379, 381
keyenvoy コマンド, 220
keylogin コマンド, 220
keylogout コマンド, 220
keyname ライブラリルーチン, 313
keypad ライブラリルーチン, 313
keyserv コマンド, 220
keytables システムファイル, 373
 /keytables ディレクトリ, 383
key_decryptsession ライブラリルーチン, 313
key_encryptsession ライブラリルーチン, 313
key_gendes ライブラリルーチン, 313
key_setsecret ライブラリルーチン, 313
kgmon コマンド, 220
killchar ライブラリルーチン, 313
killpg システムコール, 262
kill コマンド, 220
kill システムコール, 261
klm_prot ライブラリルーチン, 313
Korn シェル, 51 - 53
 制限付きシェル, 47
ksh コマンド, 51 - 53
ksyms ファイル, 162
kvm_close ライブラリルーチン, 313
kvm_getcmd ライブラリルーチン, 313
kvm_getproc ライブラリルーチン, 313
kvm_getu ライブラリルーチン, 313
kvm_nextproc ライブラリルーチン, 313
kvm_nlist ライブラリルーチン, 313
kvm_open ライブラリルーチン, 313
kvm_read ライブラリルーチン, 313
kvm_setproc ライブラリルーチン, 314
kvm_write ライブラリルーチン, 314

/kvm ディレクトリ, 381

L

l3tol ライブラリルーチン, 185, 314

l64a ライブラリルーチン, 314

labelit コマンド, 87, 105, 220

label ライブラリルーチン, 314

langinfo ライブラリルーチン, 185

lastcomm コマンド, 220

lastlogin コマンド, 220

last コマンド, 220

lcong48 ライブラリルーチン, 314

ldaclose ライブラリルーチン, 314

ldahread ライブラリルーチン, 314

ldaopen ライブラリルーチン, 314

LDAP (lightweight directory access protocol), 131

ldclose ライブラリルーチン, 314

ldconfig コマンド, 220

ldd コマンド, 220

ldexp ライブラリルーチン, 315

ldfcn ライブラリルーチン, 315

ldfhread ライブラリルーチン, 315

ldgetname ライブラリルーチン, 315

ldlinit ライブラリルーチン, 315

ldlitem ライブラリルーチン, 315

ldlread ライブラリルーチン, 315

ldlseek ライブラリルーチン, 315

ldnlseek ライブラリルーチン, 315

ldnrseek ライブラリルーチン, 315

ldnshread ライブラリルーチン, 315

ldnsseek ライブラリルーチン, 315

ldohseek ライブラリルーチン, 315

ldopen ライブラリルーチン, 315

“Operation not applicable for FSType” メッセージ, 87

ldrseek ライブラリルーチン, 315

ldshread ライブラリルーチン, 315

ldsseek ライブラリルーチン, 316

ldtbindindex ライブラリルーチン, 316

ldtbread ライブラリルーチン, 316

ldtbseek ライブラリルーチン, 316

ld コマンド, 176, 220

leaveok ライブラリルーチン, 316

leave コマンド, 221

lex コマンド, 177, 221

lfind ライブラリルーチン, 316

lgamma ライブラリルーチン, 316

libc ディレクトリ, 184

libdevinfo, 10

libintl ディレクトリ, 184

/libp ディレクトリ, 170, 382

libsocket ディレクトリ, 182

libw ディレクトリ, 183

/libxpg2.a ライブラリ, 184

/libxpg ディレクトリ, 184

lightweight directory access protocol (LDAP), 131

limits, xv

linemod ライブラリルーチン, 316

line コマンド, 221

line ライブラリルーチン, 316

link コマンド, 221

link システムコール, 262

link システムファイル, 373

lint コマンド, 151, 179, 221

lint ライブラリ, 151

listen port monitor, 221

listen システムコール, 262

listen ポートモニタ, 27, 126

ln コマンド, 221

loadkeys コマンド, 221

/local.cshrc ファイル, 53

/local.login ファイル, 54

/local.profile ファイル, 54

localdtconv ライブラリルーチン, 316

localeconv ライブラリルーチン, 316

locale データベース, 373, 382

localization データベース, 382

localtime ライブラリルーチン, 316, 354

/local ディレクトリ, 383

lockd コマンド, 221

lockf ライブラリルーチン, 316

locks

network lock デモン, 387

ロックファイルのスパーリング, 380

lockscreen コマンド, 221

/locks ディレクトリ, 380

LOFS (ループバックファイルシステム), 84

log10 ライブラリルーチン, 317

log1p ライブラリルーチン, 317

log2 ライブラリルーチン, 317

logb ライブラリルーチン, 317

- logger コマンド, 222
- login コマンド, 222
- login ファイル, 47, 53
- logname コマンド, 222
- logname ライブラリルーチン, 185
- log ファイル
 - cron, 380
 - Service Access Facility (SAF), 380
 - uucp, 381
 - システム, 379
- log ライブラリルーチン, 316
- longjmp ライブラリルーチン, 317
- _longjmp ライブラリルーチン, 314
- longname ライブラリルーチン, 317
- lorder コマンド, 177
- lpadmin コマンド, 123, 222
- lpc コマンド, 122, 123, 222
- lpd コマンド, 122, 222, 387
- lpmove コマンド, 124
- lpq コマンド, 26, 122, 222, 385
- lprm コマンド, 26, 122, 123, 222, 385
- lprof コマンド, 241
- lpr コマンド, 26, 122, 222, 385
- lpsched コマンド, 387
- lpstat コマンド, 26, 122, 222, 385
- lpssystem コマンド, 123
- lpstest コマンド, 222
- lp コマンド, 26, 122, 124, 222, 385
- lrand48 ライブラリルーチン, 317
- lsearch ライブラリルーチン, 317
- lseek システムコール, 262
- lstat システムコール, 262
- lsw コマンド, 222
- ls コマンド, 101, 222
- lto13 ライブラリルーチン, 317
- lwp_checkstkset ライブラリルーチン, 317
- lwp_create ライブラリルーチン, 318
- lwp_ctxinit ライブラリルーチン, 318
- lwp_ctxmemget ライブラリルーチン, 318
- lwp_ctxmemset ライブラリルーチン, 318
- lwp_ctxremove ライブラリルーチン, 318
- lwp_ctxset ライブラリルーチン, 318
- lwp_datastk ライブラリルーチン, 318
- lwp_destroy ライブラリルーチン, 318
- lwp_enumerate ライブラリルーチン, 318
- lwp_errstr ライブラリルーチン, 318

- lwp_fpset ライブラリルーチン, 318
- lwp_geterr ライブラリルーチン, 318
- lwp_getregs ライブラリルーチン, 318
- lwp_getstate ライブラリルーチン, 318
- lwp_join ライブラリルーチン, 318
- lwp_libcset ライブラリルーチン, 318
- lwp_newstk ライブラリルーチン, 318
- lwp_perror ライブラリルーチン, 318
- lwp_ping ライブラリルーチン, 318
- lwp_resched ライブラリルーチン, 318
- lwp_resume ライブラリルーチン, 319
- lwp_self ライブラリルーチン, 319
- lwp_setpri ライブラリルーチン, 319
- lwp_setregs ライブラリルーチン, 319
- lwp_setstkcache ライブラリルーチン, 319
- lwp_sleep ライブラリルーチン, 319
- lwp_stkcswwset ライブラリルーチン, 319
- lwp_suspend ライブラリルーチン, 319
- lwp_yield ライブラリルーチン, 319

M

- m4 コマンド, 177, 222
- m68k コマンド, 222
- mach コマンド, 223
- madd ライブラリルーチン, 319
- madvise ライブラリルーチン, 319
- magic システムファイル, 373
- mail
 - 構成情報ディレクトリ, 378
 - ディレクトリ, 380, 387
 - 補助プログラムのディレクトリ, 382
 - 待ち行列のディレクトリ, 380
- mailrc_to_defaults コマンド, 223
- mailstat コマンド, 223
- Mail Tool (OpenWindows), 223
- mailto インタフェース, 57, 223
- mailx コマンド, 57, 222, 223, 383
- /mailx ディレクトリ, 383
- mailxのヘルプファイルコマンド, 383
- mailx プログラム, 23
- mail コマンド, xv, 56, 57, 202
- Mail コマンド, 222
- mail コマンド, 223
- makedbm コマンド, 223
- MAKEDEV 環境変数, 74

makedev コマンド, 223
 Makefiles, 171, 172
 makekey コマンド, 223
 make コマンド, 171, 172, 177, 223
 mallocmap ライブラリルーチン, 320
 malloc_debug ライブラリルーチン, 320
 malloc_verify ライブラリルーチン, 320
 malloc ライブラリルーチン, 185, 319
 man.cf ファイル, 59
 man68010 コマンド, 224
 man68020 コマンド, 224
 man68881version コマンド, 224
 MANPATH 環境変数, 60
 MANSECTS 環境変数, 60
 man コマンド, 59 - 61, 224
 /man ディレクトリ, 58, 60
 man ページ

- whatis データベース, 60
- windex database, 199
- windex データベース, 60
- ディレクトリ構成の変更, 59

 matherr ライブラリルーチン, 320
 max_normal ライブラリルーチン, 320
 max_subnormal ライブラリルーチン, 320
 mblen ライブラリルーチン, 320
 mbstowcs ライブラリルーチン, 320
 mbtowc ライブラリルーチン, 320
 mcmp ライブラリルーチン, 320
 mconnect コマンド, 224
 mcs コマンド, 178
 mctl システムコール, 262
 mdiv ライブラリルーチン, 320
 memalign ライブラリルーチン, 320
 memccpy ライブラリルーチン, 320
 memchr ライブラリルーチン, 320
 memcmp ライブラリルーチン, 286, 320
 memcntl システムコール, 262
 memcpy ライブラリルーチン, 286, 320
 memset ライブラリルーチン, 287, 320
 merging ファイル, 37, 39
 mesg コマンド, 224
 metadb コマンド, 33
 metastat コマンド, 33
 meta ライブラリルーチン, 320
 mfree ライブラリルーチン, 320
 mincore system call, 262
 min_normal ライブラリルーチン, 321
 min_subnormal ライブラリルーチン, 321
 min ライブラリルーチン, 320
 /misc ディレクトリ, 97
 mkdir コマンド, 224
 mkdir システムコール, 263
 mkfifo system call, 263
 mkfile コマンド, 224
 mkfs コマンド, 104
 mkfs コマンド, 87, 103, 224
 mknod コマンド, 224
 mknod システムコール, 264
 mkproto コマンド, 224
 mkstemp ライブラリルーチン, 321
 mkstr コマンド, 224
 mktemp ライブラリルーチン, 321
 mktime ライブラリルーチン, 354
 mlockall ライブラリルーチン, 321
 mlock ライブラリルーチン, 321
 mmap システムコール, 264
 mnttab ファイル, 387
 /mnt ディレクトリ, 379
 moddebug マクロ, 161
 moddir 変数, 21, 188
 modf ライブラリルーチン, 321
 modinfo コマンド, 187, 225
 modload コマンド, 20, 188, 191, 224
 modstat コマンド, 225
 modunload コマンド, 20, 188, 225
 monacct コマンド, 225
 moncontrol ライブラリルーチン, 322
 MONITOR ライブラリルーチン, 319
 monitor ライブラリルーチン, 322
 monstartup ライブラリルーチン, 322
 mon_break ライブラリルーチン, 321
 mon_cond_enter ライブラリルーチン, 321
 mon_create ライブラリルーチン, 321
 mon_destroy ライブラリルーチン, 321
 mon_enter ライブラリルーチン, 321
 mon_enumerate ライブラリルーチン, 321
 mon_exit ライブラリルーチン, 321
 mon_waiters ライブラリルーチン, 322
 more コマンド, 225
 Motif admin ツール, 4
 mountall コマンド, 87, 386
 mountd コマンド, 225
 mount_tfs コマンド, 225

mount コマンド, 87, 100, 225, 386
mount システムコール, 265
mout ライブラリルーチン, 322
move ライブラリルーチン, 323
mprotect system call, 265
/mqueue ディレクトリ, 380
mrand48 ライブラリルーチン, 323
msgctl システムコール, 266
msgfmt コマンド, 184, 219
msgget システムコール, 266
msgrcv システムコール, 266
msgsnd システムコール, 266
msg_enumrecv ライブラリルーチン, 323
msg_enumsend ライブラリルーチン, 323
MSG_RECVALL ライブラリルーチン, 319
msg_rcv ライブラリルーチン, 323
msg_reply ライブラリルーチン, 323
msg_send ライブラリルーチン, 323
msub ライブラリルーチン, 323
msync システムコール, 266
msync ライブラリルーチン, 323
mtab system ファイル, 387
mtab システムファイル, 373
mtio ioctl, 163, 165
mtx ライブラリルーチン, 323
mt コマンド, 225
MT (マルチスレッド) カーネル, 10
mult ライブラリルーチン, 323
munlockall ライブラリルーチン, 323
munlock ライブラリルーチン, 323
munmap システムコール, 266
mutexes, 161
mutex マクロ, 161
mvaddch ライブラリルーチン, 323
mvaddstr ライブラリルーチン, 323
mvcur ライブラリルーチン, 324
mvdelch ライブラリルーチン, 324
mvgetch ライブラリルーチン, 324
mvgetstr ライブラリルーチン, 324
mvinch ライブラリルーチン, 324
mvinsch ライブラリルーチン, 324
mvprintw ライブラリルーチン, 324
mvscanw ライブラリルーチン, 324
mvwaddch ライブラリルーチン, 324
mvwaddstr ライブラリルーチン, 324
mvwdelch ライブラリルーチン, 324
mvwgetch ライブラリルーチン, 325

mvwgetstr ライブラリルーチン, 325
mvwinch ライブラリルーチン, 325
mvwinsch ライブラリルーチン, 325
mvwin ライブラリルーチン, 325
mvwprintw ライブラリルーチン, 325
mvwscanw ライブラリルーチン, 325
mv コマンド, 225

N

named コマンド, 225
NAMEFS (ネームファイルシステム), 81, 85
Name Service Switch, 136, 182
napms ライブラリルーチン, 325
nawk コマンド, 225
ncheck コマンド, 87, 207, 225
ndbootd コマンド, 225
neqn コマンド, 225
netdir_getbyname ライブラリルーチン, 306
netgroup システムファイル, 39, 373
netmasks データベース, 374
netname2host ライブラリルーチン, 326
netname2user ライブラリルーチン, 326
netrc システムファイル, 374
netstat コマンド, 225
/netsvc ディレクトリ, 235, 382
Network Interface Tap (NIT), 182
network lock デーモン, 387
networks データベース, 374
net_addr ライブラリルーチン, 325
newaliases コマンド, 225
newfs コマンド, 103, 226
newgrp コマンド, 226
newkey コマンド, 226
newpad ライブラリルーチン, 326
/news ディレクトリ, 380
newterm ライブラリルーチン, 326
newwin ライブラリルーチン, 326
nextafter ライブラリルーチン, 326
nextkey ライブラリルーチン, 326
NFS
 共用 (エクスポートされた) NFS ファイル
 システムのリスト, 386
 コマンドの変更, 130
 セキュリティ, 45
 デーモン, 382, 387
 に対するサポート, 84

- バイナリプロセス, 387
- ファイルシステムの自動マウント shared through, 21
- 補助プログラムとデーモンのディレクトリ, 382
- マウントされたNFSファイルシステムのリスト, 386
- マウント要求 サーバ, 387
- nfsd コマンド, 226, 387
- nfsstat コマンド, 226
- nfssvc システムコール, 266
- nfssys system call, 266
- /nfs ディレクトリ, 382, 387
- nice コマンド, 226
- nice ライブラリルーチン, 326
- nint ライブラリルーチン, 326
- NIS+ (Network Information Services Plus), xv
 - database ディレクトリ, 380
 - DNS 対, 136, 137
 - NIS 対, 136, 137
 - TCP/IP と, 129
 - アップグレードの計画, 139
 - 説明, 136, 181
 - へのNIS の移行, 138, 139
- NIS+ (ネットワーク情報サービスプラス)
 - 検索, 25
 - 更新, 25
 - 描写, 5, 9, 25
- NIS (Network Information Services), xv
 - NIS+ 対, 136, 137
 - NIS+ の移行, 138, 139
- NIS (Network Information Services), クライアント, xv
- nispaswd コマンド, 48
- /nis ディレクトリ, 380
- NIS (ネットワーク情報サービス)
 - クライアントシステムデータの変換, 37
 - マスタサーバ
 - システムデータの変換, 37
 - システムデータを保存, 34
- NIT (Network Interface Tap), 182
- nlist ライブラリルーチン, 326
- nlm_prot ライブラリルーチン, 326
- nlsadmin コマンド, 27, 226
- nl_init ライブラリルーチン, 326
- nl_langinfo ライブラリルーチン, 326
- nl コマンド, 226

- nl ライブラリルーチン, 326
- nm コマンド, 177, 227
- nocbreak ライブラリルーチン, 326
- nocrmode ライブラリルーチン, 327
- nodelay ライブラリルーチン, 327
- noecho ライブラリルーチン, 327
- nohup コマンド, 227
- nonl ライブラリルーチン, 327
- nonstandard_arithmetic ライブラリルーチン, 327
- noraw ライブラリルーチン, 327
- notimeout ライブラリルーチン, 327
- nrand84 ライブラリルーチン, 327
- nroff コマンド, 227, 382 - 384
- nroff用端末テーブル, 383
- nslookup コマンド, 227
- nsquery コマンド, 227
- nsswitch.conf ファイル, 182
- /nterm ディレクトリ, 383
- ntohl ライブラリルーチン, 327
- ntohs ライブラリルーチン, 327
- nulladm コマンド, 227

O

- /oasys ディレクトリ, 383
- objdump コマンド, 179
- od コマンド, 227
- old-analyze コマンド, 227
- old-cat コマンド, 227
- old-clocktool コマンド, 227
- old-compact コマンド, 227
- old-eyacc コマンド, 227
- old-filemerge コマンド, 227
- old-make コマンド, 227
- old-perfmon コマンド, 227
- old-prmail コマンド, 227
- old-pti コマンド, 227
- old-setkeys コマンド, 228
- old-sun3cvt コマンド, 228
- old-syslog コマンド, 228
- old-uncompact コマンド, 228
- old-vc コマンド, 228
- /old ディレクトリ, 96, 178, 383
- OLIT (OPEN LOOK Intrinsic Toolkit), 175
- on_exit ライブラリルーチン, 328

on コマンド, 228
 opendir ライブラリルーチン, 328
 openlog ライブラリルーチン, 328
 OPEN LOOK Intrinsics Toolkit (OLIT), 175
 openpl ライブラリルーチン, 328
 OpenWindows, xv
 clock コマンド, 205
 Developer's Guide File Chooser 対 XView
 File Chooser, 55
 Icon Edit tool, 217
 indent コマンド, 199
 Mail Tool, 223
 mail インタフェース, 57
 SunView replaced by, 240
 Text Edit tool, 242
 version differences, 55
 xlock コマンド, 221
 xset コマンド, 237
 コマンドツール, 205
 シェルツール, 238
 パフォーマンスメータツール, 229
 ファイルマネージャの変更, 28, 68
 複数のディスプレイ, 199
 プロパティウィンド, 205, 207, 223, 237
 プロパティシート, 218
 マウントまたはインストールポイン
 ト, 383
 /openwin ディレクトリ, 383
 open システムコール, 266
 /opt, 81
 /opt/sunwsprow ディレクトリ, 379
 optarg ライブラリルーチン, 328
 optind ライブラリルーチン, 328
 /options ディレクトリ, 380
 /opt ディレクトリ, xv, 91, 98, 379
 /opt ファイルシステム, 81, 83
 OSF/Motif, 142
 CDE に準拠した, 142
 overlay ライブラリルーチン, 328
 overview コマンド, 228
 overwrite ライブラリルーチン, 329

P

pack コマンド, 228, 245
 pac コマンド, 228
 pagesize コマンド, 228
 page コマンド, 228
 passwd.adjunct システムファイル, 374
 passwd2des ライブラリルーチン, 329
 passwd コマンド, 48, 229
 passwd ファイル, 40, 46, 47, 374, 387
 paste コマンド, 229
 pathconf システムコール, 267
 pause ライブラリルーチン, 329
 paxcpio コマンド, 229
 pax コマンド, 229
 pcat コマンド, 229
 PCFS (PC ファイルシステム), 84
 pclose ライブラリルーチン, 329
 pdpll コマンド, 229
 pechochar ライブラリルーチン, 329
 perfmeter コマンド, 229
 perror ライブラリルーチン, 329
 pgrep コマンド, 229
 pg コマンド, 229
 phones データベース, 374
 /pics ディレクトリ, 382
 PIC アーカイブ, 382
 ping コマンド, 229
 pipe システムコール, 267
 pkgadd コマンド, 14, 15, 175, 213
 pkgask コマンド, 175
 pkgchk コマンド, 15, 175
 pkginfo コマンド, 15, 175
 pkginfo ファイル, 173
 pkgmk コマンド, 174
 pkgparam コマンド, 175
 pkgproto コマンド, 174
 pkgrm コマンド, 14, 15, 175
 pkgtrans コマンド, 174
 /pkg ディレクトリ, 380
 pkill command, 229
 plock ライブラリルーチン, 329
 plottoa コマンド, 229
 plot コマンド, 229
 plot システムファイル, 374
 plot ライブラリルーチン, 329
 Pluggable Authentication Module (PAM), 50
 pmadm コマンド, 27, 28, 125, 127
 pmap_getmaps ライブラリルーチン, 329
 pmap_getport ライブラリルーチン, 307, 329

pmap_rmtcall ライブラリルーチン, 329
pmap_set ライブラリルーチン, 330
pmap_unset ライブラリルーチン, 330
pnoutrefresh ライブラリルーチン, 330
pnp ライブラリルーチン, 330
pod_getexit ライブラリルーチン, 330
pod_getmaxpri ライブラリルーチン, 330
pod_getmaxsize ライブラリルーチン, 330
pod_setexit ライブラリルーチン, 330
pod_setmaxpri ライブラリルーチン, 330
point ライブラリルーチン, 330
poll システムコール, 267
popen ライブラリルーチン, 330
portmap コマンド, 229
PostScript フィルタ, 57
pow ライブラリルーチン, 331
PPP (point-to-point protocol), 131
praudit コマンド, 230
prctmp コマンド, 230
prdaily コマンド, 230
prefresh ライブラリルーチン, 331
/preserve ディレクトリ, 380
printenv コマンド, 230
printf ライブラリルーチン, 331
printw ライブラリルーチン, 332
priocntl コマンド, 232, 382
priocntl システムコール, 260, 271
PROCFS (プロセスアクセスファイルシステム), 81, 84
/proc ディレクトリ, 81
/proc ファイルシステム, 83
.profile ファイル, 53
profil システムコール, 267
profil ライブラリルーチン, 322
prof コマンド, 177, 230
prof ライブラリルーチン, 332
PROM, からブートする, 73
protocols データベース, 374
prototype ファイル, 174
proto システムファイル, 374
prs コマンド, 177, 230
prtacct コマンド, 230
prtconf コマンド, 67, 214
prvtoc コマンド, 66, 67, 210
prt コマンド, 177, 230

pr コマンド, 230
pscat (C/A/T) フィルタ, 26, 57
psignal ライブラリルーチン, 332
psrinfo command, 231
pstat コマンド, 231, 386
ps コマンド, 230, 386
ptrace システムコール, 267
ptrace 要求値, 166, 167
/pts ディレクトリ, 90, 377
ptx コマンド, 231
publickey データベース, 374
/pub ディレクトリ, 383
putchar ライブラリルーチン, 332
putc ライブラリルーチン, 332
putenv ライブラリルーチン, 332
putmntent ライブラリルーチン, 283
putmsg システムコール, 267
putpwent ライブラリルーチン, 332
putp ライブラリルーチン, 332
puts ライブラリルーチン, 332
putw ライブラリルーチン, 332
pwck コマンド, 231
pwdauthd コマンド, 231
pwdauth ライブラリルーチン, 332
pwd コマンド, 231

Q

qsort ライブラリルーチン, 333
queuedefs システムファイル, 374
QuickCheck, 31
quiet_nan ライブラリルーチン, 333
quotacheck コマンド, 231
quotactl システムコール, 267
quotaoff コマンド, 231
quotaon コマンド, 231
quota コマンド, 231
quot コマンド, 102, 231

R

random ライブラリルーチン, 333
rand ライブラリルーチン, 333
ranlib コマンド, 177, 231
rarpd コマンド, 231
rasfilter8tol コマンド, 231

rasterfile システムファイル, 374
 raster image フィルタ, 57
 rastrepl コマンド, 231
 raw ディスクデバイス, 377
 raw ディスクデバイス, 用のディレクトリ, 90
 raw ディスクデバイス用ディレクトリ, 193
 raw テープデバイス, 377
 raw テープデバイス, ディレクトリ用の, 90
 raw ライブラリルーチン, 333
 rc.boot スクリプト, 74, 75, 190, 232
 rc.d スクリプト, 91, 378, 388
 rc.local スクリプト, 74, 75, 232, 388
 rc.single スクリプト, 74, 90, 190
 rc.ブートスクリプト, 90
 rc.ロケール スクリプト, 90, 190
 rcmd ライブラリルーチン, 333
 rcp コマンド, 232
 rcS.d script, 388
 rcS.d スクリプト, 91, 378
 rcS スクリプト, 74, 76, 80, 91, 190
 rc スクリプト, 74 - 76, 80, 90, 91, 95, 190, 232, 378, 388
 rdate コマンド, 232
 rdist コマンド, 232
 /rdsk ディレクトリ, 90, 193, 377
 rdump コマンド, 232
 readdir ライブラリルーチン, 333
 readlink システムコール, 268
 readv システムコール, 268
 read システムコール, 268
 realloc ライブラリルーチン, 334
 realpath ライブラリルーチン, 334
 reboot コマンド, 72, 76, 78, 232
 reboot システムコール, 269
 recvfrom システムコール, 269
 recvmsg システムコール, 269
 recv システムコール, 269
 red コマンド, 232
 refer コマンド, 232
 /refer ディレクトリ, 382
 refresh ライブラリルーチン, 334
 regexp ライブラリルーチン, 334
 registerrpc ライブラリルーチン, 334
 rehash コマンド, 232
 reject コマンド, 124
 remainder ライブラリルーチン, 335
 remexportent ライブラリルーチン, 335
 remote システムファイル, 374
 remove_brackets コマンド, 232
 remque ライブラリルーチン, 335
 rem_drv コマンド, 191
 rename システムコール, 269
 renice コマンド, 232
 repquota コマンド, 232
 resetterm ライブラリルーチン, 335
 resetty ライブラリルーチン, 335
 reset_prog_mode ライブラリルーチン, 301, 335
 reset_shell_mode ライブラリルーチン, 335
 reset コマンド, 232
 resolv.conf システムファイル, 374
 restartterm ライブラリルーチン, 335
 restore コマンド, xv, 105, 232, 386
 res_init ライブラリルーチン, 335
 res_mkquery ライブラリルーチン, 335
 res_send ライブラリルーチン, 335
 rev コマンド, 233
 rewinddir ライブラリルーチン, 335
 rewind ライブラリルーチン, 335
 rexd コマンド, 233
 rexecd コマンド, 233
 rexec ライブラリルーチン, 335
 rex ライブラリルーチン, 335
 re_comp ライブラリルーチン, 334
 re_exec ライブラリルーチン, 334
 rfadmin コマンド, 233
 rfmaster システムファイル, 374
 rfpasswd コマンド, 233
 rfstart コマンド, 233
 rfstop コマンド, 233
 /RFS ファイル システム, 80
 rfuadmin コマンド, 233
 rfudaemon コマンド, 233
 rgb システムファイル, 374
 .rhosts.equiv ファイル, 45
 rhosts システムファイル, 374
 .rhosts ファイル, 45
 rindex ライブラリルーチン, 335
 ring_alarm コマンド, 233
 rint ライブラリルーチン, 335
 ripoffline ライブラリルーチン, 336
 rksh コマンド, 47
 rlogind コマンド, 233

rlogin コマンド, 233
rmail コマンド, 234
rmdel コマンド, 177, 234
rmdir コマンド, 234
rmdir システムコール, 269
rmmount.conf ファイル, 70
rmmount コマンド, 70
rmnstat コマンド, 234
rmtab システムファイル, 374
rmt コマンド, 234
/rmt ディレクトリ, 90, 377
rm_client コマンド, 233
rm_services コマンド, 233
rm コマンド, 233
rmusers ライブラリルーチン, 336
roffbib コマンド, 234
rootmenu システムファイル, 374
root アクセス、デフォルト, 47
root ファイルシステム (/), 33, 377, 381
routed コマンド, 234
route コマンド, 234
RPC, セキュリティ, 46, 49
rpc.bootparamd コマンド, 234
rpc.etherd コマンド, 234
rpc.lockd コマンド, 234, 387
rpc.mountd コマンド, 234, 387
rpc.rexd コマンド, 234
rpc.rquotad コマンド, 235
rpc.rstatd コマンド, 235
rpc.rusersd コマンド, 235
rpc.rwalld コマンド, 235
rpc.showfhd コマンド, 235
rpc.sprayd コマンド, 235
rpc.statd コマンド, 235
rpc.user_agentd コマンド, 235
rpc.yppasswdd コマンド, 235
rpc.yppupdated コマンド, 235
rpcbind コマンド, 229
rpcb_getaddr ライブラリルーチン, 329
rpcb_getmaps ライブラリルーチン, 329
rpcb_rmtcall ライブラリルーチン, 329
rpcb_set ライブラリルーチン, 330
rpcb_unset ライブラリルーチン, 330
rpcgen コマンド, 235
rpcinfo コマンド, 235
rpc_broadcast ライブラリルーチン, 289

rpc_call ライブラリルーチン, 288
rpc_createerr ライブラリルーチン, 336
rpc データベース, 374
rpow ライブラリルーチン, 336
rquota ライブラリルーチン, 336
rrestore コマンド, 235
rresvport ライブラリルーチン, 336
rsh コマンド, 47, 235
rstat ライブラリルーチン, 336
rtime ライブラリルーチン, 336
runacct コマンド, 235
ruptime コマンド, 235
rup コマンド, 235
rusage コマンド, 236
ruserok ライブラリルーチン, 336
rusers コマンド, 236
rusers ライブラリルーチン, 336
rwall コマンド, 236
rwall ライブラリルーチン, 336
rwho コマンド, 236

S

s. ファイル, 172
S5 (System V ファイルシステム), 86
sacadm コマンド, 27, 126
sact コマンド, 177, 236
SAC (サービスアクセス管理), 27
/sad ディレクトリ, 90, 377
SAMECV ライブラリルーチン, 337
SAMEMON ライブラリルーチン, 337
SAMETHREAD ライブラリルーチン, 337
sar コマンド, 198, 231
savecore コマンド, 236
saveterm ライブラリルーチン, 337
savetty ライブラリルーチン, 337
sa コマンド, 236
/sa ディレクトリ, 382
/sbin/init コマンド, 72 - 74, 76
/sbin/rcS スクリプト, 76, 80, 91, 95
/sbin/rc スクリプト, xv, 75, 76, 80, 90, 91, 95
/sbin ディレクトリ, xv
記述, 379
説明, 83, 89, 95
の変更, 80
sbrk システムコール, 269
sbrk ライブラリルーチン, 185

scalbn ライブラリルーチン, 337
 scalb ライブラリルーチン, 337
 scandir ライブラリルーチン, 337
 scanf ライブラリルーチン, 337
 scanw ライブラリルーチン, 338
 sccs-admin コマンド, 236
 sccs-cdc コマンド, 236
 sccs-comb コマンド, 236
 sccs-delta コマンド, 236
 sccs-get コマンド, 236
 sccs-help コマンド, 237
 sccs-prs コマンド, 237
 sccs-prt コマンド, 237
 sccs-rmdel コマンド, 237
 sccs-sact コマンド, 237
 sccs-sccsdiff コマンド, 237
 sccs-unget コマンド, 237
 sccs-val コマンド, 237
 sccsdiff コマンド, 177, 237
 sccsfile システムファイル, 375
 sccs コマンド, 177, 236
 SCCS (ソースコード管理システム), 172
 /sccs ディレクトリ, 172, 176, 178
 /sched ディレクトリ, 97
 screenblank コマンド, 237
 screendump コマンド, 237
 screenload コマンド, 237
 script コマンド, 237
 scrolldefaults コマンド, 237
 scrollok ライブラリルーチン, 338
 scroll ライブラリルーチン, 338
 scr_dump ライブラリルーチン, 338
 scr_init ライブラリルーチン, 338
 scr_restore ライブラリルーチン, 338
 SCSI ディスク
 命名規則, 64
 sd1g コマンド, 74
 sdiff コマンド, 237
 seconvert ライブラリルーチン, 338
 sed コマンド, 237
 seed48 ライブラリルーチン, 339
 seekdir ライブラリルーチン, 339
 selection_svc コマンド, 237
 select システムコール, 269, 357
 semctl システムコール, 269
 semget システムコール, 269
 semop システムコール, 269
 sendmail.cf ファイル, 40
 sendmail コマンド, 23, 234, 238
 sendmsg system call, 270
 sendto システムコール, 270
 send システムコール, 269
 Serial Port Manager (Administration Tool), 125
 Serial Port Manager (admintool), 125
 /servers ディレクトリ, 383
 Service Access Facility (SAF), 125, 127
 Service Access Controller (SAC) and, 125
 コマンド, 126
 説明, 125, 127
 ディレクトリ, 379
 補助プログラムとデーモンのディレクト
 リ, 382
 用ディレクトリ, 91
 ログアカウントファイルディレクト
 リ, 380
 ログファイルおよびアカウントファイ
 ルのディレクトリ, 97
 services システムファイル, 375
 set4 コマンド, 238
 setac ライブラリルーチン, 339
 setaudit system call, 270
 setauid システムコール, 270
 setbuffer ライブラリルーチン, 339
 setbuf ライブラリルーチン, 339
 setdomainname システムコール, 270
 setegid システムコール, 271
 setegid ライブラリルーチン, 339
 seteuid システムコール, 271
 seteuid ライブラリルーチン, 339
 setexportent ライブラリルーチン, 339
 setfsent ライブラリルーチン, 339
 setgid システムコール, 271
 setgid ライブラリルーチン, 339, 340
 setgraent ライブラリルーチン, 339
 setgrent ライブラリルーチン, 339
 setgroups system call, 270
 sethostent ライブラリルーチン, 339
 sethostname システムコール, 270
 setitimer システムコール, 270, 357
 setjmp ライブラリルーチン, 339
 _setjmp ライブラリルーチン, 337
 setkeys コマンド, 238
 setkey ライブラリルーチン, 339

setlinebuf ライブラリルーチン, 339
setlocale コマンド, 185
setlocale ライブラリルーチン, 339
setlogmask ライブラリルーチン, 340
setmntent ライブラリルーチン, 340
setnetent ライブラリルーチン, 340
setnetgrent ライブラリルーチン, 340
setpgid システムコール, 270
setpgrp システムコール, 271
setpriority システムコール, 271
setprotoent ライブラリルーチン, 340
setpwaent ライブラリルーチン, 340
setpwent ライブラリルーチン, 340
setpwfile ライブラリルーチン, 340
setregid システムコール, 271
setreuid システムコール, 271
setrgid ライブラリルーチン, 340
setrlimit システムコール, 271
setrpcnt ライブラリルーチン, 340
setruid ライブラリルーチン, 340
setscrreg ライブラリルーチン, 340
setservent ライブラリルーチン, 340
setsid コマンド, 238
setsid システムコール, 271
setsockopt system call, 271
setstate ライブラリルーチン, 340
setsyx ライブラリルーチン, 340
setterm ライブラリルーチン, 341
settimeofday システムコール, 272
setttyent ライブラリルーチン, 341
setuid システムコール, 271, 340
setuid ライブラリルーチン, 341
setupterm ライブラリルーチン, 341
setup_client コマンド, 238
setup_exec コマンド, 238
setuseraudit システムコール, 272
setusershell ライブラリルーチン, 341
setvbuf ライブラリルーチン, 341
set_alarm コマンド, 238
set_curterm ライブラリルーチン, 339
set_term ライブラリルーチン, 340
sfconvert ライブラリルーチン, 341
sgconvert ライブラリルーチン, 341
sgctl システムコール, 272
shadow ファイル, 40, 46, 387
shareall コマンド, 102, 386

shared ファイルシステム
 default root of, 379
 構成情報ディレクトリ, 378
 転送, 34, 39
 リスト, 387
sharetab ファイル, 283, 387
share コマンド, 100, 102, 212, 386
/share ファイルシステム, 99
shelltool コマンド, 238
shift_lines コマンド, 238
shmat システムコール, 272
shmctl システムコール, 272
shmdt システムコール, 272
shmget システムコール, 272
showfh コマンド, 235
showfh コマンド, 238
showmount コマンド, 100, 238, 386
shutacct コマンド, 238
shutdown コマンド, 72, 76, 77, 238
shutdown システムコール, 272
sh コマンド, 51 - 53, 238
sigaction システムコール, 272, 273
sigaction ライブラリルーチン, 341
sigaddset ライブラリルーチン, 341
sigaltstack システムコール, 273
sigblock システムコール, 272
sigdelset ライブラリルーチン, 341
sigemptyset ライブラリルーチン, 341
sigfillset ライブラリルーチン, 341
sigfpe ライブラリルーチン, 342
siginterrupt ライブラリルーチン, 342
sigismember ライブラリルーチン, 342
siglongjmp ライブラリルーチン, 314, 342
sigmask システムコール, 272
signaling_nan ライブラリルーチン, 342
signal ライブラリルーチン, 342
signbit ライブラリルーチン, 342
significant ライブラリルーチン, 342
sigpause システムコール, 272
sigpending システムコール, 272
sigprocmask システムコール, 273
sigprocmask ルーチン, 273
sigsetjmp ライブラリルーチン, 337, 342
sigsetmask システムコール, 273
sigsetops ルーチン, 272

sigstack システムコール, 273
 sigsuspend システムコール, 273
 sigvec システムコール, 273
 single_precision ライブラリルーチン, 343
 single_to_decimal ライブラリルーチン, 343
 sinh ライブラリルーチン, 343
 sin ライブラリルーチン, 343
 size コマンド, 177, 239
 /skel ディレクトリ, 54, 379
 skyversion コマンド, 239
 sleep コマンド, 239
 sleep ライブラリルーチン, 343
 slk_clear ライブラリルーチン, 343
 slk_init ライブラリルーチン, 343
 slk_label ライブラリルーチン, 343
 slk_noutrefresh ライブラリルーチン, 343
 slk_refresh ライブラリルーチン, 343
 slk_restore ライブラリルーチン, 343
 slk_set ライブラリルーチン, 344
 slk_touch ライブラリルーチン, 344
 /sm.bak ディレクトリ, 379
 sm_inter ライブラリルーチン, 344
 sm システムファイル, 375
 /sm ディレクトリ, 379
 /snadm ディレクトリ, 96, 384
 snoop コマンド, 212
 socketpair system call, 273
 sockets, 273
 socket システムコール, 273
 sockio ioctl, 163, 165
 soelim コマンド, 239
 Solaris 2.4 操作環境, xv
 Solaris 2.6 DDI/DKI, 10
 Solaris 2.6 SPARC DDI/DKI, 10
 Solaris 2.6 サービス, xv
 Solaris 2.6操作環境
 関連マニュアル, xxi
 Solaris 2.6 動作環境
 インストール機能, 29
 主な変更, 14
 Solaris 2.7
 主な変更, 28
 Solaris 7
 アドミンツール, 24
 Solaris 2.7 サーバ, SunOS リリース 4.x デイスク
 レスクライアントサポ
 ート, 111
 Solaris CDE
 OpenWindowsから移動する, 145
 スタイル・マネージャ, 144
 ファイル・マネージャ, 145
 フロントパネル, 143
 Solaris 操作環境, xv
 Solaris 動作環境
 SVR4 vs., 4, 6, 11
 移植性, 5
 開発者機能, 9, 10
 機能, 3, 6, 10
 互換性, 4, 6
 システム管理者用の機能, 8, 9
 スケーラビリティ, 6, 10
 相互運用性, 6
 大規模な組織, 6
 メリット, 3, 6
 ユーザ機能, 7
 Solstice DiskSuite, インストール, 32, 33
 sortbib コマンド, 239
 sort コマンド, 239
 source code ディレクトリ, 384
 space ライブラリルーチン, 344
 SPARC DDI/DKI, 10
 SPARCServer Manager, インストール, 32
 SPARCserver Manager, インストール, 33
 sparc コマンド, 239
 SPARC プラットフォーム, 10
 SPECFS (特殊デバイスファイルシステム), 84
 spellin コマンド, 239
 spell コマンド, 239, 382, 383
 /spell ディレクトリ, 383
 spline コマンド, 239
 split コマンド, 239
 spoken メッセージ, 5
 spray コマンド, 239
 spray ライブラリルーチン, 344
 sprint ライブラリルーチン, 344
 sputil システムコール, 273
 sqrt ライブラリルーチン, 344
 srand48 ライブラリルーチン, 344
 srand ライブラリルーチン, 344
 /src ディレクトリ, 384
 sscanf ライブラリルーチン, 344
 signal ライブラリルーチン, 342, 344

standard_arithmetic ライブラリルーチン, 345
standend ライブラリルーチン, 345
standout ライブラリルーチン, 345
startup コマンド, 239
statfs システムコール, 273
statmon システムファイル, 375
stat system call, 273
statvfs システムコール, 273
STKTOP ライブラリルーチン, 337
store ライブラリルーチン, 345
strcasecmp ライブラリルーチン, 345
strcat ライブラリルーチン, 345
strchr ライブラリルーチン, 310, 345
strcmp ライブラリルーチン, 345
strcoll ライブラリルーチン, 345
strcpy ライブラリルーチン, 345
strcspn ライブラリルーチン, 345
strdup ライブラリルーチン, 345
streamio ioctl, 163, 165
STREAMS
 Ethernet ドライバと, 182
 描写, 10
 モジュールの位置, 97
 用の疑似ファイルシステム, 81
STREAMS 管理ドライバのエントリポイント, 90, 377
strftime コマンド, 184
strftime ライブラリルーチン, 346
strings コマンド, 239
string_to_decimal ライブラリルーチン, 346
strip コマンド, 177, 240
strlen ライブラリルーチン, 346
/strmod ディレクトリ, 97
strncasecmp ライブラリルーチン, 346
strncat ライブラリルーチン, 347
strncmp ライブラリルーチン, 347
strncpy ライブラリルーチン, 347
strpbrk ライブラリルーチン, 347
strptime ライブラリルーチン, 347
strrchr ライブラリルーチン, 335, 347
strspn ライブラリルーチン, 347
strtod ライブラリルーチン, 347
strtok ライブラリルーチン, 347
strtol ライブラリルーチン, 347
strxfrm ライブラリルーチン, 347
stty_from_defaults コマンド, 240
stty コマンド, 232, 240

stty ライブラリルーチン, 347
subpad ライブラリルーチン, 347
subwin ライブラリルーチン, 348
sum コマンド, 240
Sun C コンパイラ, 151
Sun DDI (Sun デバイスドライバインタフェース), 10
sundiag コマンド, 240
suninstall コマンド, 240
SunOS release 5.7
 デバイスの命名, 17
 ファイルシステム, 18
SunOS/BSD ソース互換パッケージ, 41, 42
SunOS 4.x
 SunOS release 5.7 との互換性, 54
 互換 SunOS release 5.7, 41
 コマンドの変更, 41
 作業環境、Solaris 2.7 を使用した, 41
SunOS 5.6
 プリントサブシステム, 122
SunOS release 5.7
 コマンドの変更, 41
 互換パッケージ, 41, 54
 セキュリティ, 45
 プリントサブシステム, 122
SunOS リリース 4.x
 Backup Copilot, 31
 QuickCheck, 31
 Solaris 2.6 主な変更について, 14
 Solaris 2.6 の作業環境, 54
 Solaris 7 主な変更, 28
 システムデータを復元, 40
 ツール, finding, 176
 ディスクパーティション情報の保存, 31, 32
 ディスクレスクライアント、用のSolaris 2.6 サーバー サポート, 111
 デバイスの命名, 17
 バックアップファイルシステム, 31, 36
 ファイルシステム情報の保存, 32
 メタデバイス構成情報, 32, 33
SunOS リリース 5.6
 Admintool, 9
 相互互換性, 4
 印刷サブシステム, 26
SunOS release 5.7
 アドミンツール, 24

- 印刷サブシステム, 26
- カーネルの構成, 20
- カーネル名, 19
- 自動マウント, 21
- ファイルシステム, 20
- SunShield 基本セキュリティ モジュール (BSM), 50
- SunView, 55, 381
- /sunview1 ディレクトリ, 381
- sunview コマンド, 240
- sunview システムファイル, 375
- SUNWhinst, 112, 113
- Sun WorkShop Debugger, 160
- sun コマンド, 240
- su コマンド, 240
- su ファイル, 47
- svcerr_auth ライブラリルーチン, 349
- svcerr_decode ライブラリルーチン, 349
- svcerr_noproc ライブラリルーチン, 349
- svcerr_noprogram ライブラリルーチン, 349
- svcerr_progvers ライブラリルーチン, 349
- svcerr_systemerr ライブラリルーチン, 349
- svcerr_weakauth ライブラリルーチン, 349
- svcfcd_create ライブラリルーチン, 349
- svcrow_create ライブラリルーチン, 349
- svctcp_create ライブラリルーチン, 349
- svcudp_bufcreate ライブラリルーチン, 349
- svcudp_create ライブラリルーチン, 350
- svc_create ライブラリルーチン, 349
- svc_destroy ライブラリルーチン, 348
- svc_dg_create ライブラリルーチン, 349
- svc_fdset ライブラリルーチン, 348
- svc_fds ライブラリルーチン, 348
- svc_fd_create ライブラリルーチン, 349
- svc_freeargs ライブラリルーチン, 348
- svc_getargs ライブラリルーチン, 348
- svc_getcaller ライブラリルーチン, 348
- svc_getreqset ライブラリルーチン, 348
- svc_getreq ライブラリルーチン, 348
- svc_raw_create ライブラリルーチン, 349
- svc_register ライブラリルーチン, 348
- svc_reg ライブラリルーチン, 348
- svc_run ライブラリルーチン, 349
- svc_sendreply ライブラリルーチン, 349
- svc_tli_create ライブラリルーチン, 349
- svc_unregister ライブラリルーチン, 349
- svc_unreg ライブラリルーチン, 349
- svc_vc_create ライブラリルーチン, 349
- svdtab システムファイル, 375
- SVR4
 - Solaris vs., 3, 4, 6, 11
 - サポートされないファイルシステムの形式, 86
 - データ交換する, 106
- svr4.make コマンド, 171, 172, 223
- sv_acquire コマンド, 240
- sv_release コマンド, 240
- sv_xv_sel_svc コマンド, 237
- swab ライブラリルーチン, 185, 350
- SWAPFS (スワップファイルシステム), 81, 85
- swapon コマンド, 241
- swapon システムコール, 273
- swap コマンド, 231, 241, 386
- swin コマンド, 241
- switcher コマンド, 241
- swmtool コマンド
 - add_services command vs., 199
 - extract_unbundled command vs., 213
 - rm_services command vs., 233
 - 描写, 14, 15
- symlink システムコール, 273
- symorder コマンド, 177, 241
- sync コマンド, 241
- sync システムコール, 273
- /sys, 19, 80
- sysadm メニューユーティリティ, 11
- syscall システムコール, 274
- sysconf システムコール, 259, 260, 274
- sysdef コマンド, 66, 208, 216
- sysinfo system call, 257, 270
- sysinfo システムコール, 258, 270
- syslog.conf システムファイル, 375
- syslogd コマンド, 241
- syslog ライブラリルーチン, 350, 359
- systems システムファイル, 375
- System V Interface Definition, 168
- System V ファイルシステム (S5), 11, 86
- system コマンド, 74
- system ライブラリルーチン, 350
- sys_siglist ライブラリルーチン, 350
- sys_unconfig コマンド, 241
- /sys ディレクトリ, 80, 98

T

- t300s コマンド, 241
- t300 コマンド, 241
- t4013 コマンド, 241
- t450 コマンド, 241
- /tabset ディレクトリ, 384
- tabs コマンド, 241
- tail コマンド, 241
- talk コマンド, 241
- tanh ライブラリルーチン, 352
- tan ライブラリルーチン, 352
- tar コマンド, xv
 - bar command vs., 201
 - changes affecting, 241
 - support for, 104
 - ustar command vs., 245
 - 影響する変更, 107
 - 説明, 106, 107
- tar システムファイル, 375
- tbl コマンド, 241
- tcdrain ライブラリルーチン, 352
- tcflow ライブラリルーチン, 352
- tcflush ライブラリルーチン, 352
- tcgetattr ライブラリルーチン, 352
- tcgetpgrp ライブラリルーチン, 352
- tcopy コマンド, 241
- tcov コマンド, 179, 241
- TCP 選択肯定応答 (SACK), 130
- TCP/IP, 129
- tcsendbreak ライブラリルーチン, 352
- tcsetattr ライブラリルーチン, 352
- tcsetpgrp ライブラリルーチン, 352
- tdelete ライブラリルーチン, 352
- tee コマンド, 241
- tektool コマンド, 242
- tek コマンド, 242
- telldir ライブラリルーチン, 352
- tell system call, 274
- telnet コマンド, 242
- tempnam ライブラリルーチン, 352
- termcap データベース, 80, 375
- terminfo データベース, 80, 122, 123, 375, 384, 387
- termio ioctl, 163, 166
- termios ioctl, 163, 166
- termio インタフェース, 308, 347
- term システムファイル, 375
 - /term ディレクトリ, 90, 378
- test コマンド, 242
- textdomain ライブラリルーチン, 185, 352
- Text Edit tool (OpenWindows), 242
- textedit_filters コマンド, 242
- textedit コマンド, 242
- TEX フィルタ, 26, 57
- tfind ライブラリルーチン, 352
- tfsd コマンド, 242
- TFS (疑似ファイルシステム), 80, 84
 - /tftpboot ディレクトリ, 34
- tftp コマンド, 242
- tgetent ライブラリルーチン, 352
- tgetflag ライブラリルーチン, 353
- tgetnum ライブラリルーチン, 353
- tgetstr ライブラリルーチン, 353
- tgoto ライブラリルーチン, 354
- threadlist マクロ, 161
- thread マクロ, 161
- tic コマンド, 242
- tigetflag ライブラリルーチン, 354
- tigetnum ライブラリルーチン, 354
- tigetstr ライブラリルーチン, 354
- timegm ライブラリルーチン, 354
- timelocal ライブラリルーチン, 354
- times ライブラリルーチン, 354
- timezone ライブラリルーチン, 354
- time コマンド, 242
- time ライブラリルーチン, 302, 354
- tip コマンド, 242
 - /tmac ディレクトリ, 384
- tmpfile ライブラリルーチン, 354
- tmpnam ライブラリルーチン, 354
 - /tmp 一時ファイルシステム (TMPFS) に対するサポート, 84
 - /tmp ディレクトリ, 379
 - /tmp ファイルシステム (TMPFS) 説明, 83
 - 転送, 33
- /tm ディレクトリ, 379
- toascii ライブラリルーチン, 354
- toc システムファイル, 375
- tolower ライブラリルーチン, 354
- _tolower ライブラリルーチン, 350
- toolplaces コマンド, 242
- ToolTalk, 7

touchline ライブラリルーチン, 355
touchoverlap ライブラリルーチン, 355
touchwin ライブラリルーチン, 355
touch コマンド, 242
toupper ライブラリルーチン, 355
_toupper ライブラリルーチン, 350
tparm ライブラリルーチン, 355
tputs ライブラリルーチン, 355
tput コマンド, 243
traceoff ライブラリルーチン, 356
traceon ライブラリルーチン, 356
traceroute ユーティリティ, 130
trace コマンド, xv, 162, 243
traffic コマンド, 243
translate システムファイル, 375
troff コマンド, 243
 の変更, 124
 プリプロセッサ, 382
 変更, 26, 58
 マクロパッケージ, 384
trpt コマンド, 243
true コマンド, 243
truncate システムコール, 274
truss コマンド, xv, 162, 178, 243
tr コマンド, 243
tsearch ライブラリルーチン, 356
tset コマンド, 243
tsort コマンド, 177, 243
ttyadm コマンド, 27
ttymon ポートモニタ, 27, 126
ttyname ライブラリルーチン, 356
ttypslot ライブラリルーチン, 356
ttypsoftcar コマンド, 243
ttypstab システムファイル, 40, 375
tty コマンド, 243
TTY デバイスの管理と問題の解決, 27
tunefs コマンド, 243
turnacct コマンド, 243
tvconfig コマンド, 243
twalk ライブラリルーチン, 356
typeahead ライブラリルーチン, 356
types システムファイル, 375
tzfile システムファイル, 375
tzsetup コマンド, 243
tzsetwall ライブラリルーチン, 356
tzset ライブラリルーチン, 356
t_accept ライブラリルーチン, 350

t_alloc ライブラリルーチン, 350
t_bind ライブラリルーチン, 350
t_close ライブラリルーチン, 350
t_connect ライブラリルーチン, 351
t_error ライブラリルーチン, 351
t_free ライブラリルーチン, 351
t_getinfo ライブラリルーチン, 351
t_getstate ライブラリルーチン, 351
t_listen ライブラリルーチン, 351
t_look ライブラリルーチン, 351
t_open ライブラリルーチン, 351
t_optmgmt ライブラリルーチン, 351
t_rcvconnect ライブラリルーチン, 351
t_rcvdis ライブラリルーチン, 351
t_rcvrel ライブラリルーチン, 351
t_rcvudata ライブラリルーチン, 351
t_rcvuderr ライブラリルーチン, 351
t_rcv ライブラリルーチン, 351
t_snddis ライブラリルーチン, 351
t_sndrel ライブラリルーチン, 351
t_sndudata ライブラリルーチン, 351
t_snd ライブラリルーチン, 351
t_sync ライブラリルーチン, 351
t_unbind ライブラリルーチン, 352

U

u370 コマンド, 244
u3b15 コマンド, 244
u3b2 コマンド, 244
u3b5 コマンド, 244
u3b コマンド, 244
uadmin system call, 269
uadmin システムコール, 273
ualarm ライブラリルーチン, 357
/ucbininclude ディレクトリ, 384
/ucblib ディレクトリ, 170, 171, 384
UFS ログ, 109
ufsbootblk コマンド, 74
ufsboot コマンド, 73, 189, 190
ufsdump コマンド, xv
 説明, 104 - 106
 代替コマンド, 211, 232, 386
 によって置き換えられたコマンド, 105
ufsrestore コマンド, xv
 クイックリファレンス, 386
 説明, 106

代替コマンド, 232, 235
 によって置き換えられたコマンド, 105
 ulimit ライブラリルーチン, 357
 ul コマンド, 244
 umask コマンド, 244
 umask システムコール, 274
 umountall コマンド, 87, 100, 386
 umount_tfs コマンド, 244
 umount コマンド, 87, 100, 244, 386
 umount システムコール, 274
 unadv コマンド, 244
 uname コマンド, 200, 216, 223, 244
 uname システムコール, 274
 uncompress コマンド, 244
 unconfigure コマンド, 244
 unctrl ライブラリルーチン, 357
 unexpand コマンド, 244
 ungetch ライブラリルーチン, 357
 ungetc ライブラリルーチン, 357
 unget コマンド, 177, 244
 unifdef コマンド, 177, 244
 uniq コマンド, 244
 units コマンド, 245
 unix, 19, 81
 Unix-to-Unix Copy (UUCP), xv, 39, 132, 134
 unix2dos コマンド, 245
 unix kernel name, xv
 unix カーネル名, 19, 81, 190
 /unix ディレクトリ, 30, 74, 98, 188
 Unix ファイルシステム (UFS), xv
 作成する, 103
 シリンダグループ, 17
 に対するサポート, 84
 unlink コマンド, 245
 unlink システムコール, 274
 unmount システムコール, 274
 unpack コマンド, 245
 unshare コマンド, 100, 386
 unwhiteout コマンド, 245
 updaters システムファイル, 375
 update コマンド, 245
 uptime コマンド, 245
 user2netname ライブラリルーチン, 357
 useradd コマンド, 55, 56
 userdel コマンド, 55, 56
 usermod コマンド, 55, 56
 users, Solaris 機能, 7

users コマンド, 245
 user ディレクトリ, デフォルトのルート, 379
 usleep ライブラリルーチン, 357
 /usr/4lib ディレクトリ, 381
 /usr/5bin ディレクトリ, 96
 /usr/5include ディレクトリ, 96
 /usr/5lib ディレクトリ, 96, 169 - 171
 /usr/bin/csh コマンド, 51 - 53, 206
 /usr/bin/ksh コマンド, 51 - 53
 /usr/bin/mailx コマンド, 57, 222, 223, 383
 /usr/bin/mail コマンド, 56, 57, 202, 223
 /usr/bin/rksh コマンド, 47
 /usr/bin/rsh コマンド, 47, 235
 /usr/bin/sh コマンド, 51 - 53, 238
 /usr/bin/sunview1 ディレクトリ, 381
 /usr/bin ディレクトリ
 generic file system administrative
 commands in, 87
 SunOS release 4.x files moved to, 96
 記述, 381
 クイックリファレンス, 387
 説明, 86
 の汎用ファイルシステム管理コマンド, 86
 プログラミングツール, 176, 178
 /usr/ccs/bin/make command, 171, 172, 177
 /usr/ccs/bin/make コマンド, 223
 /usr/ccs/bin ディレクトリ, 172, 176, 381
 /usr/ccs/lib/svr4.make command, 171, 172
 /usr/ccs/lib/svr4.make コマンド, 223
 /usr/ccs/lib ディレクトリ, 170
 /usr/ccs/lib ディレクトリ, xv, 170, 171, 176,
 381
 /usr/ccs ディレクトリ, 96, 381
 /usr/demo ディレクトリ, 381
 /usr/etc/biod コマンド, 202, 388
 /usr/etc/nfsd コマンド, 226, 387
 /usr/etc/rpc.lockd コマンド, 234, 387
 /usr/etc/rpc.mountd コマンド, 234, 387
 /usr/etc/ypbind コマンド, 248, 387
 /usr/etc ディレクトリ, xv, 96
 /usr/games ディレクトリ, 381
 /usr/include ディレクトリ, 96, 381
 /usr/kvm ディレクトリ, 381
 /usr/lib/acct ディレクトリ, 382
 /usr/lib/class ディレクトリ, 382
 /usr/lib/dict ディレクトリ, 382
 /usr/lib/font ディレクトリ, 382

- /usr/lib/fs ディレクトリ, 89, 90, 382
- /usr/lib/iconv ディレクトリ, 382
- /usr/lib/libc ディレクトリ, 184
- /usr/lib/libintl ディレクトリ, 184
- /usr/lib/libp ディレクトリ, 382
- /usr/lib/libw ディレクトリ, 183
- /usr/lib/locale ディレクトリ, 373, 382
- /usr/lib/lp/lpsched コマンド, 387
- /usr/lib/lpd コマンド, 122, 222, 387
- /usr/lib/lp ディレクトリ, xv, 382
- /usr/lib/mail ディレクトリ, xv, 382
- /usr/lib/netstvc/yp/ypbind プロセス, 248, 387
- /usr/lib/netstvc ディレクトリ, 235, 382
- /usr/lib/nfs/lockd デーモン, 387
- /usr/lib/nfs/mountd サーバ, 387
- /usr/lib/nfs/nfsd デーモン, 387
- /usr/lib/nfs ディレクトリ, 382
- /usr/lib/pics ディレクトリ, 382
- /usr/lib/refer ディレクトリ, 382
- /usr/lib/rsh コマンド, 47
- /usr/lib/saf ディレクトリ, xv, 382
- /usr/lib/sa ディレクトリ, 382
- /usr/lib/spell ディレクトリ, 383
- /usr/lib/uucp ディレクトリ, xv, 383
- /usr/lib ディレクトリ, xv, 96, 169, 170, 176, 178, 382
- /usr/local ディレクトリ, 383
- /usr/net/servers ディレクトリ, xv, 383
- /usr/oasys ディレクトリ, 383
- /usr/old ディレクトリ, 96, 178, 383
- /usr/openwin ディレクトリ, 383
- /usr/opt, 81
- /usr/opt ファイル, 81
- /usr/sadm/bin ディレクトリ, 383
- /usr/sadm/install ディレクトリ, 383
- /usr/sadm ディレクトリ, 383
- /usr/sbin/static ディレクトリ, 383
- /usr/sbin ディレクトリ, 89, 90, 96, 383, 387
- /usr/sccs ディレクトリ, 172, 176, 178
- /usr/share/lib/keytables ディレクトリ, 383
- /usr/share/lib/mailx ディレクトリ, 383
- /usr/share/lib/nterm ディレクトリ, 383
- /usr/share/lib/pub ディレクトリ, 383
- /usr/share/lib/spell ディレクトリ, 383
- /usr/share/lib/tabset ディレクトリ, 384
- /usr/share/lib/terminfo データベース, 80, 122, 123, 375, 384, 387
- /usr/share/lib/tmac ディレクトリ, 384
- /usr/share/lib/zoneinfo ディレクトリ, 384
- /usr/share/lib ディレクトリ, xv, 383
- /usr/share/man ディレクトリ, 58, 60
- /usr/share/src ディレクトリ, 384
- /usr/share ディレクトリ, 383
- /usr/share ファイルシステム
マウント, 99
- /usr/snadm ディレクトリ, 96, 384
- /usr/ucb/df コマンド, 66, 86, 101, 208
- /usr/ucb/du コマンド, 66, 101, 210
- /usr/ucb/mail コマンド, 56, 57, 202, 223
- /usr/ucbinclude ディレクトリ, 384
- /usr/ucbllib ディレクトリ, 170, 171, 384
- /usr/ucb ディレクトリ, 42, 176, 384
- /usr/vmsys ディレクトリ, 384
- /usr/xpg2bin ディレクトリ, 96
- /usr/xpg2include ディレクトリ, 96
- /usr/xpg2lib/libxpg2.a ライブラリ, 184
- /usr/xpg2lib/libxpg ディレクトリ, 184
- /usr/xpg2lib ディレクトリ, 96, 171
- /usr ディレクトリ
記述, 379
説明, 80, 95
の変更, 95
- /usr ファイルシステム
layout changes, 381, 384
説明, 82
転送, 33
マウントポイント, 379
- ustar コマンド, 245
- ustat システムコール, 274
- utimes システムコール, 274
- utime ライブラリルーチン, 358
- utmp システムファイル, 375
- uucheck コマンド, 245
- uucico コマンド, 245
- uucleanup コマンド, 245
- UUCP (Unix-to-Unix Copy), 39, 132, 134
- uucp コマンド, 234, 245
file deposit ディレクトリ, 380
構成情報ディレクトリ, 379
補助プログラムとデーモンのディレクトリ, 383

待ち行列に登録されたジョブのディレ
トリ, 380
ログと状態ファイルのディレトリ, 381
ログファイルとステータスファイルの
ディレトリ, 134
uudecode コマンド, 245
uuencode コマンド, 245
uuencode システムファイル, 376
uuglist コマンド, 132
uulog コマンド, 245
uuname コマンド, 246
uupick コマンド, 246
uusched コマンド, 246
uusend コマンド, 246
uustat コマンド, 246
uuto コマンド, 246
uuxqt コマンド, 246
uux コマンド, 246

V

vacation コマンド, 246
vadvice システムコール, 274
valloc ライブラリルーチン, 358
val コマンド, 178, 246
/var/adm ディレトリ, 379
/var/crash ディレトリ, 379
/var/cron ディレトリ, 380
/var/lp ディレトリ, xv, 380
/var/mail ディレトリ, xv, 80, 97, 380, 387
/var/news ディレトリ, 380
/var/nis ディレトリ, 380
/var/opt/ ディレトリ, 97
/var/options ディレトリ, 380
/var/opt ディレトリ, 380
/var/preserve ディレトリ, 380
/var/sadm ディレトリ, xv, 97, 380
/var/saf ディレトリ, xv, 97, 380
/var/spool/cron ディレトリ, 380
/var/spool/locks ディレトリ, 380
/var/spool/lp ディレトリ, xv, 123, 380
/var/spool/mail ディレトリ, xv, 80, 97, 387
/var/spool/queue ディレトリ, 380
/var/spool/pkg ディレトリ, 380
/var/spool/uucppublic ディレトリ, 380
/var/spool/uucp ディレトリ, xv, 380
/var/spool ディレトリ, 123, 380

/var/tmp ディレトリ, 380
/var/uucp/.Admin/perflog ファイル, 134
/var/uucp/.Admin/security ファイル, 134
/var/uucp/.Admin/アカウントファイル, 134
/var/uucp/.Admin/コマンドファイル, 134
/var/uucp ディレトリ, xv, 381
/var/yp ディレトリ, 381
varargs ライブラリルーチン, 358
/var ディレトリ, 123, 379
 changes in, 379, 381
 記述, 379
 説明, 80, 83, 97
 の変更, 80, 97
vax コマンド, 246
vc コマンド, 178
vedit コマンド, 246
vfontinfo コマンド, 246
vfont システムファイル, 376
vfork システムコール, 275
vfprintf ライブラリルーチン, 358
VFS (仮想ファイルシステムアーキテク
 チャ), 83, 89
vgrindefs データベース, 376
vgrind コマンド, 246
vhangup システムコール, 275
vidattr ライブラリルーチン, 358
vidputs ライブラリルーチン, 358
view コマンド, 246
vipw コマンド, 246
vi エディタ, 246, 380
viとexエディタ用のバックアップファイル, 380
vlimit ライブラリルーチン, 358
vmstat コマンド, 246
/vmsys ディレトリ, 384
/vmunix file, 188, 190
/vmunix ファイル, 21, 30, 74
/vol/dev/aliases/cdrom0 ファイルシステ
 ム, 69
/vol/dev/aliases/floppy0 ファイルシステ
 ム, 69
volcancel コマンド, 70
volcheck コマンド, 70
volcopy コマンド, 87, 105
vold.conf ファイル, 70
vold コマンド, 70, 81
volmissing コマンド, 70

/vol ファイルシステム, 81, 83
vplot コマンド, 246
vprintf ライブラリルーチン, 358
vsprintf ライブラリルーチン, 358
vswap コマンド, 246
vsyslog ライブラリルーチン, 359
vtimes ライブラリルーチン, 359
vtroff コマンド, 247
vwidth コマンド, 247
vwprintw ライブラリルーチン, 359
vwscanw ライブラリルーチン, 359

W

waddch ライブラリルーチン, 359
waddstr ライブラリルーチン, 360
wait3 システムコール, 276
wait4 システムコール, 276
waitpid システムコール, 276, 277
wait コマンド, 247
wait システムコール, 276
wall コマンド, 247
wattroff ライブラリルーチン, 360
wattron ライブラリルーチン, 360
wattrset ライブラリルーチン, 360
wclear ライブラリルーチン, 360
wclrtobot ライブラリルーチン, 361
wclrtoeol ライブラリルーチン, 361
wcstombs ライブラリルーチン, 361
wctomb ライブラリルーチン, 361
wc コマンド, 247
wdelch ライブラリルーチン, 362
wdeleteln ライブラリルーチン, 362
WebNFS
 SDK, 11
wechochar ライブラリルーチン, 362
werase ライブラリルーチン, 362
WEXITSTATUS システムコール, 275
wgetch ライブラリルーチン, 363
wgetstr ライブラリルーチン, 363
whatis database, 199
whatis コマンド, 247
whatis データベース, 60
what コマンド, 178, 247
whereis コマンド, 247
which コマンド, 247
whoami コマンド, 247
whois コマンド, 247

who コマンド, 245, 247
WIFEXITED システムコール, 275
WIFSIGNALED システムコール, 275
WIFSTOPPED システムコール, 275
WIFSTOPSIG システムコール, 275
winch ライブラリルーチン, 363
windex database, 199
windex データベース, 60
winsch ライブラリルーチン, 363
winsertln ライブラリルーチン, 363
wmove ライブラリルーチン, 364
wnoutrefresh ライブラリルーチン, 364
wprintw ライブラリルーチン, 364
wrefresh ライブラリルーチン, 364
writev システムコール, 279
write コマンド, 247
write システムコール, 278
wscanw ライブラリルーチン, 365
wsetscreg ライブラリルーチン, 365
wstandend ライブラリルーチン, 365
wstandout ライブラリルーチン, 366
WTERMSIG system call, 275
w コマンド, 247

X

xargs コマンド, 247
xcrypt ライブラリルーチン, 366
xdecrypt ライブラリルーチン, 366
xdrmem_create ライブラリルーチン, 368
xdrrec_create ライブラリルーチン, 368
xdrrec_endofrecord ライブラリルーチン, 368
xdrrec_eof ライブラリルーチン, 368
xdrrec_skiprecord ライブラリルーチン, 368
xdrstdio_create ライブラリルーチン, 368
xdr_accepted_reply ライブラリルーチン, 366
xdr_array ライブラリルーチン, 366
xdr_authsys_parms ライブラリルーチン, 366
xdr_authunix_parms ライブラリルーチン, 366
xdr_bool ライブラリルーチン, 367
xdr_bytes ライブラリルーチン, 367
xdr_callhdr ライブラリルーチン, 367
xdr_callmsg ライブラリルーチン, 367
xdr_char ライブラリルーチン, 367
xdr_destroy ライブラリルーチン, 367
xdr_double ライブラリルーチン, 367

xdr_enum ライブラリルーチン, 367
xdr_float ライブラリルーチン, 367
xdr_free ライブラリルーチン, 367
xdr_getpos ライブラリルーチン, 367
xdr_inline ライブラリルーチン, 367
xdr_int ライブラリルーチン, 367
xdr_long ライブラリルーチン, 367
xdr_opaque ライブラリルーチン, 367
xdr_pointer ライブラリルーチン, 367
xdr_reference ライブラリルーチン, 367
xdr_setpos ライブラリルーチン, 367
xdr_short ライブラリルーチン, 367
xdr_string ライブラリルーチン, 367
xdr_union ライブラリルーチン, 368
xdr_u_char ライブラリルーチン, 367
xdr_u_int ライブラリルーチン, 367
xdr_u_long ライブラリルーチン, 367
xdr_u_short ライブラリルーチン, 368
xdr_vector ライブラリルーチン, 368
xdr_void ライブラリルーチン, 368
xdr_wrapstring ライブラリルーチン, 368
XENIX, Solaris SPARC リリース, 11
XENIX セマフォファイルシステム
 (xnamefs), 86
xgettext コマンド, 184
xget コマンド, 247
xlock コマンド (OpenWindows), 221
xnamefs, 86
 /xpg2bin ディレクトリ, 96
 /xpg2include ディレクトリ, 96
xsend コマンド, 247
xset コマンド (OpenWindows), 237
xstr コマンド, 247
xtab システムファイル, 376, 387
xtom ライブラリルーチン, 368
XView File Chooser、OpenWindows 開発者
 ガイド File Chooser, 55
XView Window Toolkit, 175
xv_get_sel コマンド, 215

Y

y0 ライブラリルーチン, 368
y1 ライブラリルーチン, 369
yaccpar コマンド, 178
yacc コマンド, 178, 247
yes コマンド, 248

yn ライブラリルーチン, 369
ypaliases システムファイル, 376
ypbatchupd コマンド, 248
ypbind コマンド, 248, 387
ypcat コマンド, 248
ypfiles データベース, 376
ypgroup システムファイル, 376
ypinit コマンド, 248
ypmatch コマンド, 248
yppasswd コマンド, 248
yppasswd システムファイル, 376
yppoll コマンド, 248
ypprintcap データベース, 376
yppush コマンド, 248
ypserv コマンド, 248
ypset コマンド, 248
ypupdated コマンド, 248
ypwhich コマンド, 248
ypxfrd コマンド, 248
ypxfr コマンド, 248
yp データベースディレクトリ, 381

Z

zcat コマンド, 248
zdump コマンド, 249
zic コマンド, 249
/zoneinfo ディレクトリ, 384

あ

アーカイブ、ランダムライブラリに変換する, 177
アーキテクチャ独立型データベース, 383
アカウントティングファイル
 UUCP, 134
アカウントシステム, xv
アジア系言語の文字セット, 183
アドミンツール
 描写, 24
アドミンツール コマンド, 24
アプリケーション
 移植、理由, 42
 互換性の判断, 172, 173
アプリケーションの多重度, 10

アプリケーションを移植する、理由, 42
アンバンドル言語製品のマウントポイント
とインストールポイント, 379
アンバンドルのソフトウェア, 81
アンバンドルのソフトウェア用のファイルシ
ステム, 81, 83
アンロード, デバイスドライバ, 188

い

移植性, 5
一時ファイルシステム
転送, 34
一時ファイル用のシステム
スプールされた一時ファイル用のディレ
クトリ, 380
ブート処理中に消去されない一時ファ
イル用のディレクトリ, 380
一覧表示
ソフトウェアパッケージのインストー
ル, 15
ファイル名と統計, 86
イメージコピー, ファイルシステム, 87
イメージツール, 4, 7
イメージフィルタ、 unsupported, 26
インクルードヘッダファイル, 381
印刷サブシステム
描写, 26
インスタンスデバイス名, 194
インスタンスのデバイス名, 193
インストール, 29, 39, 40
sソフトウェアパッケージ, 14
インストール後の段階, 29, 36, 40
インストール段階, 29
インストールの段階, 36
インストール前の段階, 31
概要, 9, 29
新機能, 29, 30
ソフトウェアパッケージ, 15, 382

インストール前の段階

概要, 29
ディスクパーティション情報, 31
ディスクパーティション情報の保
存, 32
ネットワークのインストール順序, 35
バックアップファイルシステム, 31
ファイルシステム情報の保存, 32
ファイルシステムのバックアップ, 36
ファイルと保存するファイルのリス
トを作成する, 33
保存するシステムコンポーネントの
リストを作成する, 33
メタデバイス構成情報, 32, 33
インターネットサービス
構成情報ディレクトリ, 91, 378
セキュリティ, 46
ネットワークサービスのディレクト
リ, 382

う

ウィンドウ管理サービス, 7
ウィンドウシステム、 デフォルト, 55

え

エラー メッセージ
ファイルシステム コマンド, 87
ボリュームマネージャ, 70
遠隔手続き呼び出し、 管理, 28
エンドユーザシステムサポート, 14

お

オブジェクトコード逆アセンブラ,
COFF用, 178
オプションパッケージ, 378, 380

か

/カーネル, 19, 81
MT (マルチスレッド), 10
アーキテクチャ固有, 30
カスタマイズ, 74
構成, 20
デバッグする, 161
動的, 9, 20, 187, 188

- 名前, 19
- ブートと, 74
- モジュール, 19
- レイアウト, 21, 188
- モジュール
 - アンロード, 20
 - アンロードする, 188
 - ディレクトリ検索パス, 21
 - ディレクトリの検索パス, 188
 - 配置, 19, 21, 30
 - レポートする情報, 187
 - ロード, 20
 - ロードする, 188
- /カーネルディレクトリ, 21
- 開発者, 149
 - Solaris 機能, 9, 10
 - tools and resources, 163
 - 移行情報の概要, 149
 - 国際化, 182, 185
 - コンパイラ, 151
 - システム構成, 187, 191
 - ツールと資源, 178
 - デバイスの構成, 191, 194
 - デバッグ, 86, 160
 - ネットワーク機能, 181, 182
 - バイナリ互換パッケージ, 43, 172, 173
 - リンカ, 152, 160
- 開発者システムサポート, 14
- 外部ネームサービス要求用エントリポイント, 383
- 拡張 UNIX コード (EUC), 183
- カスタマイズ
 - man コマンド検索パス, 59, 60
 - カーネル, 74
 - ユーザ環境, 53, 54
- カスタマイズする
 - ポートモニタ, 126
- 仮想ファイルシステムアーキテクチャ (VFS), 83, 89
- カット&ペースト, 7
- カレンダーマネージャ, 4, 7
- 環境
 - SVR4 vs., 3
- 環境初期設定ファイル
 - カスタマイズユーザ環境, 54
- 管理ツール, xv
 - ファイル ディレクトリ, 96

- き
- キーボード配置の記述テーブル, 383
- 疑似端末(pty) スレーブデバイス, ディレクトリ, 90
- 疑似端末スレーブデバイス, 377
- 疑似デバイスドライバモジュール用のディレクトリ, 97
- 疑似ファイルシステム, 80, 81, 84, 85
- 起動, 71, 75
- 共通デスクトップ環境(CDE)
 - 開発の歴史, 141
 - セッションマネージャ
 - ウィンドウマネージャ, 142
- 共用オブジェクト, 169
- 共用可能な資源のリスト, 387
- 共用ファイルシステム
 - SunOS リリース 4.x ディスクレスククライアント用のSolaris 2.6サーバ, 111
- 説明, 102
- マウント, 99
- 共用ライブラリ
 - version numbering, 168
 - 作成する, 155, 158
 - の変更, 168
 - バージョン番号, 157, 158

- く
- クイックリファレンステーブル, 388, 385
- クライアント, xv
 - ディスクレス, 111, 119
- クラスタ, 14, 30
- グラフィカルユーザインタフェース (GUI), xv
 - Admintool, 24
 - アドミンツール, 24
 - グラフィカルユーザインタフェース (GUI)を採用しています。 , 4
 - ソフトウェアマネージャ, 14, 15
- グラフィカルユーザインタフェース (GUIs)
 - XView Windows Toolkit, 175
- グラフィックライブラリ, 7
- グループ
 - 管理, 56
- グループ管理, 9, 55
- グループ生産性ツール, 4

け

ゲームのバイナリとデータ, 381

検査

ポートモニタの状態, 126

検索

NIS+ テーブル, 25

用のファイル, 102

検索パス

kernel モジュール, 21

カーネルモジュール, 188

リンカ, 156, 157

検査する

ファイルシステム, 86

プリンタの状態, 122

こ

コアシステムサポート, 14

構成, xv

カーネル, 20

自動構成, 9

新機能, 29, 30

構文, ファイルシステム コマンド, 87, 88

互換

SunOS 4.x と SunOS 5.7, 41, 54

互換性

アプリケーションの判断, 172, 173

相互ハードウェア, 4, 6

互換パッケージ, 41

国際化, 182, 185, 382

個人生産性ツール, 4, 7

コマンド, xv

NFS, 130

Service Access Facility (SAF), 126

local, 383

table of, 249

クイックリファレンス, 386

クイックリファレンステーブル, 385

サービスアクセス機能 (SAF), 27, 28

テーブル, 195

ファイルシステム, 86, 89, 98

プリンタ, 123, 124

変更, xv, 41

ボリュームマネージャ, 70

メッセージ化, 184

コマンド行ユーティリティソフトウェア

パッケージ管理, 15

コマンドツール (OpenWindows), 205

コマンドラインインタフェース
ユーザとグループの管理, 55

コマンドリファレンステーブル, 195, 249

コマンドログファイル(UUCP), 134

コマンドを確認, 123

コミュニティサービスメッセージ, 380

コメントセクションオブジェクトファイ
ル, 178

コンパイラ, 151, 178

さ

サードパーティのソフトウェア用のファイル
システム, 81, 83

サービス, xv

サービスアクセス管理 (SAC), 27

サービスアクセス機能 (SAF), 27

サービスアクセス管理 (SAC), 27

コマンド, 27

サービスアクセス管理 (SAC), 28

描写, 27

再構成ブート, 191

サイト固有のコマンド, 383

削除

NIS+ 情報, 25

ソフトウェアパッケージ, 14, 15

ネットワークデバイス, 27

削除する

ソフトウェアパッケージ, 175

デバイス, 191

ポートモニタ, 126

ポートモニタサービス, 127

作成する

ファイルシステム, 87, 103

し

シエル, xv

機能, 52, 59

初期設定, 53

制限付きシエル, 47

デフォルトの選択, 51, 52

デフォルトホームディレクトリの起動
ファイル, 54, 59

シエルツール (OpenWindows), 238

時間帯情報, 384

資源の制限, 168, 171
システムアカウント
ユーティリティ概要, 8
システム管理関連のディレクトリ, 383
システム管理者, xv
Solaris 機能, 8, 9
セキュリティ手続き, 45
システム構成, xv
データを復元, 40
デフォルト, 91
の変更, 187, 191
システム構成情報のアカウント
directories for, 378, 379
システムコールのリファレンステーブル, 279
システムコールリファレンステーブル, 251
システム情報ユーティリティ, 8
システム動作レポートパッケージ, 382
システムファイルリファレンステーブル, 371, 376
実行可能ファイル作成, 155
実行状態の変更操作のディレクトリ, 91
自動構成, 9
自動セキュリティ拡張ツール (ASET), 9, 49
自動マウント, 21, 99 - 101
自動ロード、デバイスドライバ, 10
シャットダウン
fastboot コマンド, 72, 78, 213
fasthalt コマンド, 76, 78, 213
halt コマンド, 72, 76, 78, 216
reboot コマンド, 72, 76, 78, 232
shutdown コマンド, 72, 76, 77
shutdown システムコール, 272
シャドウファイル, 95
シャットダウン
fasthalt コマンド, 74
順序、Solaris 2.6 ネットワークのインストール, 35
状態監視情報, 379
状態ファイル, 381
商標ファイル, 379
情報の報告
ソフトウェアパッケージ, 15
ディスクデバイス, 66, 67
デバイス管理, 8
デバイスの管理, 66, 67
情報報告, xv
初期設定 states, 変更する, 76
初期設定状態, 変更, 74

初期設定ファイル
ユーザ環境のカスタマイズ, 53
ジョブの処理順序, 132, 133
シリアルポート管理, 27
シリアルポート、制御, 40
シリンダグループ, 17
シリンダグループマップ, 17
シングルユーザシステム, シャットダウン, 74
シングルユーザの実行レベル, 75, 76

す

スクリプト
FMLI スクリプトが使用するバイナリのディレクトリ, 383
rc, 74 - 76, 80, 90, 95, 190, 232, 378, 388
spell コマンド, 383
アカウント, 382
システム動作レポートパッケージ, 382
新規ユーザアカウント用のデフォルトのプロファイルスクリプト, 379
ソフトウェアパッケージ, 175
パッケージ管理, 383
スケーラビリティ, 6, 10
スタイル・マネージャ, 147
スタイル・マネージャのカスタマイズ, 147
スワップデバイス, デフォルト, 81, 85

せ

正規表現コマンド, 178
制限
資源, 168, 170
生成する
ソフトウェアパッケージ, 174
性能ログファイル (uucp), 134
セキュリティ, xv, 45
ASET, 9, 49
Kerberos, 49
PAM, 50
SunShield, 50
changes in, 200
アンバンドル, 49
概要, 45
機能, 45
制限付きシェル, 47
変更, 45

レベル, 9, 49
セキュリティログファイル (uucp), 134
接続相手関連の外部ネームサービス要求用エ
ントリポイント, 383
全体ディストリビューション, 14

そ

相互運用性, 6
相互互換性, 4
装置の速度, 改善, 85
ソースコード管理システム (SCCS), 172
ソケット, 182
ソフトウェア, xv
 サードパーティとアンバンドル, 81, 83
ソフトウェアグループ, 30

た

大規模な組織、Solaris にとってのメリット, 6
多重度, 10
タブ設定用エスケープシーケンス, 384
端末管理, 27, 125, 127
端末記述ファイル, 384
端末装置, 用の ディレクトリ, 90
端末デバイス, 378

ち

チェック
 ソフトウェアパッケージのインストー
 ル, 15
 ファイルシステム, 104
 プリンタの状態, 385
チェックする
 ソフトウェアパッケージ installation, 175

つ

追加
 NIS+ 情報, 25
 ソフトウェアパッケージ, 14, 175
 デバイス, 67, 68, 191, 192
 ネットワークデバイス, 27
 ポートモニタ, 126
 ポートモニタサービス, 127
ツール(developer), xv
ツール (開発者), 163, 178

Binary 互換パッケージ, 41
OLIT, 175
SCCS, 172
XView, 175
finding, 176, 178
ioctl 要求, 163, 166
make コマンド, 171, 172
ptrace 要求値, 166, 168
ソフトウェアパッケージ, 173, 175
バイナリ互換パッケージ, 42, 172, 173
ライブラリ, 168, 171
ツールキット, 175

て

ディスクスライス, 16
ディスクデバイス
 システムに接続しているディスクの名
 前, 31
 情報の報告, 66, 67
 パーティション情報、保存, 31, 32
 命名規則, 64, 65, 193
 レポートする情報, 86, 101
ディスクドライブ
 ディレクトリ, 90
ディスクレスクライアント, SunOS リリース
 4.xをサポートするSolaris 2.6
 サーバー, 111
ディスクレスファイルシステム, 101
低速装置, 性能の改善, 80, 85
デバイス構成, xv
デバイスドライバ, xv
デバイスネーミング, xv
ディレクトリ, xv, 81
 カーネルモジュール, 21, 188
 監視する, 101
 デフォルト, 82, 83
 名前, 80
 の変更, 80, 89, 92
 ファイルシステムに対するパス名の作
 成, 87
 モジュール, 81
 リモート自動マウント, 21
 リンク時に検索する, 156, 157
データアクセス, 共通, 81
データ型, 拡張された, 10
データの転送

- 復元, 36, 41
- 保存, 31
- データベース
 - NIS+, 380
 - bootparams, 372
 - gettytab, 373
 - hosts, 373
 - inetd.conf, 373
 - locale, 373, 382
 - netmasks, 374
 - networks, 374
 - phones, 374
 - printcap, 40, 80, 122, 123, 374, 387
 - protocols, 374
 - publickey, 374
 - rpc, 374
 - spell コマンド, 382, 383
 - termcap, 80, 375
 - terminfo, 80, 122, 123, 375, 384, 387
 - vgrindefs, 376
 - whatis, 60, 199
 - windex, 60, 199
 - yp, 376, 381
 - アーキテクチャ独立型, 383
 - ソフトウェアパッケージ管理ユーティリティ, 380
 - ソフトウェアパッケージの管理ユーティリティによって管理される, 97
 - プリンタサブシステム, 382
 - プリントキャップ, 26
- テープデバイス
 - raw用のディレクトリ, 90
 - リモートへのバックアップ, 106
- デーモンクイックリファレンス, 387, 388
- デスクトップ
 - 概要, 142
- デスクトップ統合サービス, 7
- デバイス管理
 - 自動ロード, 10
 - 情報の報告, 8
 - デバイスのアンロード, 20
 - デバイスロード, 10
 - デバイスをロードする, 20
 - ボリュームマネージャ, 28
 - 命名規則, 17
- デバイス構成, 40
- デバイスドライバ
 - インタフェース型, 10
 - インタフェース互換性, 10
 - ディレクトリ, 97
- デバイスドライバインタフェース/ドライバカーネルインタフェース (DDI/DKI), 10
- デバイスドライバインタフェース (DDI), 10
- デバイスドライバのアンロード, 21
- デバイスドライバのロード, 10, 20, 188
- デバイスの管理, 63, 70
 - information reporting, 66
 - 情報の報告, 67
 - 性能の改善, 80, 85
 - デバイスの追加, 68, 192
 - デバイスのロード, 188
 - デバイスをアンロード, 188
 - デバイスを削除する, 191
 - デバイスを追加, 67
 - ボリュームマネージャ, 68, 70
 - 命名規則, 63, 65, 191
 - 用のボリューム管理, 81
- デバイスの構成, 91, 191, 194
 - デバイスの追加, 191
 - 命名規則, 194
- デバイスの速度, 改善する, 80
- デバイスのツリーを, 192
- デバイスの物理名, 193
- デバイスの命名
 - 変更, 17
 - インスタンス名, 193, 194
 - 開発者に関する, 191
 - 開発者の関係, 194
 - ディスク, 193
 - 物理名, 193
 - 論理名, 193
- デバイスの論理名, 193
- デバイス名
 - CD-ROMS, 65
 - ディスク, 64, 65
 - 変更, 63, 65
- デバック, xv, 86, 160, 179
- デバッグ (Sun WorkShop), 160
- デフォルト
 - root アクセス, 47
 - ウィンドウシステム, 55
 - シェル, 51, 52
 - 自動マウント, 83
 - スワップデバイス, 81, 85

ディレクトリ, 46, 82, 83, 378
ファイルシステム, 82, 83, 88
ボリューム管理ファイルシステム, 81
用のディレクトリ, 91
テンプレートファイル、シェルの初期設定, 54
テンポラリファイルシステム, xv

と

動的カーネル, xv
 defined, 20
 定義, 9
 定義された, 187
動的なカーネル
 定義された, 188
動的リンク, 10
特殊デバイスファイルシステム(SPECFS), 84
特別デバイス, xv
ドライバカーネルインタフェース (DKI), 10
ドラッグ&ドロップ, 7

な

名前
 カーネル名, 19
 システムに接続しているディスク, 31
 ディレクトリ, 80
 ファイル, 37, 80, 86
 明示的, ファイル記述子を使用してファイ
 ルをオープン, 81
 ライブラリ, 169 - 171
 デバイス
 CD-ROMs, 65
 インスタンス, 193, 194
 開発者の関係, 191, 194
 ディスク, 64, 65, 193
 物理, 193
 変更, 17, 63, 65
 論理, 193

ね

ネームサービススイッチ, 25
ネットワーク
 Solaris 2.6 インストールの注文, 35
 Solaris 7 の機能, 181, 182
 サービス管理, 27, 129, 134

トランスポート独立ネットワークサービ
 スの構成情報, 378

バックアップ, 106
ブート, 189
 ブートする, 74, 190
ネットワークデバイス管理, 27
ネットワークライブラリ, 169

は

バージョン番号, 共用ライブラリ, 157, 158, 168
バイナリ, 383
バイナリ互換パッケージ, 41, 42, 172, 173, 381
パイプファイル用の疑似ファイルシステム, 81
パス名
 ファイルシステムに対して作成する, 87
パス名, xv
パスワード
 location of, 374, 387
 配置, 40, 46
 変更, 46, 48
 有効期限, 46, 47
 パスワード有効期限, 46
バックアップ, xv
 インストール, 31, 36
 の変更, 104, 105
パッケージ
 administering, 383
 optional, 378, 380
 インストール, 14, 15
 開発者と, 173, 175
 互換パッケージ, 41
 コンポーネント, 173, 174
 削除, 14, 15
 削除する, 175
 情報の報告, 15
 スプールされたディレクトリ, 380
 生成する, 174
 追加, 175
 定義, 14
 定義された, 173
 メリット, 14, 30
 レポートする情報, 175
パフォーマンスメータツール
 (OpenWindows), 229
汎用ファイルシステム管理コマンド, 86
汎用ファイルシステムコマンド, 89

ひ

表記上の規則, xix
標準化, 3, 5, 6
標準システムコマンド, 381
標準プロセスクイックリファレンス, 387, 388

ふ

ファイル, xv
環境初期設定, 53, 54
監視する, 101
クイックリファレンス, 387, 388
結合する, 37, 38
システム, 371, 376
名前, 37, 80, 86
パイプ, 用の疑似ファイルシステム, 81
バックアップ, 31, 36, 104
ファイル記述子を使用してファイル
をオープン, 81
ファイルを検索する, 102
復元, 106
変換, 39
変更, 40
保存するファイルのリストを作成する, 33
レポートする情報, 86
ファイル記述子ファイルシステム(FDFS), 81,
84
ファイル記述子を使用してファイルをオープ
ン, 81
ファイルシステム, xv, 34, 80, 82, 377, 381
CD-ROM デバイス, 68
System V, 11, 86
automounting, 201
/etc/vfstab ファイルに指定する, 101
labels, 220
エラー メッセージ, 87
仮想アーキテクチャ(VFS), 83, 89
監視する, 101
疑似, 80, 81, 84, 85
形式の判定, 86
検査する, 86
作成する, 87, 103
サポートされないSVR4, 86
サポートされる形式, 84, 86
実装するカーネルモジュール, 97
自動マウント, 21, 99 - 101
情報の保存, 32

シリンダグループ, 17
チェック, 104
追加, 81
ディスクスライスとパーティション, 16
デバッグ, 86
デフォルト, 82, 83, 88
のイメージコピー, 87
の変更, 80, 81
パス名のリスト generation, 87
バックアップ, 31, 36, 104
復元, 106
フロッピーディスクのデバイス, 68
保存するシステムのリスト, 33
マウント, 21, 39, 87, 99, 101, 225, 265, 386
マウントを解除する, 87, 100
ユーティリティ概要, 8
ラベル, 87, 105
changes in
layout, 377, 384
コマンド
位置, 89
構文, 87, 88
使用, 98
説明, 86, 89
の変更, 98
の変更
位置, 80
名前, 80
変更
名前, 37
フォーマット, 31
マウント
automounting, 201
自動マウント, 21, 99 - 101
マウントテーブル, 387
リモート
自動マウント, 21
デフォルトの形式, 88
マウント, 87, 99
マウントを解除する, 87
ファイルシステムの監視, 101
ファイルとファイルシステムの復元, 36
ファイルマネージャ, 4, 28, 68
フィルタ、イメージ、unsupported, 26
ブート, xv, 71, 75, 189, 190
PROMからの, 73
from PROMからの, 73

- の変更, 71, 73, 189, 190
- ブートする
 - の変更, 74
- ブートファイルシステム(BFS), 86
- ブートブロック, 73, 189, 190
- フォント
 - デバイスに依存しない, 57
- 復元, 40
 - SunOS リリース 4.x システムデータ, 40
 - ファイルシステム, 106
- プリンタ, 121, 124
 - status チェック, 385
 - 印刷ジョブの中止, 26
 - 印刷ジョブをキャンセルする, 202, 385
 - 印刷ジョブを取り消す, 123
 - 構成, 40, 91
 - コマンド, xv
 - 状態を検査する, 122
 - スプールファイルディレクトリ, 380
 - デーモンファイル, 387
 - ネットワーク要求の管理と問題の解決, 27
 - の変更, 122
 - 変更, 26
 - setting up
 - コマンド, 385
 - コマンド
 - 使用, 122
 - 使用する, 124
 - 変更, 26
 - 設定する
 - コマンド, 122, 124
- プリンタサブシステム
 - backend executables directory, 382
 - 構成情報ディレクトリ, 378
 - ログ情報ディレクトリ, 380
- プリントサブシステム
 - database ディレクトリ, 382
- プログラムを静的にリンクしたディレクトリ, 383
- プロセス間通信ユーティリティ, 8
- プロセス管理, xv
 - ユーティリティ, 8
- プロセスの管理
 - 疑似ファイルシステム, 81
- ブロックディスクデバイス, 377
- ブロックディスクデバイス用ディレクトリ, 90, 193
- フロッピーディスク装置、管理, 68

- フロッピーディスクデバイス、管理, 28
- フロッピーディスクのファイルシステム、自動的, 68
- フロッピーデバイス、管理, 70
- プロパティウィンド (OpenWindows), 205, 207, 223, 237
- プロパティシート (OpenWindows), 218
- プロファイル ライブラリ, 382
- 分散ファイルシステム(DFS) の管理, 130
- 文書ツールの使用, 57, 58

へ

- 変換, xv
 - ファイル, 39
- 変換する
 - アーカイブ からランダムライブラリ, 177
- 変更, xv
 - NIS+ 情報, 25
 - シェル, 51, 52
 - システム 実行レベル, 75
 - パスワード, 46, 48
 - ファイル, 40
- 変更する
 - システム実行レベル, 76

ほ

- ポイントツーポイント・プロトコル (PPP), 131
- ポートモニタ
 - 管理, 27, 125, 126
 - 説明, 126
- ポートモニタサービス管理, 27, 125, 127
- ポートモニタサービスを使用可能にする, 127
- ポートモニタサービスを使用不可能にする, 127
- ホスト管理者, xv
- 保存, xv
 - ディスクパーティション情報, 31, 32
 - ファイルシステム情報, 32
 - メタデバイス構成情報, 32, 33
- ボリューム管理
 - 用のデフォルトのファイルシステム, 81
- ボリュームマネージャ
 - エラーメッセージ, 70
 - 概要, 28, 68
 - 構成ファイル, 70

コマンド, 70
使用, 68, 70

ま

マウント

ファイルシステム, 21, 39, 87, 101, 225,
265, 386

マウントされている資源のリスト, 387

リモートリソース, 87, 99

ファイルシステム

automounting, 201

自動マウント, 21, 100, 101

マウントテーブル, 387

マウント する

ファイルシステム, 99

ファイルシステム

自動マウント, 99

マウントを解除する

ファイルシステム, 87, 100

リモートリソース, 87

マクロ

kadb, 161

nroff, 384

troff, 384

マニュアルページ

whatis database, 199

構成の変更, 58

マルチ OS オペレーション, 112

マルチスレッド (MT) カーネル, 10

マルチメディアメール, 4, 5

マルチユーザシステム, シャットダウン, 76, 77

マルチユーザの実行レベル, 75

め

メールボックススプールディレクトリ, 23

メール

使用, 56, 57

マルチメディアメール, 4, 5

メールの管理, 23

メタデバイス構成情報, 32, 33

メッセージ, spoken, 5

メッセージカタログ, 183

メニュー、sysadm メニューユーティリ

ティ, 11

も

文字セット, 183

文字列, ソースファイルから抽出する, 178

モデム管理, 27, 125, 127

問題の解決

TTY デバイス, 27

サービスアクセス管理 (SAC), 27

ネットワークから要求された印刷サービ
ス, 27

ゆ

ユーアカウント管理者, xv

ユーザ、Solaris 機能, 8

ユーザアカウント, xv

ユーザアカウントマネージャ (管理ツール)

パスワードの変更, 48

ユーザー環境の管理

デフォルトシェルの選択, 52

ユーザアクセス制御, 56

ユーザ環境管理, xv

SunOS 4.x 操作環境を Solaris 7 で使用, 41

ユーザ環境の管理, 51, 61

Solaris 7 での SunOS 4.x 作業環境の使
用, 54

man コマンド検索パス構成, 60

man コマンド検索パスのカスタマイズ, 59

ウィンドウシステム, 55

デフォルトシェルの選択, 51

文書ツールの使用, 57, 58

マニュアルページディレクトリ構成の変
更, 59

マニュアルページの構成の変更, 58

メールの使用, 56, 57

ユーザとグループの管理, 55

ユーザ管理ユーティリティ, 9

ユーザとグループ追加, 55

ユーザの管理とグループ, 55

ら

ライブラリ

changes in, 168

lint, 151

アーカイブをランダムに変換する, 177

共用, 168

検索パスの規則, 156

- 検索パスの指定, 156
- 資源の制限, 168, 170
- 動的リンク, 10
- 名前と位置, 169 - 171
- ネットワーク, 169
- の変更, 171
- プロファイル, 382
- ルーチンの表, 281, 369
- 共用
 - 構築する, 158
 - 作成する, 155
 - バージョン番号, 157, 158
- ライブラリルーチンリファレンス, 281
- ライブラリルーチンリファレンス表, 369
- ラストイメージフィルタ, 26

り

- リアルタイムの優先スケジュール, 5
- リスト
 - shared ファイルシステム, 387
 - 共用可能な資源, 387
 - マウントされている資源, 387
- リストを作成する
 - ファイルシステムを保存する, 33
- リモート CD-ROM デバイス, install4x プログラム と, 114, 115
- リモートシステム
 - ソフトウェアパッケージ管理, 14, 15
- リモートテープドライブにバックアップ, 106
- リモートファイルシステム
 - 自動マウント, 21
 - デフォルトの形式, 88
 - マウント, 87, 99
 - マウントを解除する, 87
- リンカ, 152, 160
 - 共用ライブラリを作成する, 155, 158
 - 実行可能ファイルを作成, 155
 - 動的リング, 10
 - バージョン番号, 157, 158
 - ライブラリの検索パスの規則, 156
 - ライブラリの検索パスの指定, 156
 - リンクエディタオプションの相違, 152, 155
 - 例, 158, 160

る

- ルートファイルシステム(/), 82
- ループバックファイルシステム(LOFS), 84

れ

- レポートする情報
 - カーネルモジュール, 187
 - ソフトウェアパッケージ, 175
 - ディスクデバイス, 86, 101
 - ファイルシステム, 86, 104

ろ

- ログイン、管理, 28, 46
- ログインシェル
 - 機能, 52, 59
 - 初期設定ファイル, 53
 - 制限付きシェル, 47
 - デフォルトの選択, 51, 52
 - デフォルトホームディレクトリの起動ファイル, 54, 59

- ログファイル
 - uucp, 134
- ロック
 - mutexes, 161

わ

- ワークスペースプログラムサブメニュー, accessing, 147
- ワークスペースマネージャ, 7

オ

- オペレーティング環境
 - 32 ビット, 10
 - 64 ビット, 10

デ

- デバイス構成ライブラリ, 10

パ

- バッチの管理, 15

ユ

ユーザアカウント、追加, 56