



Solaris 移行ガイド

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

Part Number 806-2724-10
2000年3月

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

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。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コ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, docs.sun.com, AnswerBook, AnswerBook2, DeskSet, Direct Xlib, Java, NFS, ONC, OpenWindows, PostScript, Solaris PEX, Solaris JumpStart, Solstice AdminSuite, Solstice AdminTools, Solstice AutoClient, Solstice DiskSuite, SPARC, SPARCstation, Sun-3, SunOS, SunSHIELD, Sun Workshop, ToolTalk, UltraSPARC, WebNFS, XView, SunView, XGL, XIL 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。

サンのロゴマークおよび 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、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. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

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。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政省が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

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

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

DtComboBox 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。(© 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、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: 805-6331-10

Revision A



目次

- はじめに 13
- 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 23
 - Solaris オペレーティング環境に移行するメリット 23
 - 移植性、スケーラビリティ、相互運用性、互換性 25
 - 大規模な組織にとってのメリット 26
 - SVR4 と Solaris オペレーティング環境の比較 26
 - Solaris オペレーティング環境で追加された機能 26
 - Solaris オペレーティング環境に含まれない SVR4 の機能 31
- 2. 主な変更について 33
 - ソフトウェアパッケージとクラスタ 34
 - パッケージの管理 34
 - パッチの管理 35
 - ディスクスライス (またはディスクパーティション) 36
 - シリンダグループ 37
 - デバイスの命名 37
 - ファイルシステム 38
 - ファイルシステムの位置と名前の変更 38
 - 疑似ファイルシステム 39

	追加されたファイルシステム	39
	除去されたファイルシステム	40
	カーネルの構成	40
	カーネルのレイアウト	41
	自動マウント	41
	メールの管理	43
	Admintool	44
	ネットワーク情報サービスプラス (NIS+)	45
	印刷サブシステム	45
	印刷ツール	46
	コマンドの変更	46
	サービスアクセス機能	47
	ボリュームマネージャ	48
3.	SunOS 4 システムから Solaris 7 オペレーティング環境への変換	49
	インストールに関する変更点	49
	Solaris をインストールする前に行うこと	50
	ディスクパーティション情報の保存	51
	ファイルシステム情報の保存	52
	メタデバイス構成情報の保存	52
	バックアップ内容の決定	53
	ディスク空間の所要量の決定	55
	ネットワークのインストール順序の決定	55
	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	55
	Solaris ソフトウェアのインストール	56
	保存オプション	56
	インストール後のファイルとファイルシステムの復元	56
	SunOS 4 ファイルシステムとユーザファイルの復元	56
	SunOS 4 システム構成ファイルの復元	57

4.	互換パッケージの使用方法	61
	アプリケーションを移植する理由	61
	SunOS/BSD ソース互換パッケージ	62
	バイナリ互換パッケージ	62
	SunOS 4 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	63
5.	セキュリティ	65
	Solaris 7 のセキュリティ機能	65
	/etc/passwd と /etc/shadow ファイル	66
	/etc/default ファイル	66
	制限付きシェル	67
	パスワード有効期限の変更	67
	アクセス制御リスト (ACL)	68
	自動セキュリティ拡張ツール (ASET)	69
	セキュリティオプション	69
	Kerberos 4.0 セキュリティ	69
	SunSHIELD パッケージ	70
	PAM	70
6.	ユーザ環境の管理	71
	デフォルトシェルの選択	71
	ユーザ環境のカスタマイズ	73
	Solaris ソフトウェアでの SunOS 4 作業環境の使用	74
	ウィンドウシステム	74
	ユーザとグループの管理	75
	ユーザとグループ管理の選択	75
	ユーザアカウントの追加	76
	メールの使用	76
	文書ツールの使用	77

マニュアルページの構成の変更	78
man コマンド検索パスのカスタマイズ	79
whatis データベースと windex データベース	80
man コマンドの使用	81
7. デバイスの管理	83
デバイス命名規則	83
ディスクに関する規則	84
テープドライブに関する規則	85
ディスク情報の表示	85
df コマンド	86
du コマンド	86
dtkinfo コマンド	86
devinfo コマンド	87
デバイスをシステムへ追加する	87
動的再構成	88
ボリュームマネージャの使用	88
8. 起動とシャットダウン	91
ブート	91
boot コマンドの変更	92
PROM からのブート	93
ブート処理の相違	93
init コマンドの使用	94
init コマンドの変更	94
システム実行レベルの変更	95
シャットダウン	96
shutdown コマンドの変更	97
fasthalt コマンドと fastboot コマンドの使用	98
halt コマンドと reboot コマンドの使用	98

9.	ファイルシステムの管理	99
	ファイルシステムの変更	99
	疑似ファイルシステム	100
	ファイルシステムの追加	101
	デフォルトのファイルシステムとディレクトリ	101
	仮想ファイルシステムアーキテクチャ	103
	サポートされるファイルシステムの形式	104
	サポートされない SVR4 ファイルシステムの形式	106
	汎用ファイルシステムコマンド	106
	ディレクトリとファイルの変更	110
	/dev ディレクトリ	110
	/etc ディレクトリ	110
	/sbin ディレクトリ	115
	/usr ディレクトリ	115
	/var ディレクトリ	116
	/kernel ディレクトリ	117
	/opt ディレクトリ	118
	/sys ディレクトリ	118
	ファイルシステム管理コマンドの使用	118
	ファイルシステムのマウントと autofs	119
	ファイルシステムの監視	121
	ファイルシステムの共用	122
	新しいファイルシステムの作成	123
	ファイルシステムのチェック	124
	ファイルのバックアップと復元	124
	UFS ログ	129
10.	SunOS 4 ディスクレスクライアントをサポートする Solaris 7 サーバの設定	131

- SunOS 4 をサポートする機能を Solaris 7 サーバへ追加する 131
 - discover4x の実行 133
 - CD-ROM ドライブを install4x 用に設定する 133
 - install4x の実行 135
 - convert4x の実行 138
- 11. プリンタ、端末、モデムの管理 141**
 - 印刷 141
 - 印刷の変更点の要約 141
 - 印刷コマンドと互換パッケージ 142
 - プリンタコマンドの使用 142
 - SunOS 5.7 プリンタ管理コマンドの使用 143
 - シリアルポート管理 144
 - 端末とモデムの管理 145
 - サービスアクセス機能 (SAF) 145
- 12. ネットワークサービスの管理 149**
 - TCP/IP の変更 149
 - TCP と SACK 150
 - NFS の変更 150
 - PPP 151
 - LDAP 151
 - IIIMP 151
 - UUCP 151
 - チェックポイントからの再起動 153
 - ユーザジョブの処理順序 153
 - Limits ファイル 154
 - Config ファイル 154
 - ログファイル 154
- 13. ネームサービスの使用方法 155**

	ネームサービススイッチ	156
	NIS+	156
	DNS	156
	DNS と NIS+ の比較	156
	NIS と NIS+ の比較	157
	NIS+ アップグレードの計画	159
14.	Solaris 共通デスクトップ環境	161
	Solaris 共通デスクトップ環境について	161
	開発者、エンドユーザ、および CDE	162
	デスクトップの概要	162
	フロントパネル	162
	スタイル・マネージャ	163
	ファイル・マネージャ	164
	OpenWindows 環境から CDE への移行	165
	デスクトップサービス	165
	CDE でのウィンドウ、メニュー、ボタンおよびマウスの使用	166
	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	166
	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	166
	CDE での OpenWindows アプリケーションの実行	166
	アプリケーションの設定とプロパティ	167
	キーボードのデフォルト値の変更	167
	マウスのデフォルト値の変更	167
	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	171
	コンパイラ	171
	リンカ	172
	リンクエディタオプションの相違	172
	共用ライブラリの作成	175

- 実行可能ファイルの作成 175
- ライブラリ検索パスの指定 176
- 検索パスの規則 176
- バージョン番号 177
- 例 178
- デバッガ 180
 - dbx と dbxtool 180
 - adb と kadb 180
 - kadb マクロ 181
 - 動作中のカーネルのデバッグ 182
 - truss コマンド 182
- 16. ツールとリソース 185**
 - ioctl() 要求 185
 - ptrace() 要求値 188
 - ライブラリ 190
 - 再編成ライブラリ 190
 - 共用ライブラリ 190
 - リソースの制限 190
 - make の使用 193
 - SCCS の使用 194
 - アプリケーション互換性の判断 194
 - アプリケーションパッケージ作成 195
 - パッケージ作成ユーティリティ 196
 - ツールキット 197
 - OLIT 197
 - XView 197
 - SunOS 4 ツールの検索 198
- 17. ネットワークと国際化機能 203**

ネットワーク	203
NIS と NIS+	203
nsswitch.conf ファイル	204
Network Interface Tap	204
ソケット	204
国際化	204
文字サポート	205
メッセージカタログ	205
ロケールデータベース	206
コマンド	206
ライブラリ	206
18. システムとデバイスの構成	209
システム構成	209
動的にロードされるカーネル	209
カーネルの配置	210
config コマンド	210
/etc/system ファイル	210
boot コマンド	211
ブートの相違点の要約	212
再構成ブート	213
開発者に関するデバイスの命名規則	213
/devices	213
/dev	214
デバイスドライバの命名規則	214
19. デバイスドライバと STREAMS	217
デバイスドライバと STREAMS デバイスドライバ	217
デバイスドライバのインタフェース	217
devinfo コマンド	220

	移行に関する留意点	221
	STREAMS	223
	Solaris 2.x ドライバアーキテクチャ	224
	デバイスドライバ関連コマンド	225
A.	コマンドリファレンス	227
	リファレンスの使い方	227
	例	228
	コマンドリファレンス	230
B.	システムコールリファレンス	283
	リファレンスの使い方	283
	例	284
	システムコール	285
C.	ライブラリルーチンリファレンス	315
	リファレンスの使い方	315
	例	316
	ライブラリルーチン	317
D.	システムファイルリファレンス	405
	リファレンスの使い方	405
	システムファイル	405
E.	/ と /usr ファイルシステムの変更	415
	/ ファイルシステムの配置	415
	/usr ファイルシステムの配置	419
F.	基本的な変更についてのクイックリファレンス	423
	変更一覧表	423
	用語集	431
	索引	435

はじめに

『Solaris 移行ガイド』は、SunOS™ リリース 4 と SunOS リリース 5.7 オペレーティングシステムの違いを説明するものです。SunOS 4 を使い慣れているユーザを対象に、SunOS 5.7 環境に移行する際に役に立つように作られています。

この版の『Solaris 移行ガイド』は、Solaris™ 8 オペレーティング環境とともに配布され、前バージョンにおける小さな技術的問題や表の表示における問題を修正したものです。

このマニュアルでは、引き続き、SunOS 4 と SunOS 5.7 オペレーティングシステム間の違いを説明しています。SunOS リリース 5.8 特有の変更点については、同じく Solaris 8 オペレーティング環境で配布される『Solaris 移行ガイド (追補)』で説明されています。

対象読者

このマニュアルは、SunOS リリース 4 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 環境と SunOS 5.7 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

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。

内容の紹介

このマニュアルは2つのパートに分かれており、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。

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。

パート I 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 と SunOS 5.7 環境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挙げ、SunOS 4 のデータを Solaris 7 オペレーティング環境に最も容易に移行できるようにしています。
- 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 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、移行が容易になります。
- 第5章では、SunOS 4 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 5.7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、ディレクトリ、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。Solstice Admintools™ と サービスアクセス機能 (SAF) を使用し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。

パート II — 開発者のための移行情報

パート II は、主に開発者やプログラミング環境に関する Solaris 7 の変更点について説明します。

パート II 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リソース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 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。
- 第 19 章では、デバイスドライバのインタフェースにおける変更点、`devinfo` コマンド、移行に関する留意点、STREAMS、Solaris 7 ドライバアーキテクチャ、などのデバイスドライバの問題について説明します。

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 と SunOS 5.7 のコマンドを比較します。
- 付録 B では、SunOS 4 と SunOS 5.7 のシステムコールを比較します。
- 付録 C では、SunOS 4 と SunOS 5.7 のライブラリルーチンを比較します。
- 付録 D では、SunOS 4 と SunOS 5.7 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変更点を示します。
- 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プロセスの変更点に対するクイックリファレンスです。

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) 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、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

<http://docs.sun.com> では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。

表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。

表 P-1 表記上の規則 続く

字体または記号	意味	例
[]	参照する章、節、ボタンやメ ニュー名、強調する単語を示しま す。	第 5 章「衝突の回避」を参照して ください。 この操作ができるのは、「スー パーユーザー」だけです。
\	枠で囲まれたコード例で、テキス トがページ行幅を超える場合に、 継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

ただし AnswerBook2™ では、ユーザー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。

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 y|n [filename]
```

■ Bourne シェルおよび Korn シェルのプロンプト

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

■ スーパーユーザーのプロンプト

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

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

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

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

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

一般規則

- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

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名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れ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)」と記されている場合は、このマニュアルページは、『*man pages section 1 : User Commands*』の中に記載されています。

関連マニュアル

Solaris 7 オペレーティング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 7 インストールライブラリ (SPARC 版)*』
- 『*Solaris のシステム管理 (第 1 巻)*』
- 『*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*』

- 『Source Compatibility Guide』
- 『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』
- 『リンカーとライブラリ』
- 『プログラミングユーティリティ』

Sun 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.com/smcc/solaris-migration/index.html>

Solaris Migration Initiative ホームページは、SunOS リリース 4 からの移行に役立つツール、マニュアル、情報の配布の中心となるポイントです。

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、および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。

- 23ページの「Solaris オペレーティング環境に移行するメリット」
- 26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 5.7 は、UltraSPARC™ システム対応の UNIX System V リリース 4 (SVR4) に基づく 64 ビット Solaris オペレーティング環境と SPARC および IA プラットフォーム対応の 32 ビット Solaris アプリケーション環境です。
- 相互互換性により、SunOS 5.7 は SPARC™ 上だけでなく、IA CPU でも動作します。
- SVR4 や、ONC™ ファミリのネットワークングプロトコルなどの業界標準を採用しています。
- OpenLook 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、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、その他のアプリケーションにアクセスできるフロントパネルで構成されるデスクトップ環境です。
- 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、Web ブラウザインタフェースを採用したオンラインドキュメンテーションです。

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、シェ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。新しい技術が採用されると、既存のソフトウェアへの投資が無駄にならないようにすることが必要になります。

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™、Direct Xlib™、Solaris 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ができるので、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 をサポートしています。
- 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。

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。デバイスカー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 (DDI)、デバイスドライバインタフェース / デバイスカーネルインタフェース (DDI/DKI) 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。
- オペレーティング環境 - 32 ビットアプリケーション開発環境および 32 ビットアプリケーションをサポートし、既存の多くの 32 ビットアプリケーションが動作します。また、64 ビットアプリケーション開発環境および 64 ビットのアプリケーションをサポートし、大きなアドレス空間を処理できる新しい 64 ビットアプリケーションのほか、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ため、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

SunOS リリース 5.7 環境には、SunOS 4 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。

- 34ページの「ソフトウェアパッケージとクラスタ」
- 36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)」
- 37ページの「デバイスの命名」
- 38ページの「ファイルシステム」
- 40ページの「カーネルの構成」
- 41ページの「自動マウント」
- 44ページの「Admintool」
- 45ページの「ネットワーク情報サービスプラス (NIS+)」
- 45ページの「印刷サブシステム」
- 47ページの「サービスアクセス機能」
- 48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、Sun 社製以外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 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 ではディスクパーティションといいます。SunOS 5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 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 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 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。

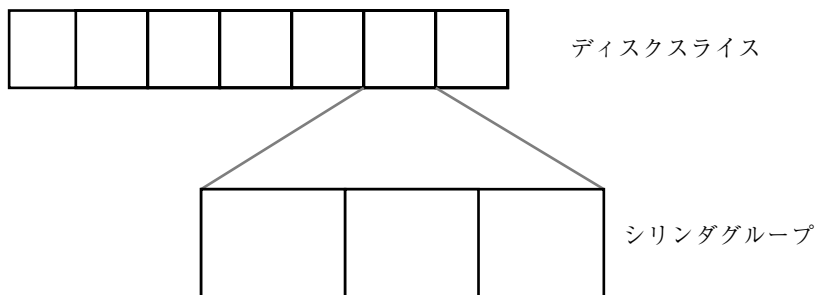


図 2-1 ディスクスライスとシリンダグループ

デバイスの命名

SunOS 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 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 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、83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、213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 5.7 と SunOS 4 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 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。

99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 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 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 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 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 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 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 とは異なり、SunOS 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 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 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 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 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 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 `automountd` 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、SunOS 4 の `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、119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 および SunOS 4 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 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 では、sendmail.mx と呼ばれるプログラムを DNS サイトを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 と SunOS 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、および『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 7 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フェース。Solaris 7 リリースから利用可能
になりました。

詳細は、第 11 章、および『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 Shell Tool ウィンドウで使用しても、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) とラスタ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リソースを制御します。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 88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 システムから Solaris 7 オペレーティング環境への変換

SunOS 4 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 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。

- 49ページの「インストールに関する変更点」
- 50ページの「Solaris をインストールする前に行うこと」
- 55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」
- 56ページの「Solaris ソフトウェアのインストール」
- 56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 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。ローカル CD-ROM ドライブを持たないシステムについては、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 8 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 Solaris 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 8 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

インストール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 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 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の場合や、/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、83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 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™ ユーティリティを使用して SunOS 4.1.1 を実行している場合や、SunOS 4.1.2 システム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 ユーティリティを使用せずに 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、ファイルシステムは、すべてが同じとは限りませんが、上方および下方に互換性があります。

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、`dkinfo(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す。

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、データベース構成の状態の情報がわかります。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 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 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全体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 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 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。リストの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の例として、次のものがあります。

- `./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 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

SunOS 4 システムのネットワークを Solaris 7 ソフトウェアに移行するときは、ユーザが不便にならない状況を想定して、移行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 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 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 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 7 インストールライブラリ (SPARC 版)』または『Solaris 7 インストールライブラリ (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 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 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 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 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 SunOS 4 ファイルとファイルシステムの復元に関する項目を説明します。

SunOS 4 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 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 ユーザファイルはすべて復元し、それを新しいシステムにコピーします。

SunOS 4 システム構成ファイルの復元

最初に、SunOS 4 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らない場合もあります。

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 のシステムで変更されたとは限りません。SunOS 4 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 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 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 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 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 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 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 5.7 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 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 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 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 5.7 は、SunOS 4 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、SunOS リリース 4 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、アプリケーションの移行、および 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 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。

- 61ページの「アプリケーションを移植する理由」
- 62ページの「SunOS/BSD ソース互換パッケージ」
- 62ページの「バイナリ互換パッケージ」

SunOS 4 のコマンドの中には、Solaris 7 では利用できないものもあります。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 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 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 5.7 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 5.7 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、『*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 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 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 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

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 5.7 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用すると、SunOS 4 システムソフトウェアに基づいた「適切に動作する」アプリケーションバイナリは、変更や再コンパイルを行うことなく、SunOS 5.7 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 5.7 ソフトウェアのアプリケーションの開発は、SunOS 5.7 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 の機能と AT&T SVR4 の機能を組み合わせたもので、また新しい環境に特に追加された機能も含まれています。いくつかの SunOS 4 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 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。

- 65ページの「Solaris 7 のセキュリティ機能」
- 67ページの「パスワード有効期限の変更」
- 69ページの「自動セキュリティ拡張ツール (ASET)」
- 69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 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+ セキュリティが強化されています。

『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 5.7 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 5.7 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 5.7 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 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。

- 71ページの「デフォルトシェルの選択」
- 73ページの「ユーザ環境のカスタマイズ」
- 74ページの「ウィンドウシステム」
- 75ページの「ユーザとグループの管理」
- 76ページの「メールの使用」
- 77ページの「文書ツールの使用」
- 78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 5.7 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 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 作業環境の使用

SunOS 4 作業環境を引き続き使用したい場合、SunOS 5.7 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 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 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 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、簡単に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 ソフトウェアは Solaris 7 環境でデスクトップとして使用することもできます。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付くと思います。

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 5.7 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 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す。

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 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 5.7 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 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 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 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、『*OpenWindows Version 3.1 User's Guide*』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 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 の /usr/bin/mail バージョン (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 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 5.7 は SunOS 4 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 5.7 システムでは、T_EX 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 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 5.7 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 5.7 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 5.7 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 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 5.7 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数は、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/
```

```
SunOS5.7% 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 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 5.7 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
```



```
SunOS5.7% 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 5.7 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 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 5.7 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 5.7 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。

- 83ページの「デバイス命名規則」
- 85ページの「ディスク情報の表示」
- 87ページの「デバイスをシステムへ追加する」
- 88ページの「ボリュームマネージャの使用」

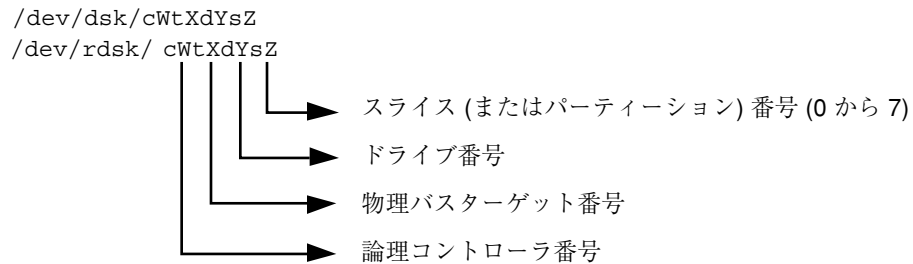
デバイス命名規則

SunOS 5.7 プラットフォームのデバイス命名規則は、SunOS 4 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 5.7 コマンドは、SunOS 5.7 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 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 5.7 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

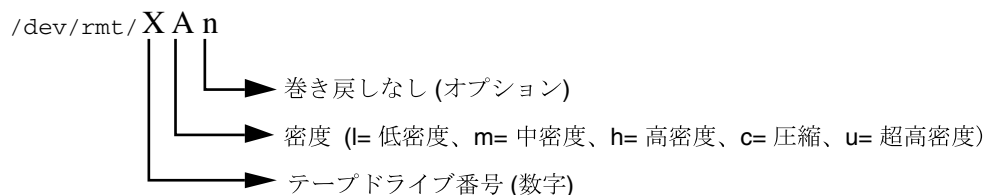


表 7-1 に、SunOS 4 と SunOS 5.7 のデバイス命名規則を比較した例をいくつか示します。

表 7-1 SunOS 4 と SunOS 5.7 のデバイス名

デバイスの種類	SunOS 4 デバイス名	SunOS 5.7 デバイス名
ディスクデバイス	/dev/sd0g	/dev/dsk/c0t3d0s6
	/dev/rsd3b	/dev/rdisk/c0t0d0s1
	/dev/rsd3a	/dev/rdisk/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 5.7 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 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 5.7 コマンドの構文は、SunOS 4 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、103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 5.7 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 5.7 のデバイス命名規則を使用してください。詳細は、83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 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。

例 7-1 は、SunOS 5.7 prtvtoc コマンドの出力例を示します。

例 7-1 SunOS 5.7 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
*
* Partition Tag  Flags  First Sector      Last 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 の devinfo は、SunOS 5.7 とは互換性がありません。SunOS 4 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 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	<code>/cdrom/cdrom_name</code>
フロッピーディスク	<code>/floppy/floppy_name</code>

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	<code>/vol/dev/aliases/cdrom0</code>
フロッピーディスク	<code>/vol/dev/aliases/floppy0</code>

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
<code>rmmount(1)</code>	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 <code>vold</code> によって使用され、 <code>/cdrom</code> および <code>/floppy</code> を自動的にマウントする。
<code>volcancel(1)</code>	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。
<code>volcheck(1)</code>	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、 <code>/dev/diskette</code> によって示されたドライブをチェックする。
<code>volmissing(1)</code>	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。
<code>vold(1)</code>	<code>/etc/vold.conf</code> 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。

- 91ページの「ブート」
- 94ページの「init コマンドの使用」
- 96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。

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され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。
- Sun 社製以外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 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 7 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 の `halt` と `reboot` コマンドは、SunOS 5.7 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 から変更された SunOS 5.7 のコマンドを示します。

表 8-1 SunOS 5.7 でのリブート用コマンドの変更 `fastboot`

SunOS 4 コマンド	SunOS 5.7 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 5.7 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 のブートブロックに似ています。

SunOS 4 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 システムは `/vmunix` を使いました。ただし、SunOS 5.7 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 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 5.7 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 ブート時の相違の概要

SunOS 4	SunOS 5.7	機能
<code>bootsd</code>	<code>bootblk</code>	ディスクから <code>ufsboot</code> をロードする。
<code>boot program</code>	<code>ufsboot</code>	ディスクから <code>unix</code> をロードする。

表 8-2 ブート時の相違の概要 続く

SunOS 4	SunOS 5.7	機能
/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。83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 の `fasthalt` コマンドは、SunOS 5.7 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 5.7 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 5.7 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 5.7 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 5.7 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 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 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 5.7 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 5.7 システム初期設定実行レベル

実行レベル	デフォルトの SunOS 5.7 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。

表 8-3 SunOS 5.7 システム初期設定実行レベル 続く

実行レベル	デフォルトの SunOS 5.7 の機能
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 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 5.7 システムでのみ使用できます。
- SunOS 4 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 5.7 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 5.7 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 から変更された SunOS 5.7 のコマンドを示します。

表 8-4 SunOS 5.7 のシャットダウンの変更 halt、fasthalt

SunOS 4 コマンド	SunOS 5.7 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 5.7 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 5.7 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)。
-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」の応答を想定している。
-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 5.7 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、オペレータの介在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 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 5.7 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。

- 99ページの「ファイルシステムの変更」
- 101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」
- 103ページの「仮想ファイルシステムアーキテクチャ」
- 110ページの「ディレクトリとファイルの変更」
- 118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 5.7 と SunOS 4 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 5.7 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 5.7 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 5.7 には含まれていません。

含まれている疑似ファイルシステムは次のとおりです。

- CACHEFS – CD-ROM のような低速の装置の性能を改善するために使用します。
- PROCFS – プロセス番号により /proc ディレクトリにアクティブなプロセスのリストを保持するメモリ常駐ファイルシステム。proc (4) のマニュアルページを参照してください。
- FDFS – ファイル記述子を使用してファイルをオープンするため明示的なファイル名をリストします。
- FIFOFS – 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 – 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 – デフォルトのスワップデバイスを提供し、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 5.7 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 5.7 のファイルシステムは階層型です。図 9-1 は、SunOS 5.7 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。

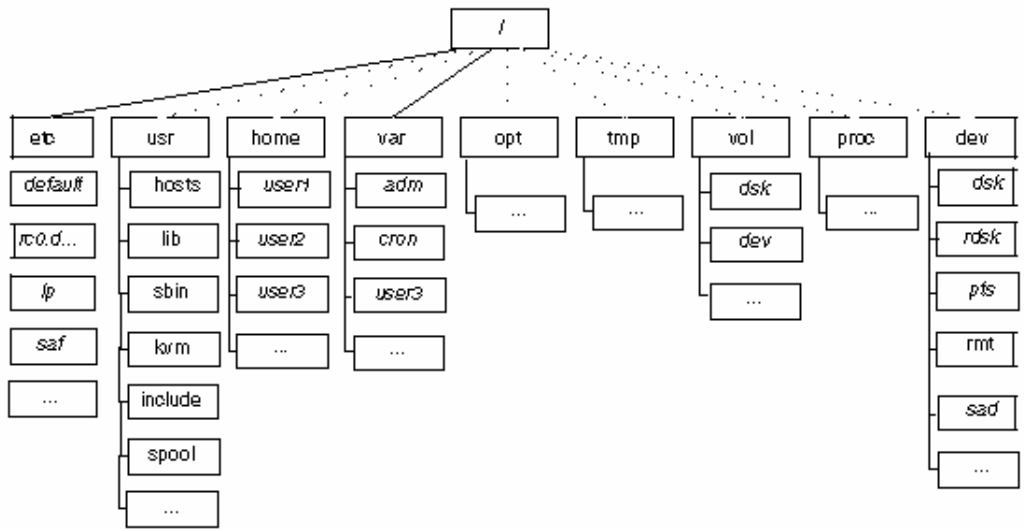


図 9-1 Solaris 7 のデフォルトのファイルシステムとディレクトリ階層

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、vold(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 5.7 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。すべての種類を学ぶのは混乱を招くもとにもなり、また困難です。SunOS 5.7 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 に含まれていたほとんどのファイルシステム形式は、SunOS 5.7 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 5.7 では廃止されました。表 9-2 に、SunOS 4 と SunOS 5.7 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	SunOS 5.7
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

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	SunOS 5.7
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
	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、および『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 5.7 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 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 5.7 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 5.7 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
clri (1M)	i ノードをクリアする。
df (1M)	空きディスクブロック数とファイル数を表示する。
ff (1M)	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。
fsck (1M)	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。
fsdb (1M)	ファイルシステムデバッガ
fstyp (1M)	ファイルシステムの形式を判定する。
labelit (1M)	テープへ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 番号をもつバ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、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、各コマンドに該当するマニュアルページ (たとえば、`mkfs_ufs(4)`) 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 5.7 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 と SunOS リリース 5.7 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 `/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 5.7 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 5.7 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

表 9-8 /etc ディレクトリへの追加 続く

サブディレクトリ	説明
/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 5.7 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、106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 (キャラクタ型) 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/rdsk/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 5.7 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 /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) のマニュアルページを参照のこと。</p>

例 9-1 は SunOS 5.7 /etc/vfstab ファイルの例です。

例 9-1 SunOS 5.7 /etc/vfstab ファイルの例

#device	device	mount	FS	fsck	auto	mount
#to mount	to	point	type	pass	mount?	options
#						
/dev/dsk/c0t0d0s0	/dev/rdsk/c0t0d0s0	/	ufs	1	no	-
/proc	-	/proc	proc	-	no	-
swap	-	/tmp	tmpfs	-	yes	-

(続く)

/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 5.7 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 5.7 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『man pages section 1M: System Administration Commands』を、スクリプトの説明は 95 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 5.7 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 5.7 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 5.7 において SunOS 4 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 における位置	SunOS 5.7 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、通常の実行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 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 で追加されました。

/kernel ディレクトリ

SunOS 5.7 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 では `/sys` ディレクトリにありました。表 9-12 では、SunOS 5.7 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 a.out を実行するためのカーネルモジュール
<code>/kernel/fs</code>	ufs、nfs、proc、fifo 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 5.7 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 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 5.7 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 5.7 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、マップのリソース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い必要があります。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	SunOS 5.7
<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、88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 5.7 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、112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と変更点を示します。

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リソース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 の df コマンドの出力は、SunOS 5.7 の df コマンドとは多少異なる。SunOS 5.7 の -k オプションを使用する出力は、SunOS 4 の -k オプションの出力と似ている。SunOS 4 の df -t filesystem は、指定された形式のファイルを表示する。一方、SunOS 5.7 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 オプションは、SunOS 5.7 コマンドでは使用できない。 <code>-n cpio-device</code> <code>cpio -c</code> 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有

SunOS 4 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 5.7 では、これと同じ概念を「リソースの共有」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有されます。`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、共有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。

自動的に共有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有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リソースを表示します。

```
% 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 5.7 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 5.7 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 (83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 5.7 の mkfs コマンドは、SunOS 4 の mkfs コマンドとは大幅に異なります。SunOS 5.7 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 (106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 5.7 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 5.7 の fsck(1M) コマンドは、SunOS 4 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 (106 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 5.7 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 5.7 のデバイス命名規則の詳細については、83 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 5.7 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 と SunOS 5.7 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 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 の bar ファイルは、

SunOS 5.7 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 の dump コマンドで作成されたファイルは、SunOS 5.7 システム上に ufsrestore で復元できます。

SunOS 5.7 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 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(スタックロード) 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、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。ダンプが完了するか、またはテープやディスクの終わりに達すると、ドライブをオフラインにして、テープを巻き戻すか、ディスクを取り出す。これは、別のプロセスがドライブを使用し、データを上書きし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 5.7 では多少異なります。SunOS 4 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 5.7 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 5.7 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 の規則に従います。

ufsrestore コマンド

SunOS 5.7 の `ufsrestore` コマンドは、SunOS 4 の `restore` コマンドに似ています。SunOS 4 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 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 5.7 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 5.7 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 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 5.7 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 5.7 の *cpio* コマンドは、SunOS 4 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	オプションで使えるコマンド	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 ディスクレス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、SunOS 4 ディスクレスクライアントをサポートするために Solaris 7 サーバを設定する方法を説明します。

次の各節では、Solaris 7 サーバの設定に使用する SUNWhinst パッケージに含まれているプログラムについて記述します。

- 131ページの「SunOS 4 をサポートする機能を Solaris 7 サーバへ追加する」
- 133ページの「discover4x の実行」
- 133ページの「CD-ROM ドライブを install4x 用に設定する」
- 135ページの「install4x の実行」
- 138ページの「convert4x の実行」

注 - SUNWhinst パッケージは、Solaris Easy Access Server 2.0 ソフトウェア CD の、AdminSuite_2.3+AutoClient_2.1/4.x ディレクトリにあります。

SunOS 4 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 ディスクレスクライアントにサービスを提供するよう Solaris 7 システム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 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 クライアントシステムを使用するものもあります。

たとえば、Sun-3™ マシンなどのシステムは、Solaris 2.2 または Solaris 7 ソフトウェアではアップグレードできません。このようなシステムおよび Solaris 7 ネットワーク上のその他の SunOS 4 クライアントに関しては、SUNWhinst パッケージをサーバ上の /export パーティションに追加すると、SunOS 4 クライアントに必要なマルチ OS オペレーションを確立することができます。

SUNWhinst パッケージには、Solaris 7 サーバ上の /export ディレクトリを設定するために実行する 3 つのプログラムが含まれています。この 3 つのプログラムは次の通りです。

- **discover4x** - このプログラムは、サーバに Solaris 7 オペレーティング環境がインストールされた後、SunOS 4 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 クライアントサポートを調べて、Solaris 7 サーバ上に SunOS 4 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 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 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 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に使用します。更新された情報により、既存の SunOS 4 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 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、134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、134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 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、134ページの「リモート CD-ROM ドライブ (SunOS 4) の使用」を参照してください。

SunOS 4 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 オペレーティング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、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、CD をドライブにインストールした後で次のコマンドを実行してください。

```
# 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実行してください。

```
# 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) の使用

`install4x` が SunOS 4 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実行してください。

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

このコマンドを実行したら、`/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実行します。

```
# 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.sun4.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      +-----+
                                | Disk Usage: |
or abort without loading any modules | 0K Selected |
                                | 53634K Free  |
[Q] Quit without loading      +-----+

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
```

(続く)


```

[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、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。

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は、選択文字の右側に **R** と表示されます。一般にロードされるソフトウェアは **C** と表示され、ロードするのが望ましいソフトウェアは **D** と表示されます。

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.sun4.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 modules
or abort without loading any modules:
[Q] Quit without loading

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 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。

- 141ページの「印刷の変更点の要約」
- 142ページの「プリンタコマンドの使用」
- 145ページの「端末とモデムの管理」
- 145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 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 5.7 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 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 5.7 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 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 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 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 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 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 5.7 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 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 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	SunOS 5.7	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	SunOS 5.7	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 5.7 プリンタ管理コマンドの使用

この節では、SunOS 4 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 サービスは、互換モードでも使用できません。

System V プリンタ管理コマンド、lpadmin(1M) と lpsystem(1M)、terminfo データベース、および /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	SunOS 5.7	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
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

表 11-2 プリンタ管理、設定、ファイルの比較 続く

troff による印刷

SunOS 4 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 5.7 の troff コマンドを示します。

表 11-3 SunOS 5.7 の troff コマンド

SunOS 5.7 コマンド	機能
<code>troff file /usr/lib/lp/postscript/dpost lp</code>	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
<code>troff file /usr/lib/lp/postscript/dpost lp -d printer</code>	特定のプリンタに送信する
<code>troff file lp-Ttroff</code>	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 デーモンを制御します。`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ときに、`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。

- 149ページの「TCP/IP の変更」
- 150ページの「NFS の変更」
- 151ページの「PPP」
- 151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 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 5.7 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リソースを共有し、NFS リソースを管理できます。特に、`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 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
- 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 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 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 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 5.7 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 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、伝送はファイルの先頭から再開されます。Solaris 7 UUCP を実行する 2 つのシステムの間通信では、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 5.7 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 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、アカウントリング、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。アカウントリング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 5.7 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 5.7 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 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。

- 156ページの「ネームサービススイッチ」
- 156ページの「NIS+」
- 156ページの「DNS」
- 156ページの「DNS と NIS+ の比較」
- 157ページの「NIS と NIS+ の比較」
- 159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) を取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
ドメイン間のリソースアクセス	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リソース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 5.7
- 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 サーバとともに動作する SunOS 5.7

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 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。

- 161ページの「Solaris 共通デスクトップ環境について」
- 162ページの「デスクトップの概要」
- 165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

共通デスクトップ環境 (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一つです (もう一つは OpenWindows デスクトップ)。CDE は、Sun、Hewlett-Packard、IBM、Novell その他多くの UNIX ワークステーション市場の標準デスクトップとして使用されるよう設計されています。Sun 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 Sun、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 1 つの標準セットを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、ユーザが必要とするツールを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。

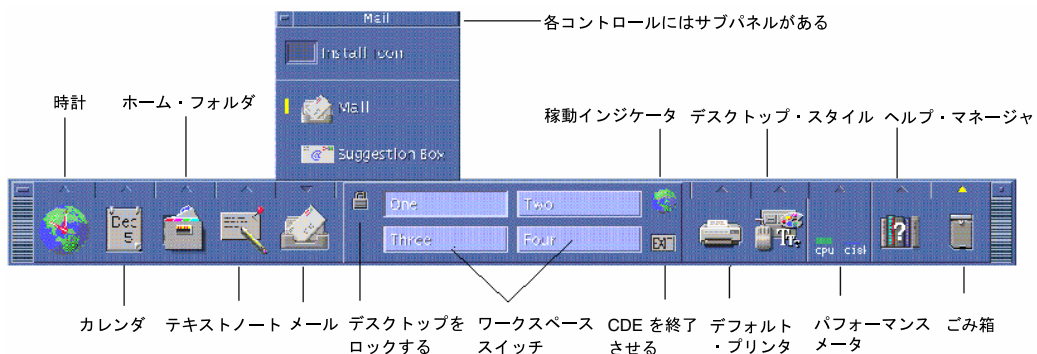


図 14-1 フロントパネルの各種コントロール

上記のフロントパネルの図では、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



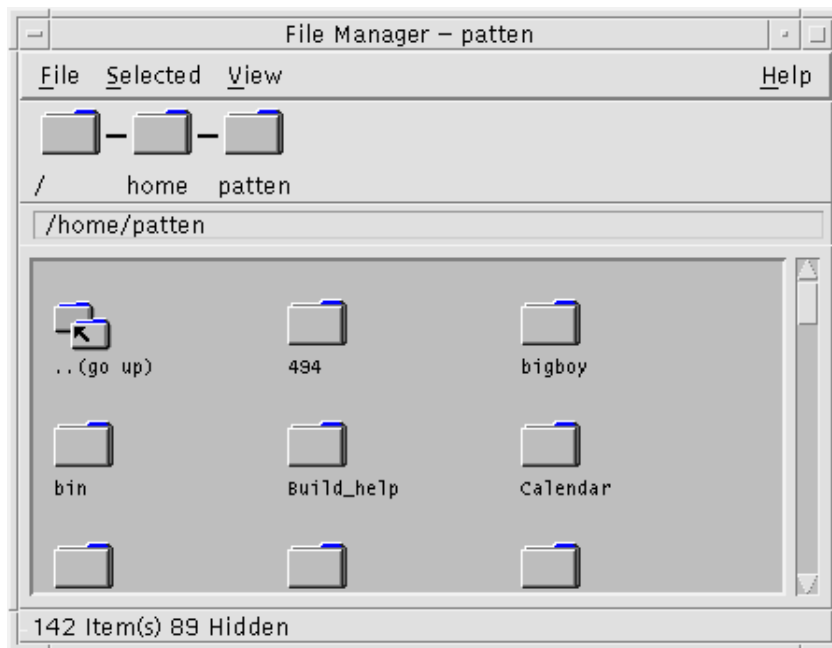
このアイコンをクリックして、起動します。

スタイル・マネージャからメニューが表示されます。そこからデスクトップの要素を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



フロントパネルからファイル・マネージャのアイコンをクリックして、起動しま



ファイル・マネージャ画面が表示されます。それを使用して、各種ファイル、フォルダ、アプリケーションをシステム上で管理します。

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、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す。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

CDE のスタイル・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。これらは、OpenWindows 環境でのワークスペースプロパティウィンドウのオプションに代わるものです。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・マネージャの詳細な説明については、『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内、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 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 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 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。

- 171ページの「コンパイラ」
- 172ページの「リンカ」
- 180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 SunOS 4 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 *Sun WorkShop Compiler C 4.2 AnswerBook Collection* で閲覧できます。

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 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 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	警告ではなく重大なエラーを発行
-assert nosymbolic	-zdefs	警告ではなく重大なエラーを発行
-assert pure-text	-ztext	警告ではなく重大なエラーを発行
-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。175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。175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 のオプション	Solaris 7 での変更	注
<code>no -e</code>	<code>-G</code>	共有オブジェクトを作成する。
<code>-lx[.v]</code>	<code>-lx</code>	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。
<code>-Ldir</code>	<code>-Ldir</code>	<i>dir</i> は実行可能プログラムに記録されない。かわりに、 <code>-R</code> オプションを使用。
<code>-M</code>	<code>-m</code>	
<code>-n</code>	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 <code>-n</code> として圧縮
<code>-N</code>	変更なし	
<code>-o name</code>	<code>-o name</code>	
<code>-p</code>	デフォルト	<code>-M mapfile</code> で取り消し可能。
<code>-r</code>	<code>-r</code>	
<code>-s</code>	変更なし	
<code>-S</code>	<code>-s</code>	
<code>-t</code>	変更なし	
<code>-T hex</code>	<code>-M mapfile</code>	<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。
<code>-Tdata hex</code>	<code>-M mapfile</code>	<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 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 では、`-e` オプションなしの場合、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。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 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 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` オプションはリンカに対して、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、176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

動的リンカと実行時リンカが SunOS 4 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 と Solaris 7 の動的リンカおよび実行時リンカ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カの検索パスは

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 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 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 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 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 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	Solaris 7
libc.so.1.7	libc.so.1
libdl.so.1.0	libdl.so.1

SunOS 4 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)` (124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ク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
```

(続く)

```
[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
```

デバッガ

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 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 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
```

(続く)

続き

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。

ツールとリソース

この章では、開発環境におけるツールとリソース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。

- 185ページの「`ioctl()` 要求」
- 188ページの「`ptrace()` 要求値」
- 190ページの「ライブラリ」
- 193ページの「`make` の使用」
- 194ページの「SCCS の使用」
- 194ページの「アプリケーション互換性の判断」
- 195ページの「アプリケーションパッケージ作成」
- 197ページの「ツールキット」
- 198ページの「SunOS 4 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 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 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 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。mtio(7) のマニュアルページを参照。
sockio	次の sockio ioctl 要求は、SVR4 と Solaris 7 において実装されている：SIOCSGRP、SIOCGGRP、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 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 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 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) のマニュアルページを参照。
termio, termios	すべての SunOS 4 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 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 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 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 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 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 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 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 Inrface 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 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 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。

リソースの制限

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リソースに使用されました。これらのリソースは、現在は動的に割り当てられません。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 にリソースの制限を示します。

表 16-3 リソースの制限

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 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 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 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 ディレクトリ	Solaris 7 ディレクトリ
librpcsvc.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 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じることがあります。

■ SunOS 4 `Makefile` - `install`:

```
install -o bin -g bin -m 444 target.c /usr/bin/target
```

■ SunOS 5.7 `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 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 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 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 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 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 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、「適切に動作する」ユーザアプリケーションで使用することができます。「適切に動作する」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 ツールの検索

ほとんどの SunOS 4 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 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 コマンド	SunOS 4 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 コマンド	SunOS 4 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 コマンド	SunOS 4 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、SunOS 5.7 でアンバンドルである SunOS 4 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。

- 203ページの「ネットワーク」
- 204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 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、156 ページの「NIS+」を参照してください。

NIS+ は、改善されたセキュリティ、名前空間オブジェクトの詳細なモデル、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、『*man pages section 3*』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 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、コンネクションレスの 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 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
- ユーザー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 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 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 5.7 のロケールデータベース (`/usr/lib/locale/locale`) は、SunOS 5 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 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 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。

- 209ページの「システム構成」
- 213ページの「再構成ブート」
- 213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 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 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 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 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	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
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、83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。213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` 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` 名はより簡単で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 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。リポ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。

デバイスドライバと STREAMS

この章では、デバイスドライバインタフェースの変更、`devinfo` コマンド、移行時の注意点、STREAMS、Solaris 7 ドライバアーキテクチャについて説明します。

- 217ページの「デバイスドライバと STREAMS デバイスドライバ」
- 225ページの「デバイスドライバ関連コマンド」

この章で説明する各項目についての詳細は、次のマニュアルを参照してください。

- 『*STREAMS Programming Guide*』
- 『システムインタフェース』
- 『Solaris のシステム管理 (第 1 巻)』

デバイスドライバと STREAMS デバイスドライバ

Solaris 7 では、デバイスドライバに多くの変更点があります。たとえば、新しい DDI/DKI ルーチン、Solaris SPARC DDI 固有ルーチン、新しいソフトウェアの特性、ロード可能なドライバなどです。また、以前からあったデバイスに関する多くの問題 (割込み、DVMA™、メモリマッピングなど) を、ドライバが意識しなくて済むようになりました。

デバイスドライバのインタフェース

以前の SunOS リリースでは、ドライバの作成者はデバイスドライバインタフェースの変更を処理しなければなりませんでした。通常、オペレーティングシステムのリ

リリースごとに移植を行なっていました。さらに、プラットフォームごとにインタフェースが異なるため、各プラットフォームに合ったデバイスドライバが必要でした。Sun 社製以外の製品のデバイスドライバのリリースには、デバイスドライバを統合するためにオペレーティングシステムの再設定および再構築を行う複雑なスクリプトも中には含まれていました。したがって、デバイスドライバのサポートと保守にはコストがかかりました。

SunOS システムの以前のリリース (SunOS 4.1.3 ソフトウェア以前) と違って、Solaris 7 環境のデバイスドライバインタフェースは統一化されて、Solaris 7 SPARC DDI/DKI と呼びます。Solaris 7 SPARC DDI/DKI は、サポートされているすべてのプラットフォームと、これらのプラットフォームの Solaris 環境のすべての将来のリリースにデバイスドライバのバイナリ互換性を提供するために作成されました。

DDI/DKI という用語は、SVR4 リリースにある元の仕様から取ったものです。これは、デバイスドライバインタフェース / デバイスカーネルインタフェースを表します。インタフェースは次の 3 グループに分割されます。

- DDI/DKI
- DKI 専用
- DDI 専用

DDI/DKI

DDI/DKI インタフェースは SVR4 で標準化されており、動作しているプラットフォームとは関係なく SVR4 のすべての処理系で共通です。

DKI

DKI 専用インタフェースは DDI/DKI インタフェースと同様に共通であり、すべての SVR4 処理系でサポートされます。ただし、System V の将来のリリースでサポートされる保証はありません。

DDI

DDI 専用インタフェースは、アーキテクチャ固有のもので、たとえば、デバイスとシステム固有ハードウェアへのアクセスおよび制御方法 (つまり、I/O レジスタ、DMA サービス、割り込み、およびメモリマッピング) が固有です。これらのインタフェースは、その他の SVR4 処理系での機能は保証されていません。

こういった特長によりドライバのサポートと保守のコストを効率よく下げることができます。この多数の SPARC プラットホームと結合した機能は、多くの新しい Sun 社製以外の製品のハードウェア開発者の役に立つでしょう。

このレベルのバイナリ互換性を提供することにより、いまでは Sun 社製以外の製品のハードウェア開発者は彼らのドライバハードウェアと一緒に DDI 準拠のデバイスドライバをシェリンクラップ形式にパッケージ化することができます。新しいドライバパッケージのインストールは、いまでは完全に自動化されています。自動構成を行うカーネルにより、ドライバを追加または削除するためにカーネルを再コンパイルする必要はなくなりました。したがって、Solaris 7 環境の DDI 準拠デバイスドライバは、他のすべての市販のソフトウェア製品と同様に扱うことができます。

Solaris 7 DDI/DKI では、DDI 専用インタフェースは Solaris 7 DDI/DKI をサポートするすべてのシステムに共通です。Sun Common SCSI Architecture (SCSA) のインタフェース、およびマルチスレッドカーネルでドライバを正常に機能させるために使用するロックインタフェースもまた、Solaris 7 環境では DDI 専用インタフェースとみなすことができますので注意してください。

SCSA により、デバイスドライバはホストアダプタのインプリメンテーションに関するプラットフォーム固有の細部を考慮しなくてもよくなりました。SCSA を使用すると、SCSI ドライバはサポートされるすべてのプラットフォームで実行できます。

上記のカテゴリのインタフェースだけを使うように制限したデバイスドライバを、「Solaris 7 DDI/DKI 準拠」といいます。Solaris 7 DDI/DKI に準拠したデバイスドライバを一般に「DDI に準拠したデバイスドライバ」といいます。

マニュアルページ

ドライバルーチン、構造体、および DDI/DKI を構成するサポートルーチンに関するマニュアルページは、『*man pages section 9: DDI and DKI Overview*』および以下のセクションを参照してください。詳細については Intro(9) のマニュアルページを参照してください。

- section 9E – DDI and DKI Driver Entry Points
- section 9F – DDI and DKI Kernel Functions
- section 9S – DDI and DKI Data Structures

デバイスドライバ開発者キット (DDK) は単独で使用可能です。

devinfo コマンド

Solaris 7 の devinfo コマンドは SunOS 4 の devinfo とは機能が異なります。新しい prtconf(1M) コマンドは SunOS 4 の devinfo コマンドで表示した情報を表示します。次の例で各コマンドの出力を示します。

```
4.1system% devinfo
Node 'SUNW,Sun 4/50', unit #0 (no driver)
  Node 'packages', unit #0 (no driver)
  Node 'openprom', unit #0 (no driver)
  Node 'zs', unit #0
  Node 'zs', unit #1
  Node 'audio', unit #0
  Node 'eeprom', unit #0 (no driver)
  Node 'counter-timer', unit #0 (no driver)
  Node 'memory-error', unit #0 (no driver)
  Node 'interrupt-enable', unit #0 (no driver)
  Node 'auxiliary-io', unit #0 (no driver)
  Node 'sbus', unit #0
    Node 'dma', unit #0
    Node 'esp', unit #0
      Node 'sr', unit #0
      Node 'sd', unit #0
    Node 'le', unit #0
    Node 'cgsix', unit #0
  Node 'memory', unit #0 (no driver)
  Node 'virtual-memory', unit #0 (no driver)
  Node 'fd', unit #0
  Node 'options', unit #0 (no driver)
```

```
5.3system% prtconf

System Configuration: Sun Microsystems sun4c
Memory size: 32 Megabytes
System Peripherals (Software Nodes):

SUNW,Sun 4_75
  packages (driver not attached)
  disk-label (driver not attached)
  deblocker (driver not attached)
  obp-tftp (driver not attached)
  openprom (driver not attached)
  zs, instance #0
  zs, instance #1
  audio (driver not attached)
  eeprom (driver not attached)
  counter-timer (driver not attached)
  memory-error (driver not attached)
  interrupt-enable (driver not attached)
  auxiliary-io (driver not attached)
  sbus, instance #0
    dma, instance #0
    esp, instance #0
      sd (driver not attached)
      st (driver not attached)
      sd, instance #0
      sd, instance #1 (driver not attached)
      sd, instance #2 (driver not attached)
      sd, instance #3
      sd, instance #4 (driver not attached)
      sd, instance #5 (driver not attached)
      sd, instance #6
    le, instance #0
    cgsix, instance #0
  memory (driver not attached)
  virtual-memory (driver not attached)
  fd (driver not attached)
  options, instance #0
  pseudo, instance #0
```

移行に関する留意点

自動構成を行うカーネルを使用すると、Solaris 7 ドライバは他のタイプのドライバよりも SBus ドライバに似ています。すべてのドライバがロード可能で、カーネルの構成は必要ありません。

SunOS 4 ソフトウェアでは、1 度に 1 つのプロセッサしかカーネルに存在できません。これはカーネル全体で 1 つのマスタロックを使うことで実現されていました。プ

プロセッサがカーネルコードを実行したいときは、ロックを獲得し(他のプロセッサがロックで保護されたコードを実行するのを防ぐ)、終了したらロックを解放します。

Solaris 7 カーネルはマルチスレッドです。1つのマスタロックの代わりに、コードの細分化された領域を保護するための多数の細分化されたロックがあります。たとえば、特定のvノードへのアクセスを保護するカーネルロックや、iノードを保護するカーネルロックがあります。一回に1つのプロセッサしかvノードを処理するコードを実行できませんが、別のプロセッサがiノードにアクセスすることは可能です。これにより多くの並列化が可能になります。

マルチスレッドカーネルは、ドライバの設計に大きな影響を与えます。spln/splrペアを使用した古いモデルは(ユニプロセッサまたはマルチプロセッサマシンにおいて)もはや動作しません。そのかわりにMTスタイルのロックを選択することができます。ドライバの最も一般的なロックは、相互排他ロック、*mutexes*(仕様ではspln/splrペアとほぼ同等)、および条件変数(*sleep()*/*wakeup()*同期化とほぼ同等)です。

注 - SpIN/pplr のペアは、割り込みをブロックしますが、マルチプロセッサ環境でデータ構造を保護する上では、この機能は役に立ちません。

sleep() を明示的に呼び出すまではプロセッサを所有しているという古い概念は、もはや成立しません。カーネルがプリエンティブになっているため、CPUはスレッドからスレッドへと切り換えられます。したがって、デバイスレジスタや共有データ構造などへの同時アクセスを防ぐため、適切なMTロックプリミティブを使用しなければなりません。

単純なデバイスドライバ用のドライバコードは、主にカーネルインタフェースルーチンで構成されていますが、そのかなりの部分に変更されます。ただし、分かりやすい変更です。SCSIドライバのように大量のデバイス固有の処理コードを持つ複雑なデバイスドライバの場合は、ドライバインタフェースの変更はわずかです。このドライバインタフェースは、カーネルからドライバへのインタフェースか、ドライバからカーネルへのインタフェースか、または、ドライバからドライバへのインタフェースです。

Solaris 7環境でどのようにドライバをサポートするかを決定する前に、ドライバがどのように動作するかについても一度調べてください。SunOS 4のドライバの動作を調べてください(特定の処理系での動作ではなく一般的な動作)。エクスポートしていたインタフェース、提供していた*ioctl()*、ハードウェアの動作、ドライバがサポートしていたハードウェアの特徴、また、ドライバが複数の*open()*呼び出しをサポートしていたかどうかなどです。

これらの変更はドライバに影響を与えるため、次の点を検討してください。

- ドライバのエントリポイントはかなり異なる
- ANSIC の要件
 - volatile キーワード
 - const キーワード
 - 関数プロトタイプ宣言
- 再配置またはリネームされたヘッダファイル (現在すべて、またはほとんどのシステムヘッダファイルは /usr/include/sys にある)
- ほとんどの構造体は不要になった。たとえば、
 - struct user
 - struct proc
 - struct dev_info

STREAMS

STREAMS モジュールでの変更点は、透過的な I/O 制御、ストリームへのモジュールの自動プッシュ、新しいメッセージタイプなどです。

透過的な ioctl()

SunOS 4 リリースでは、特定のドライバは ioctl() 要求を行う前は STREAMS ドライバであったということを知っておく必要があります。

STREAMS 以外のドライバについては、次のように直接 ioctl() 要求ができます。

```
ioctl(fd, DRIVER_IOCTL, arg);
```

STREAMS ドライバについては、次のように strioctl 構造を設定した後で使用しなければなりません。

```
ioctl(fd, I_STR, &strioctl);
```

ドライバが STREAMS ベースであったかどうかを判定する簡単な方法はありませんでした。現在では、ストリームヘッドに対する認識されない ioctl() はドライバ

に渡されるので、ドライバが STREAMS ベースであったかどうかを知る必要がなくなりました。

特に透過的 `ioctl()` をサポートするため、新しいメッセージタイプが Solaris 7 リリースに追加されました。現在、カーネルとの間のユーザデータの転送をストリームヘッドに通知するための「コピーイン」 および「コピーアウト」 メッセージがあります。

STREAMS ドライバを書く詳細については、『*STREAMS Programming Guide*』を参照してください。

autopush コマンド

SunOS 4 の `streamtab` 構造体では、デバイスが `open()` のときに特定の STREAMS モジュールをプッシュするようドライバ側で指定できました。

Solaris 7 では、システム管理者と `autopush(1M)` コマンドが、いつ STREAMS モジュールをプッシュするかを指定します。必要な場合、ドライバのインストール時に `autopush` を実行できます。

STREAMS モジュールのプッシュに関する詳しい情報については、『*STREAMS Programming Guide*』を参照してください。

Solaris 2.x ドライバアーキテクチャ

現在サポートされるハードウェアプラットフォームのすべてにバイナリ互換性を達成するため、DDI インタフェースはアーキテクチャ概念に沿って慎重に設計されました。基礎となる概念、つまり `device` ツリーは、元の SPARCstation™ 設計における `devinfo` ツリーの拡張です。デバイスツリーの各ノードはデバイス情報構造体または「`dev_info` ノード」によって記述されます。ツリーの最下部のノードをリーフノードといいます。ディスク、テープドライブ、フレームバッファ、I/O カード、およびネットワークインタフェースなどのほとんどのデバイスは、リーフノードに関連付けられるリーフデバイスの例です。対応するデバイスドライバをリーフドライバといいます。

ツリーにおける中間ノードは一般にバスと関連づけられます (たとえば SBus、SCSI、VME)。これらのノードを「`nexus` ノード」といい、それらに関連づけられるドライバを「`nexus` ドライバ」といいます。バス `nexi` は特定の要素と関連づけられるアーキテクチャの詳細をカプセル化するためのエンティティです。

現在、Solaris 7 DDI/DKI だけがリーフドライバ、*nexus*ドライバの 1 つの型、SCSI ホストバス・アダプタドライバの記述をサポートしています。

デバイスツリー構造はノード間に正式な親子関係を確立します。この親子関係はプラットフォームアーキテクチャの独立性にとって重要なポイントです。

リーフドライバがプラットフォーム依存 (たとえば DMA マッピング) のサービスを必要とする場合は、システムはサービスを提供するために要求をその親の呼び出しへと透過的に変換します。サービスを提供するのは常に *nexus* ドライバです。それぞれの *nexus* ドライバはサービスを提供するために、今度は要求をその親に渡すことができます。このアプローチにより、リーフドライバはプラットフォームのアーキテクチャとは関係なく機能することができます。

デバイスドライバ関連コマンド

デバイスドライバ関連コマンドには、`add_drv`、`rem_drv`、`modload`、`modunload` があります。

- `add_drv(1M)` – デバイスドライバが新しくインストールされたことをシステムに通知します。
- `rem_drv(1M)` – 指定したデバイスドライバモジュールが無効になったことをシステムに通知します。
- `modload(1M)` – 指定したロード可能なモジュールを実行中のシステムにロードします。
- `modunload(1M)` – 指定したロード可能なモジュールを実行中のシステムからアンロードします。

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 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 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 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 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 5.7 の欄に示されます。

注 - SunOS 5.7 のディレクトリ構造は SunOS 4 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 /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」として解釈されます。

/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、『*man pages section 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 コマンド	SunOS 5.7	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、『*man pages section 1: User Commands*』で fasthalt(1B) などのように参照してください。

表 A-2 例 2

SunOS 4 コマンド	SunOS 5.7	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 5.7 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 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 コマンド	SunOS 5.7	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 5.7 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。	N

SunOS 4 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 5.7 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 5.7 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 /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 5.7 date(1) のマニュアルページを参照してください。

表 A-4 例 4

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
rev(1)	N		N

SunOS 4 rev コマンドは SunOS 5.7 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 の全コマンドインタフェースを一覧表示し、Solaris 7 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。

表 A-5 コマンドリファレンス: ac ~ awk

SunOS 4 コマンド	SunOS 5.7	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
adbgen (8)	S		N
add_client (8)	N	admintool (1M)	N

表 A-5 コマンドリファレンス: ac ~ awk 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 オプションは SunOS 5.7 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 コマンドでは whatis データベースを使用したがる、SunOS 5.7 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 コマンドリファレンス: ac ~ awk 続く

SunOS 4 コマンド	SunOS 5.7	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 オプション、-d2、-h、-j、-J、-k、-L、-O[n]は、SunOS 5.7 コマンドでは利用できない。	N
at(1)	S	SunOS 5.7 システムの at、atq、および atrm コマンドは、SunOS 4 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 5.7 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 5.7 システムの at、atq、および atrm コマンドは、SunOS 4 システムの場合とは動作が少し異なる。SunOS 4 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 5.7 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 5.7 システムではより強化されている。	N
atrm(1)	C	SunOS 5.7 システムの at、atq、および atrm コマンドは、SunOS 4 システムの場合とは動作が少し異なる。SunOS 4 の「-」フラグは SunOS 5.7 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 5.7 システムではより強化されている。	N

表 A-5 コマンドリファレンス: ac ~ awk 続く

SunOS 4 コマンド	SunOS 5.7	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(8)	S		N
auditd(8)	S		N
automount(8)	C	次の SunOS 4 オプションは、SunOS 5.7 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

表 A-6 コマンドリファレンス: banner ~ bootparamd

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 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

表 A-6 コマンドリファレンス: banner ~ bootparamd 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
basename (1)	S	SunOS 5.7 および SunOS/BSD 互換バージョンは、ともに SunOS 4 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 5.7 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 5.7 の batch ジョブ <i>queuename</i> は指定されないで、SunOS 4 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
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 5.7 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。	N
bootparamd (8)	S		N

表 A-7 コマンドリファレンス: C2conv ~ cxref

SunOS 4 コマンド	SunOS 5.7	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は、ご購入先にお問い合わせください。	N
C2unconv (8)	N	本製品に関する情報は、ご購入先にお問い合わせください。	N
cal (1)	S		N

表 A-7 コマンドリファレンス: C2conv ~ cxref 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
calendar(1)	S		N
cancel(1)	S		N
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 5.7 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 5.7 の -t オプションは、SunOS 4 の -v オプションと同等で、非印刷文字を表示する。	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 コマンドはエラーを出力する。SunOS 5.7 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		

表 A-7 コマンドリファレンス: C2conv ~ cxref 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
checkeq(1)	S		N
checknr(1)	S		N
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 から SunOS 5.7 との間で変更された。SunOS 4 では、chgrp はシンボリック自体の所有権を変更していた。SunOS 5.7 では、chgrp はリンクをたどる。SunOS 5.7 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 5.7 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 5.7 での -R オプションはターゲットのモードを変更する。 SunOS 5.7 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 の chown はシンボリックリンクの所有権を変更していた。SunOS 5.7 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 5.7 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 5.7 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

表 A-7 コマンドリファレンス: C2conv ~ cxref 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
clear(1)	S		N
clear_colormap(1)	N		N
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。	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 5.7 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 プロパティウィ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 コマンドはエラーを出力するが、SunOS 5.7 コマンドはエラーを表示しないで無視する。	N
comm(1)	S		N

表 A-7 コマンドリファレンス: C2conv ~ cxref 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
compress(1)	S		N
config(8)	N		N
cp(1)	C	-R オプションは、SunOS 5.7 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 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 5.7 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

表 A-7 コマンドリファレンス: C2conv ~ cxref 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
ctrace(1V) -SysV	N	次の SunOS 4 オプションは、SunOS 5.7 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 および SunOS 5.7 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
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

表 A-8 コマンドリファレンス: date ~ dumpkeys

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
date(1V) -SysV	S		N
date(1V)	C	日付を設定するときに使用するフォーマットは、SunOS 5.7 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。	N
dbconfig(8)	S		N
dbx(1)	N	別パッケージの SPARCworks™ で利用できる。	N

表 A-8 コマンドリファレンス: date ~ dumpkeys 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
dbxtool(1)	N	別パッケージの SPARCworks で、コマンドデバッガとして利用できる。	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 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 5.7 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 プロパティウインドウによって処理される。	N
defaults_merge(1)	S		N
defaults_to_indentpro(1)		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
defaults_to_mailrc(1)		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
defaultsedit(1)		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
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 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 5.7 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

表 A-8 コマンドリファレンス: date ~ dumpkeys 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 と SunOS 5.7 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 では、名前引数はオプション。 SunOS 5.7 システムソフトウェアでは、必須。	N
df (1V) -SysV	C		N
df (1V)	C	SunOS 4 の df には、SunOS 5.7 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 5.7 で -k オプションを指定すれば、SunOS 4 コマンドに似た出力フォーマットを提供する。SunOS 4 では df -t 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 5.7 で df -t 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 5.7 では df -l 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 では、-c 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 5.7 では、-s 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

表 A-8 コマンドリファレンス: date ~ dumpkeys 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
dircmp(1V) -SysV	S		N
dirname(1V) -SysV	S		N
dis(1)	C	次の SunOS 4 でのオプションは SunOS 5.7 では利用できない。 -da sec: sec 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 5.7 では -D sec 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
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 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 5.7 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-8 コマンドリファレンス: date ~ dumpkeys 続く

SunOS 4 コマンド	SunOS 5.7	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 でのオプションは SunOS 5.7 にはない。</p> <p>-a <i>archive-file</i>: SunOS 5.7 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 5.7 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 5.7 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

表 A-9 コマンドリファレンス: e ~ extract_unbundled

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
e(1)	A	<p>ex(1) コマンドが同様の機能を提供。</p>	S
echo(1V) -SysV	S		N
echo(1V)	C	<p>-n オプションは SunOS 4 では改行を抑制した。SunOS 5.7 では \c を使用する。</p>	S

表 A-9 コマンドリファレンス: e ~ extract_unbundled 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
ed(1)	S		N
edit(1)	S		N
edquota(8)	S		N
EEPROM(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
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

表 A-9 コマンドリファレンス: e ~ extract_unbundled 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-10 コマンドリファレンス: false ~ fwtmp

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 でのオプションは SunOS 5.7 のコマンドにはない。 -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 のオプションは SunOS 5.7 コマンドでは利用できない。 -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-10 コマンドリファレンス: false ~ fwtmp 続く

SunOS 4 コマンド	SunOS 5.7	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
fpuversion4(8)	A	psrinfo -v で情報の入手可能。	N
from(1)	N		S
fsck(8)	C	SunOS 4 の fsck コマンドは SunOS 5.7 とはかなり異なる。SunOS 5.7 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-10 コマンドリファレンス: false ~ fwtmp 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
fuser(8)	S		N
fwtmp(8)	S		N

表 A-11 コマンドリファレンス: gcore ~ gxtest

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
gcore(1)	S		N
generic_args(1)	N		N
get(1)	C	SunOS 5.7 では ASCII ファイルだけを生成する。SunOS 4 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 ではエラーを表示する。SunOS 5.7 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

表 A-11 コマンドリファレンス: gcore ~ gxtest 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
glob(1)	S		N
goto(1)	S		N
gpconfig(8)	N		N
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

表 A-12 コマンドリファレンス: halt ~ htable

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
halt(8)	S		N
hashcheck(1)	S		N
hashmake(1)	S		N
hashstat(1)	S		N
head(1)	S		N
help(1)	S		N

表 A-12 コマンドリファレンス: halt ~ htable 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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
htable(8)	S		N

表 A-13 コマンドリファレンス: i386 ~ isainfo

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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
imentest(8C)	N		N
implot(1G)	N		Y

表 A-13 コマンドリファレンス: i386 ~ isainfo 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 と SunOS 5.7 の間で異なる。SunOS 4 では範囲は 512~1023 であり、SunOS 5.7 では 0~1023。	N
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

表 A-13 コマンドリファレンス: i386 ~ isainfo 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
infocmp(8V)	C	-s オプションの構文は、SunOS 4 と SunOS 5.7 の間で異なる。SunOS 5.7 では、-s とその引数の間にスペースを入れなければならない。SunOS 4 では、スペースはオプション。	N
init(8)	C	SunOS 5.7 では SunOS 4 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。	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
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 と SunOS リリース 5.7 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

表 A-13 コマンドリファレンス: i386 ~ isainfo 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-14 コマンドリファレンス: join

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
join(1)	C	SunOS 4 では、-a オプションは引数として 1、2、または 3 をとる。SunOS 5.7 では、この値は 1 または 2 のみが可能。SunOS 4 では -j への引数は 1 または 2 のみが可能。SunOS 5.7 ではそのような制約はない。	N

表 A-15 コマンドリファレンス: kadb ~ kill

SunOS 4 コマンド	SunOS 5.7	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

表 A-16 コマンドリファレンス: labelit ~ lsw

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
labelit(8)	S		N
last(1)	S		N
lastcomm(1)	S		N
lastlogin(8)	S		N
ld(1)	C	SunOS 4 ld コマンドと SunOS 5.7 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 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 5.7 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
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 でのオプションは SunOS 5.7 コマンドでは利用できない。-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

表 A-16 コマンドリファレンス: labelit ~ lsw 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
ln(1V)	C	すでにターゲットが存在する場合、SunOS 4 ではそれを削除しない。SunOS 5.7 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 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 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 /usr/5bin/ln -F オプションは、SunOS 5.7 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
logger(1)	N		S
login(1)	S		N
logname(1)	S		N
look(1)	S		N
lookbib(1)	S		N
lorder(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

表 A-16 コマンドリファレンス: labelit ~ lsw 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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(1) コマンドが同様の機能を提供。	S
lpstat(1)	S		N
lpctest(1)	N		S
ls(1V) -SysV	C		N
ls(1V)	S		S
lsw(1)	N		N

表 A-17 コマンドリファレンス: m4 ~ mv

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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
mach(1)	S		S
Mail(1)	A	mailx(1) コマンドが同様の機能を提供。	N
mail(1) - UCB	mailx		S

表 A-17 コマンドリファレンス: m4 ~ mv 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
mail(1)	C	現在は /usr/bin/mailにあるが、SunOS 4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の mail は以下を除く SunOS 5.7 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 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 5.7 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 プロパティウィ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 5.7 インタフェースは SunOS 4 インタフェースと互換性がある。SunOS 5.7 では SunOS 4 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-17 コマンドリファレンス: m4 ~ mv 続く

SunOS 4 コマンド	SunOS 5.7	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。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 5.7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-17 コマンドリファレンス: m4 ~ mv 続く

SunOS 4 コマンド	SunOS 5.7	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 5.7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に、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

表 A-18 コマンドリファレンス: named ~ nulladm

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-18 コマンドリファレンス: named ~ nulladm 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
netstat(8C)	S		N
newaliases(8)	S		N
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 には、 <code>csch</code> に組み込まれているものと、 <code>/usr/bin</code> にインストールされている、2つの <code>nice</code> がある。 <code>csch</code> 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、 <code>/usr/bin/nice</code> のデフォルト値は 10。SunOS 5.7 コマンドのデフォルトは 10。 <code>csch</code> に組み込まれている SunOS 4 コマンドでは、追加の <code>-+</code> オプション (<code>nice -+n</code>) は、 <code>nice</code> 値を n だけ増分するのではなく、 <code>nice</code> 値を n に設定するという点で、 <code>/usr/bin</code> にある SunOS 4 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	<code>-1</code> オプションの機能は、バージョン間で異なる。SunOS 4 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 <code>addr</code> を変更しても有効にならないが、SunOS 5.7 ソフトウェアでは即座に有効になる。SunOS 4 ソフトウェアでは <code>addr</code> は 16 進表記で指定できるが、SunOS 5.7 ソフトウェアではできない。SunOS 4 の <code>-m</code> オプションは SunOS 5.7 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-18 コマンドリファレンス: named ~ nulladm 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
nm(1)	C	次の SunOS 4 オプションは SunOS 5.7 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 および SunOS 5.7 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

表 A-19 コマンドリファレンス: od ~ overview

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-19 コマンドリファレンス: od ~ overview 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
old-perfmon(1)	N		N
old-prmail(1)	N		N
old-pti(1)	N		N
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

表 A-20 コマンドリファレンス: pac ~ pwdauthd

SunOS 4 コマンド	SunOS 5.7	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 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 5.7 では、{NAME_MAX}-2 に制限される。SunOS 5.7 の pack および unpack コマンドは SunOS 4 コマンドと互換性がある。	N
page(1)	S		N
pagesize(1)	S		S

表 A-20 コマンドリファレンス: pac ~ pwdauthd 続く

SunOS 4 コマンド	SunOS 5.7	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
pcat(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 5.7 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-20 コマンドリファレンス: pac ~ pwdauthd 続く

SunOS 4 コマンド	SunOS 5.7	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 -v オプションは SunOS 5.7 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 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 5.7 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 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 5.7 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 オプションは、SunOS 5.7 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 では、このオプションはコマンド名を表示する。SunOS 5.7 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-20 コマンドリファレンス: pac ~ pwdauthd 続く

SunOS 4 コマンド	SunOS 5.7	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、ご購入先にお問い合わせください。	N

表 A-21 コマンドリファレンス: quot ~ quotaon

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
quot(8)	S		N
quota(1)	S		N
quotacheck(8)	S		N
quotaoff(8)	S		N
quotaon(8)	S		N

表 A-22 コマンドリファレンス: ranlib ~ rwho

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
ranlib(1)	C	ar(1) コマンドが同様の機能を自動的に提供。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
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 5.7 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	prionctl(1) コマンドが同様の機能を提供。	S

表 A-22 コマンドリファレンス: ranlib ~ rwho 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
repquota(8)	S		N
reset(1)	A	stty が同様の機能を提供。	S
restore(8)	A	SunOS 5.7 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
rev(1)	N		N
rexd(8C)	A	in.rexd が同様の機能を提供。	N
rexc(8C)	A	in.rexc 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 におけるエスケープのための ~dsusp シーケンスは、SunOS 5.7 では利用できない。また、-e オプションの構文は、SunOS 4 と SunOS 5.7 の間で異なる。SunOS 4 では、構文は -ec。SunOS リリース 5.7 では、-e c。	N
rlogind(8C)	A	in.rlogind も同様の機能を提供。	N
rm(1)	S		N

表 A-22 コマンドリファレンス: ranlib ~ rwho 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
rm_client(8)	A	SunOS 5.7 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
rmail(8C)	C	SunOS 4 では、受信したりモートからのメールを uucp(1C) を使用して処理する。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 5.7 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
rmddel(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 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 5.7 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 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 5.7 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

表 A-22 コマンドリファレンス: ranlib ~ rwho 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
rpc.mountd	A	mountd も同様の機能を提供。	N
rpc.rexd(8C)	S		N
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.rwalld(8C)	S	現在は /usr/lib/netsvc/rwall にある。	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.ypupdated(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-22 コマンドリファレンス: ranlib ~ rwho 続く

SunOS 4 コマンド	SunOS 5.7	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

表 A-23 コマンドリファレンス: sa ~ syslogd

SunOS 4 コマンド	SunOS 5.7	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 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 5.7 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
sccs-help (1)	S		N
sccs-prs (1)	S		N

表 A-23 コマンドリファレンス: sa ~ syslogd 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
scs-prt (1)	S		N
scs-rmdel (1)	S		N
scs-sact (1)	S		N
scs-scsdiff (1)	S		N
scs-unget (1)	S		N
scs-val (1)	S		N
scsdiff (1)	C		N
screenblank (1)	C	OpenWindows xset -s -600 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 の /usr/5bin/sed および SunOS 5.7 の sed 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
sendmail (8)	S		N
set4 (8)	N		N

表 A-23 コマンドリファレンス: sa ~ syslogd 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 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 の shutdown(8) コマンドは SunOS 5.7 の shutdown(1M) コマンドとはかなり異なる。デフォルトでは、SunOS 5.7 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 の shutdown(8) は確認を求めない。さらに、SunOS 5.7 のコマンドには、-f、-h、-k、-n、-r の SunOS 4 オプションが存在しない。	S

表 A-23 コマンドリファレンス: sa ~ syslogd 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
size(1)	C	SunOS 4 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 5.7 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 での -h <i>spellhist</i> オプションは、SunOS 5.7 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 の -i <i>delay</i> オプションは SunOS 5.7 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-23 コマンドリファレンス: sa ~ syslogd 続く

SunOS 4 コマンド	SunOS 5.7	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 5.7 の stty コマンドによってサポートされない SunOS 4 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 の -f オプションは SunOS 4 の /usr/5bin/suu または SunOS 5.7 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 5.7 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 インストールライブラリ (SPARC 版)』を参照。	N
sunview(1)	A	SunView™ は、SunOS 5.7 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-23 コマンドリファレンス: sa ~ syslogd 続く

SunOS 4 コマンド	SunOS 5.7	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 5.7 の swap コマンドのオプションが SunOS 4 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

表 A-24 コマンドリファレンス: t300 ~ tzsetup

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-24 コマンドリファレンス: t300 ~ tzsetup 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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
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 では、SunOS 4 の /usr/5bin/time および SunOS 5.7 とは異なる出力を表示する。SunOS 4 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

表 A-24 コマンドリファレンス: t300 ~ tzsetup 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
touch(1V) -SysV	S		N
touch(1V)	C	SunOS 4 の -f オプションは利用できない。このオプションは、 <i>filename</i> 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
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

表 A-25 コマンドリファレンス: u370 ~ uuxqt

SunOS 4 コマンド	SunOS 5.7	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 5.7 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 s.file を処理する方法が異なる。SunOS 4 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。SunOS 5.7 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-25 コマンドリファレンス: u370 ~ uuxqt 続く

SunOS 4 コマンド	SunOS 5.7	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 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 5.7 では、{NAME_MAX} - 2 に制限される。SunOS 5.7 の pack および unpack コマンドは SunOS 4 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
uucheck(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-25 コマンドリファレンス: u370 ~ uuxqt 続く

SunOS 4 コマンド	SunOS 5.7	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

表 A-26 コマンドリファレンス: vacation ~ vwidth

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-26 コマンドリファレンス: vacation ~ vwidth 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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
vtroff(1)	N		N
vwidth(1)	N		N

表 A-27 コマンドリファレンス: w ~ write

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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

表 A-27 コマンドリファレンス: w ~ write 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
whois(1)	S		N
write(1)	S		N

表 A-28 コマンドリファレンス: xargs ~ xstr

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
xargs(1V) -SysV	S		N
xget(1)	N		N
xsend(1)a	N		N
xstr(1)	S		N

表 A-29 コマンドリファレンス: yacc ~ ypxfrd

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
yacc(1)	S		N
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

表 A-29 コマンドリファレンス: yacc ~ ypxfrd 続く

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
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 5.7 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

表 A-30 コマンドリファレンス: zcat ~ zic

SunOS 4 コマンド	SunOS 5.7	利用可能な代替コマンドと注	BSD
zcat(1)	S		N
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 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 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 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 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。が、`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 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 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、『*man pages section 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 システムコール	SunOS 5.7	利用可能な代替システムコールと注	ABI	SVID	SVR4	BSD
mctl() (2)	A	mementl() (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 5.7 では利用できません。mementl() 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 システムコール	SunOS 5.7	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 の getsockname() が使用する errno 値 ENOBUFS は、SVR4 と SunOS 5.7 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 5.7 と SVR4 における getsockname() コールは SunOS 4 のものと同じですが、SunOS 4 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 5.7 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

表 B-1 システムコールのリファレンス: accept() ~ auditsvc()

SunOS 4 システム コール	SunOS 5.7	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 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 5.7、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 と、SunOS 5.7、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

表 B-2 システムコールのリファレンス: bind() ~ brk()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
bind() (2)	S	現在は bind() (3N)	N	N	S	N
brk() (2)	S		N	N	S	N

表 B-3 システムコールのリファレンス: chdir() ~ creat()

SunOS 4 システム コール	SunOS 5.7	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 の chmod() によってサポートされるが、5.7、ABI、SVID、SVR4 ではサポートされない。 ただし、同等の SunOS 5.7、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 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 では、chown() の owner および group 引数は int 型になる。SunOS 5.7、ABI、SVID、SVR4 では、owner は uid_t 型、group は gid_t 型になる。 SunOS 4 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 5.7 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 5.7 の lchown() (2) を使用する。	C	C	C	N
chown() (2V) -SysV	S		S	S	S	N

表 B-3 システムコールのリファレンス: `chdir()` ~ `creat()` 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>chroot()</code> (2)	S		S	S	S	N
<code>close()</code> (2V) -SysV	S		S	S	S	N
<code>connect()</code> (2)	S#	現在は <code>connect()</code> (3N)。	N	N	S#	N
<code>creat()</code> (2V)	C#	<p>SunOS 4 では、<code>creat()</code> の <code>mode</code> 引数は <code>int</code> 型で、SunOS 5.7、ABI、SVID、SVR4 では、<code>mode</code> 引数は <code>mode_t</code> 型。また、SunOS 5.7、ABI、SVID、SVR4 では <code><fcntl.h></code> に含まれるが、SunOS 4 では含まれない。シンボリックアクセスモード (<code><sys/stat.h></code>)、<code>S_IREAD</code> (00400)、<code>S_IWRITE</code> (00200)、<code>S_IEXEC</code> (00100) は SunOS 4 の <code>creat()</code> によってサポートされるが、SunOS 5.7、ABI、SVID、SVR4 ではサポートされない。</p> <p>ただし、同等の SunOS 5.7、または ABI、SVID、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) は同じ定義を持ち、SunOS 4 の <code><sys/stat.h></code> でも定義されているので使用できる。<code>errno</code> フラグの <code>ENXIO</code>、<code>EOPNOTSUPP</code> は、SunOS 4 の <code>chmod</code> 関数には有効だが、SunOS 5.7、ABI、SVID、SVR4 では無効。</p>	C#	C#	C#	N
<code>creat()</code> (2V) -SysV	C#	<p>シンボリックアクセスモード (<code><sys/stat.h></code>)、<code>S_IREAD</code> (00400)、<code>S_IWRITE</code> (00200)、<code>S_IEXEC</code> (00100) は SunOS 4 の <code>creat()</code> によってサポートされるが、SunOS 5.7、ABI、SVID、SVR4 ではサポートされない。ただし、同等の SunOS 5.7、または ABI、SVID、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) は同じ定義を持ち、SunOS 4 の <code><sys/stat.h></code> でも定義されているので使用できる。<code>errno</code> フラグの <code>ENXIO</code>、<code>EOPNOTSUPP</code> は、SunOS 4 の <code>chmod</code> 関数には有効だが、SunOS 5.7、ABI、SVID、SVR4 では無効。</p>	C#	C#	C#	N

表 B-4 システムコールのリファレンス: dup() ~ dup2()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
dup() (2V) -SysV	S		S	S	S	N
dup2() (2V) -SysV	S	現在は dup2() (3C)。	S	S	S	N

表 B-5 システムコールのリファレンス: execve() ~ _exit()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
execve() (2V) -SysV	S		S	S	S	N
_exit() (2V) -SysV	S		S	S	S	N

表 B-6 システムコールのリファレンス: fchdir() ~ ftruncate()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 の fchmod() によりサポートされるが、SunOS 5.7、ABI、SVID、SVR4 ではサポートされない。ただし、同等5.7、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 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

表 B-6 システムコールのリファレンス: fchdir() ~ ftruncate() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
fchroot() (2)	S		N	N	N	N
fcntl() (2V) -SysV	C	SunOS 4 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 5.7、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 の F_GETOWN および F_SETOWN コマンドは SunOS 5.7、ABI、 SVID、SVR4 ではサポートされない。	C	C	C	N
flock() (2)	N		N	N	N	S
fork() (2V)	C	SunOS 4 では、fork() は int 型の値を返 す。SunOS 5.7、ABI、SVID、SVR4 で は、fork() は pid_t 型の値を返す。ま た、SunOS 5.7、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれ るが、SunOS 4 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) システムコールに同等の 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

表 B-6 システムコールのリファレンス: fchdir() ~ ftruncate() 続く

表 B-7 システムコールのリファレンス: getauid() ~ getuid()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 では、getegid() は int 型の値 を返す。SunOS 5.7、ABI、SVID、SVR4 で は、getegid() は gid_t 型の値を返す。 また、SunOS 5.7、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれ るが、SunOS 4 では含まれない。	C	C	C	N
getegid() (2V) -SysV	S		S	S	S	N
geteuid() (2V)	C	SunOS 4 では、geteuid() は int 型の値 を返す。SunOS 5.7、ABI、SVID、SVR4 で は、geteuid() は uid_t 型の値を返す。 また、SunOS 5.7、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれ るが、SunOS 4 では含まれない。	C	C	C	N
geteuid() (2V) -SysV	S		S	S	S	N

表 B-7 システムコールのリファレンス: getauid() ~ getuid() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
getgid() (2V)	C	SunOS 4 では、getgid() は int 型の値を返す。SunOS 5.7、ABI、SVID、SVR4 では、getgid() は gid_t 型の値を返す。また、SunOS 5.7、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 では含まれない。	C	C	C	N
getgid() (2V) -SysV	S		S	S	S	N
getgroups() (2V)	C	SunOS 4 では、getgroups() の gidset 引数は int 型で、SunOS 5.7、ABI、SVID、SVR4 では grouplist 引数は gid_t 型である。また、SunOS 5.7、ABI、SVID、SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 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
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 の getpeername() には有効だが、SVR4 と SunOS 5.7 では無効。	N	N	S#	N

表 B-7 システムコールのリファレンス: getauid() ~ getuid() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
getpgid() (2V)	S		S	S	S	N
getpgrp() (2V)	C	SunOS 4 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 5.7、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 getpgrp() は int 型の値を返すが、SunOS 5.7、ABI、SVID、または SVR4 の getpgrp() は pid_t 型の値を返す。SunOS 5.7、ABI、SVID、および SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 では含まれない。	C	C	C	N
getpgrp() (2V) -SysV	S		S	S	S	N
getpid() (2V)	C	SunOS 4 では、getpid() は int 型の値を返す。SunOS 5.7、ABI、SVID、または SVR4 では、getpid() は pid_t 型の値を返す。また、SunOS 5.7、ABI、SVID、および SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 では含まれない。	C	C	C	N
getpid() (2V) -SysV	S		S	S	S	N
getppid() (2V)	C	SunOS 4 では、getppid() は int 型の値を 5.7、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 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

表 B-7 システムコールのリファレンス: getauid() ~ getuid() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
getrlimit() (2)	C	RLIMIT_RSS は SunOS 4 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 5.7、ABI、SVID、または SVR4 ではサポートされていない。SunOS 5.7、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 5.7、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 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 の getsockname() が使用する errno 値 ENOBUFS は、SVR4 と SunOS 5.7 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
getuid() (2V)	C	SunOS 4 では、getuid() は int 型の値を返す。SunOS 5.7、ABI、SVID、または SVR4 では、getuid() は type uid_t の値を返す。また、SunOS 5.7、ABI、SVID、および SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 では含まれない。	C	C	C	N
getuid() (2V) -SysV	S		S	S	S	N

表 B-8 システムコールのリファレンス: `ioctl()`

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>ioctl()</code> (2)	C	185ページの「 <code>ioctl()</code> 要求」の <code>ioctl()</code> 要求を参照のこと。	C	C	C	N

表 B-9 システムコールのリファレンス: `kill()` ~ `killpg()`

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>kill()</code> (2V)	C	SunOS 4 では、プロセスのグループにシグナルが送られ (同様に、 <code>pid</code> 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 5.7、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 では、 <code>pid</code> 引数は <code>int</code> 型で、SunOS 5.7、ABI、SVID、または SVR4 では、 <code>pid</code> 引数は <code>pid_t</code> 型である。また、SunOS 5.7、ABI、SVID、SVR4 では <code><sys/types.h></code> に含まれるが、SunOS 4 では含まれない。	C	C	C	N
<code>kill()</code> (2V) <code>-SysV</code>	S		S	S	S	N
<code>killpg()</code> (2)	A	現在では <code>killpg()</code> (3C)。 <code>kill()</code> (2) が同様の機能を提供する。 <code>killpg()</code> (<code>pgrp, sig</code>) を <code>kill()</code> (<code>-pgrp, sig</code>) に置き換える。	A	A	A	S

表 B-10 システムコールのリファレンス: link() ~ lstat()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
link() (2V) -SysV	C	SunOS 5.7、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

表 B-11 システムコールのリファレンス: mctl() ~ munmap()

SunOS 4 システム コール	SunOS 5.7	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
mincore() (2)	C	SunOS 4 では、引数 len は int 型で、SVR4 と SunOS 5.7 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 5.7 では <unistd.h> をインクルードする必要がある。	N	N	C	N

表 B-11 システムコールのリファレンス: mct1() ~ munmap() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mkdir() (2V)	C	<p>SunOS 4 では、モード引数は <i>int</i> 型で、SunOS 5.7、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 5.7、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 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 の <code>mkdir()</code> によってサポートされるが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.7、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 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 の <code>mkdir()</code> によってサポートされるが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.7、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 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-11 システムコールのリファレンス: mctl() ~ munmap() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mknod() (2V) -SysV	C	mknod() の mode 引数は、SunOS 4 では int 型で、SunOS 5.7、ABI、SVID、または SVR4 では mode_t 型である。dev 引数は、SunOS 4 では int 型で、SunOS 5.7、ABI、SVID、または SVR4 では dev_t 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 の mknod() によってサポートされるが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.7、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 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 では、-mmap 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 5.7、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-11 システムコールのリファレンス: mctl() ~ munmap() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mount() (2)	C#	<p>SunOS 4 の mount() と、SunOS 5.7、ABI、SVID、SVR4 の mount() は、多くの点で互換性がない。SunOS 4 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 5.7、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 5.7、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 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 5.7、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 5.7、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 では含まれない。</p> <p>SunOS 4 の <code><sys/mount.h></code> は、SunOS 5.7、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 の mount() に固有であり、SunOS 5.7、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 5.7、ABI、SVID、または SVR4 ではサポートされない。</p> <p>SunOS 4 の mount() は、SunOS 5.7、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-11 システムコールのリファレンス: mctl() ~ munmap() 続く

SunOS 4 システム コール	SunOS 5.7	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

表 B-12 システムコールのリファレンス: nfssvc()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
nfssvc() (2)	A	このインタフェース は、nfssys() (NFS_SVC,...) ルーチンによ り、SunOS 5.7 で置き換えられる。	N	N	N	N

表 B-13 システムコールのリファレンス: open() ~ open() - SysV

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
open() (2V)	C#	<p>open() への <i>mode</i> 引数は、SunOS 4 ソフトウェアでは int 型で、SunOS 5.7、ABI、SVID、または SVR4 では mode_t 型である。SunOS 4 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 5.7、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 では、O_NDELAY または O_NONBLOCK 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 5.7、ABI、SVID、または SVR4 では、O_NDELAY または O_NONBLOCK 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。</p> <p>また、SunOS 5.7、ABI、SVID、SVR4 では <sys/types.h> と <sys/stat.h> に含まれるが、SunOS 4 では含まれない。</p> <p>errno 値 EOPNOTSUPP は、SunOS 4 の open() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では返されない。</p>	C#	C#	C#	N
open() (2V) -SysV	S#	<p>errno 値 EOPNOTSUPP は、SunOS 4 の open() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では返されない。</p>	S#	S#	S#	N

表 B-14 システムコールのリファレンス: pathconf() ~ putmsg()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
pathconf() (2V) -SysV	S		S	S	S	N
pipe() (2V) -SysV	S		S	S	S	N
poll() (2)	S		S	S	S	N

表 B-14 システムコールのリファレンス: pathconf() ~ putmsg() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
profil() (2)	S		S	S	S	N
ptrace() (2)	C#	<p>SunOS 4 の ptrace() へのオプションの <i>addr2</i> 引数は、SunOS 5.7 ルーチンではサポートされない。ptrace() への要求引数は、SunOS 4 では enum ptracereq 型で、SunOS 5.7 では int() 型。ptrace() への <i>pid</i> 引数は、SunOS 4 では int() 型で、SunOS 5.7 では pid_t() 型。</p> <p>また、SunOS 4 では <sys/types.h> に含まれるが、SunOS 5.7 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 の ptrace() には有効だが、SunOS 5.7 では無効。</p> <p>有効な <i>request</i> 値に関する情報については、188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

表 B-15 システムコールのリファレンス: quotactl()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
quotactl() (2)	A	Q_QUOTACTL ioctl() が同様の機能を提供する。	A	A	A	N

表 B-16 システムコールのリファレンス: read() ~ rmdir()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
read() (2V)	C#	errno フラグの EISDIR、EWOULDBLOCK は、SunOS 4 の read() には有効だが、SunOS 5.7、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 では int() 型で、SunOS 5.7 では unsigned() 型である。 SunOS 5.7 の read() は、(FIONBIO ioctl() 要求、あるいは <sys/file.h> から FNDELAY フラグまたは 4 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 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 の read() には有効だが、SunOS 5.7 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 には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
readv() (2V) -SysV	C#	SunOS 4、SunOS 5.7、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 の iovec の iov_len フィールドは、integer として定義されるが、SunOS 5.7、SVID または SVR4 の iov_len は unsigned として定義される。SunOS 5.7、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 のようにはサポートしない。	C#	C#	C#	N
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

表 B-16 システムコールのリファレンス: read() ~ rmdir() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 5.7、ABI、SVID、または SVR4 では <unistd.h> に含まれるが、SunOS 4 では含まれない。errno フラグ ENOTEMPTY は、SunOS 4 の rename() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。SunOS 5.7、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 5.7、ABI、SVID、または SVR4 では <unistd.h> に含まれるが、SunOS 4 では含まれない。errno フラグ ENOTEMPTY は、SunOS 4 の rmdir() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。SunOS 5.7、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

表 B-17 システムコールのリファレンス: sbrk() ~ sysconf()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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

表 B-17 システムコールのリファレンス: sbrk() ~ sysconf() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
send() (2)	S#	現在は send() (3N)。 errno フラグの ENOBUFS は、SunOS 4 の send() (2) には有効だが、SVR4、および SunOS 5.7 では無効。	N	N	S#	N
sendmsg() (2)	S#	現在は sendmsg() (3N)。 errno フラグの ENOBUFS は、SunOS 4 の sendmsg() (2) には有効だが、SVR4 と SunOS 5.7 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 の sendto() (2) には有効だが、SVR4 と SunOS 5.7 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 では、gidset 引数は int 型で、SunOS 5.7、ABI、SVID、または SVR4 では、gidset 引数は gid_t 型になっている。また、SunOS 5.7、ABI、SVID、または SVR4 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 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

表 B-17 システムコールのリファレンス: sbrk() ~ sysconf() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
setpgid() (2V) -SysV	S		S	S	S	N
setpgrp() (2V)	C	SunOS 4 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 5.7 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 の setpgrp() コールは引数なしの SunOS 5.7 の setpgrp() コールと同じになる。また、SunOS 4 の setpgrp() は int 型の値を返すが、SunOS 5.7 では、setpgrp() は pid_t 型の値を返す。さらに、SunOS 5.7 では <unistd.h> と <sys/types.h> に含まれるが、SunOS 4 では含まれない。errno フラグの EACCES、EINVAL、ESRCH は、SunOS 4 の setpgrp() には有効だが、SunOS 5.7、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 の setpgrp() (2V) には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。	S	S	S	N
setpriority() (2)	A	現在は setpriority() (3C)。prioctl() (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-17 システムコールのリファレンス: sbrk() ~ sysconf() 続く

SunOS 4 システム コール	SunOS 5.7	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 5.7、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 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 5.7 の sigpause() は、SunOS 4 の sigpause() (2) と互換性がある。	C	C	S	S
sigpending() (2V) -SysV	S		S	S	S	N

表 B-17 システムコールのリファレンス: sbrk() ~ sysconf() 続く

SunOS 4 システム コール	SunOS 5.7	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 5.7、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 では、PF_IMPIPKN はサポートされた <i>domain</i> だが、SVR4 と SunOS 5.7 では PF_IMPIPKN はサポートされない。errno フラグの ENOBUFS、EPROTOTYPE は、SunOS 4 の socket() には有効だが、SVR4 と SunOS 5.7 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-17 システムコールのリファレンス: sbrk() ~ sysconf() 続く

SunOS 4 システム コール	SunOS 5.7	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

表 B-18 システムコールのリファレンス: tell() ~ truncate()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
tell() (2V) -SysV	S		S	N	S	N
truncate() (2)	S	現在は truncate() (3C)	N	N	S	N

表 B-19 システムコールのリファレンス: umask() ~ utimes()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 の umask() によってサポートされるが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。ただし、同等の SunOS 5.7、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 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

表 B-19 システムコールのリファレンス: umask() ~ utimes() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
unlink() (2V) -SysV	S		S	S	S	N
umount() (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

表 B-20 システムコールのリファレンス: vadvice() ~ vhangup()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
vadvice() (2)	N		N	N	N	N
vfork() (2)	S		N	N	S	N
vhangup() (2)	S		N	N	N	N

表 B-21 システムコールのリファレンス: wait() ~ WTERMSIG()

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
wait() (2V)	C	SunOS 4 では、wait() は int 型の値を返す。SunOS 5.7、ABI、SVID、または SVR4 では、wait() は pid_t 型の値を返す。また、SunOS 5.7、ABI、SVID、SVR4 では <sys/types.h> に含まれるが、SunOS 4 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 でサポートされていたが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。SunOS 4 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 5.7、ABI、SVID、または SVR4 では、wait() は、シグナルが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 でサポートされていたが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。SunOS 4 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 5.7、ABI、SVID、または SVR4 では、wait() (2) は、シグナルが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-21 システムコールのリファレンス: wait() ~ WTERMSIG() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
waitpid() (2V)	C	SunOS 4 では、waitpid() は int 型の値を返す。SunOS 5.7、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 では int 型で、SunOS 5.7、ABI、SVID、または SVR4 では pid_t 型となっている。また、SunOS 5.7、ABI、SVID、SVR4 では <sys/types.h> に含まれるが、SunOS 4 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 でサポートされていたが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。SunOS 4 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 5.7、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 でサポートされていたが、SunOS 5.7、ABI、SVID、または SVR4 ではサポートされない。SunOS 4 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 5.7、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
WEXITSTATUS() (2)	C	union wait は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	S	S
WIFEXITED() (2)	C	union wait は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	C	S
WIFSIGNALED() (2)	C	union wait は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	C	S

表 B-21 システムコールのリファレンス: wait() ~ WTERMSIG() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
WIFSTOPPED() (2)	C	<i>union wait</i> は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	C	S
write() (2V)	C#	<p>SunOS 4 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、<i>ioctl()</i> の <i>FIONBIO</i> 要求を使用するか、<i>FNDELAY</i> や <i>O_NDELAY</i> フラグのセットのため <i>fcntl()</i> を使用することにより、<i>write()</i> は -1 を返して <i>errno</i> に <i>EWOULDBLOCK</i> を設定する。</p> <p>SunOS 5.7 ソフトウェアでは、通常ファイルへの <i>write()</i> では、<i>O_NDELAY</i> または <i>O_NONBLOCK</i> が設定されていれば、<i>write()</i> は -1 を返して <i>errno</i> に <i>EAGAIN</i> を設定する。</p> <p><i>O_NONBLOCK</i> または <i>O_NDELAY</i> が設定されているパイプまたは FIFO への <i>write()</i> 要求では、<i>write()</i> 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、<i>write()</i> 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、<i>O_NONBLOCK</i> が設定されると -1 を戻して <i>errno</i> に <i>EAGAIN</i> を設定し、<i>O_NDELAY</i> が設定されると 0 を返す。<i>O_NDELAY</i> が設定されると、{<i>PIPE_BUF</i>} またはそれ以下のバイト数の <i>write()</i> 要求は、完全に正常終了して <i>nbytes</i> を返すか、または 0 を返す。</p> <p>{<i>PIPE_BUF</i>} バイトより大きな <i>write()</i> 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 {<i>PIPE_BUF</i>} 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、<i>write()</i> は少なくとも {<i>PIPE_BUF</i>} バイトを転送する。SunOS 5.7 の <i>write()</i> 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><i>errno</i> フラグの <i>EWOULDBLOCK</i> は、SunOS 4 の <i>write()</i> には有効だが、SunOS 5.7 では無効。</p>	C#	C#	C#	N

表 B-21 システムコールのリファレンス: wait() ~ WTERMSIG() 続く

SunOS 4 システム コール	SunOS 5.7	注	ABI	SVID	SVR4	BSD
write() (2V) -SysV	C#	SunOS 5.7、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 の write() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
writev() (2V)	C#	SunOS 5.7、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 の writev() には有効だが、SunOS 5.7、ABI、SVID、または SVR4 では無効。	C#	C#	C#	N
WSTOPSIG() (2)	C	<i>union wait</i> は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	C	S
WTERMSIG() (2)	C	<i>union wait</i> は、SunOS 4 でサポートされていたが、SVR4 および SunOS 5.7 ではサポートされない。	N	N	C	S

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 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 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 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 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 5.7 の欄に示されます。

SunOS 4 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 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、『man pages section 3』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 コマンド	SunOS 5.7	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 5.7 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 コマンド	SunOS 5.7	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 と SunOS 5.7 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 リリースと同様に動作します。

ライブラリルーチン

表 C-1 ライブラリルーチンのリファレンス: a64l() ~ authunix_create_default()

SunOS 4	SunOS 5.7	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 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 5.7、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 ライブラリルーチンのリファレンス: a64l() ~ authunix_create_default() 続く

SunOS 4	SunOS 5.7	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 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 5.7、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 ライブラリルーチンのリファレンス: a64l() ~ authunix_create_default() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
assert() (3V)	C	SunOS 4 の assert() では exit() (3C)、SunOS 5.7、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 5.7、SVID、または SVR4 では 0 (ゼロ) を返し errno に EDOM を設定。SunOS 4 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 5.7、ABI、SVID、SVR4 ではなるべく authdes_seccreate() (3N) を使用する。	A	A	A	N

表 C-1 ライブラリルーチンのリファレンス: a641() ~ authunix_create_default() 続く

SunOS 4	SunOS 5.7	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

表 C-2 ライブラリルーチンのリファレンス: baudrate() ~ bzero()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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-2 ライブラリルーチンのリファレンス: baudrate() ~ bzero() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
box() (3V)	C	SunOS 4 の box() は関数。 SunOS 5.7 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。 SunOS 5.7、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 ソフトウェアの引数 verch および horch は char 型だが、SunOS 5.7、SVID、または SVR4 では ch 型。	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

表 C-3 ライブラリルーチンのリファレンス: calloc() ~ cv_waiters()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
calloc() (3)	S		S	S	S	N
callrpc() (3N)	A	利用可能。ただし、なるべく rpc_call() (3N) を使用する。	N	N	S	N

表 C-3 ライブラリルーチンのリファレンス: calloc() ~ cv_waiters() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
catclose() (3C)	S		S	S	S	N
catgetmsg() (3C)	A	SunOS 5.7、ABI、SVID、または SVR4 では、catgets() (3C) と strncpy() (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
catgets() (3C)	S		S	S	S	N
catopen() (3C)	S		S	S	S	N
cbc_crypt() (3)	S		N	N	N	N
cbreak() (3V) -SysV	S		N	S	S	S
cbrt() (3M)	S		N	S	S	N
ceil() (3M)	S		N	S	S	N
cfgetispeed() (3V)	S		S	S	S	N
cfgetospeed() (3V)	S		S	S	S	N
cfree() (3)	A	void free(void*ptr) () (malloc() (3C) を参照) に置換。	A	A	A	N
cfsetispeed() (3V)	S		S	S	S	N
cfsetospeed() (3V)	S		S	S	S	N
CHECK() (3L)	N		N	N	N	N
circle() (3X)	S		N	N	N	N
clear() (3V) -SysV	S		N	S	S	S
clearerr() (3V) -SysV	S		S	S	S	N

表 C-3 ライブラリルーチンのリファレンス: calloc() ~ cv_waiters() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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_spccreateerror() (3N)	S		S	S	S	N
clnt_sperrno() (3N)	S		S	S	S	N
clnt_sperro() (3N)	S		S	S	S	N
clntraw_create() (3N)	S	利用可能。ただし SunOS 5.7 および SVR4 ではなるべく clnt_raw_create() (3N) を使用する。	A	A	S	N

表 C-3 ライブラリルーチンのリファレンス: calloc() ~ cv_waiters() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
clnttcp_create() (3N)	S	利用可能。ただしSunOS 5.7、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 5.7、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
clrtoobot() (3V) -SysV	S		N	S	S	S
clrtoeol() (3V) -SysV	S		N	S	S	S
compound() (3M)	N		N	N	N	N
cont() (3X)	S		N	N	N	N

表 C-3 ライブラリルーチンのリファレンス: `calloc()` ~ `cv_waiters()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<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 5.7、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 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>curls_inopts()</code> (3X) を参照) に置換。	A	A	A	N
<code>_crypt()</code> (3)	C	<code>crypt()</code> (3C) ルーチンが同様の機能を提供。	N	A	A	N

表 C-3 ライブラリルーチンのリファレンス: `calloc()` ~ `cv_waiters()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>crypt()</code> (3)	C	SunOS 4 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 5.7、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-3 ライブラリルーチンのリファレンス: `calloc()` ~ `cv_waiters()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>ctime()</code> (3V)	C	<p>SunOS 4 の <code>tm</code> 構造体には、SunOS 5.7、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 5.7、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を外部変数 <code>daylight</code> で示される。さらに SunOS 5.7、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 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 と SunOS 5.7、ABI、SVID、または SVR4 とでは異なる。SunOS 4 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 5.7、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-3 ライブラリルーチンのリファレンス: `calloc()` ~ `cv_waiters()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>cv_notify()</code> (3L)	N		N	N	N	N
<code>cv_send()</code> (3L)	N		N	N	N	N
<code>cv_wait()</code> (3L)	N		N	N	N	N
<code>cv_waiters()</code> (3L)	N		N	N	N	N

表 C-4 ライブラリルーチンのリファレンス: `dbm_clearerr()` ~ `dysize()`

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>dbm_clearerr()</code> (3)	S		N	N	N	N
<code>dbm_close()</code> (3X)	S	<code>dbm_close()</code> (3) ルーチンが同様の機能を提供。	N	N	N	N
<code>dbm_delete()</code> (3)	S	<code>dbm_delete()</code> (3) ルーチンが同様の機能を提供。	N	N	N	N
<code>dbm_error()</code> (3)	S		N	N	N	N
<code>dbm_fetch()</code> (3)	S		N	N	N	N
<code>dbm_firstkey()</code> (3)	S		N	N	N	N
<code>dbm_nextkey()</code> (3)	S		N	N	N	N
<code>dbm_open()</code> (3)	S		N	N	N	N
<code>dbm_store()</code> (3)	S		N	N	N	N
<code>dbmclose()</code> (3X)	N		N	N	N	S
<code>dbminit()</code> (3X)	S		N	N	N	S
<code>decimal_to_double()</code> (3)	S		N	N	N	N

表 C-4 ライブラリルーチンのリファレンス: dbm_clearerr() ~ dysize() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
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

表 C-4 ライブラリルーチンのリファレンス: dbm_clearerr() ~ dysize() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
double_to_decimal() (3)	S		N	N	N	N
douupdate() (3V) -SysV	S		N	S	S	N
draino() (3V) -SysV	S		N	N	N	N
drand48() (3)	S		N	S	S	N
dysize() (3V)	N		N	N	N	N

表 C-5 ライブラリルーチンのリファレンス: ecb_crypt() ~ extended_to_decimal()

SunOS 4	SunOS 5.7	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-5 ライブラリルーチンのリファレンス: `ecb_crypt()` ~ `extended_to_decimal()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>endfsent()</code> (3)	A	<code>fclose()</code> (3) に置換。	A	A	A	N
<code>endgraent()</code> (3)	N		N	N	N	N
<code>endgrent()</code> (3V)	S		S	S	S	N
<code>endhostent()</code> (3N)	S		N	N	S	N
<code>endmntent()</code> (3)	A	<code>fclose()</code> (3) に置換。	A	A	A	N
<code>endnetent()</code> (3N)	S		N	N	S	N
<code>endnetgrent()</code> (3N)	S		N	N	N	N
<code>endprotoent()</code> (3N)	S		N	N	S	N
<code>endpwaent()</code> (3)	N		N	N	N	N
<code>endpwent()</code> (3V)	S		S	S	S	N
<code>endrpcent()</code> (3N)	S		N	N	S	N
<code>endservent()</code> (3N)	S		N	N	S	N
<code>endttyent()</code> (3)	N	SunOS 5.7 の <code>tty</code> システムに関する情報は <code>ttymon</code> (1) および <code>ttydefs</code> (4) を参照。	N	N	N	N
<code>endusershell()</code> (3)	S		N	N	N	N
<code>endwin()</code> (3V)	C	SunOS 4 での <code>endwin()</code> 戻り値は不定。SunOS 5.7、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>endwin()</code> (3V) -SysV	S		N	S	S	N
<code>erand48()</code> (3)	S		N	S	S	N

表 C-5 ライブラリルーチンのリファレンス: ecb_crypt() ~ extended_to_decimal() 続く

SunOS 4	SunOS 5.7	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-5 ライブラリルーチンのリファレンス: `ecb_crypt()` ~ `extended_to_decimal()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>execl(3V)()</code> (3V) -SysV	S		S	S	S	N
<code>execle()</code> (3V)	C		C	C	C	N
<code>execle()</code> (3V) -SysV	S		S	S	S	N
<code>execlp()</code> (3V)	C		C	C	C	N
<code>execlp()</code> (3V) -SysV	S		S	S	S	N
<code>execv()</code> (3V)	C		C	C	C	N
<code>execv()</code> (3V) -SysV	S		S	S	S	N
<code>execvp()</code> (3V)	C		C	C	C	N
<code>execvp()</code> (3V) -SysV	S		S	S	S	N
<code>exit()</code> (3)	C	SunOS 4 および SunOS 5.7、ABI、SVID、または SVR4 でも、プロセスが終了する前に追加処理を行う。SunOS 4 では <code>on_exit()</code> (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 5.7、ABI、SVID、または SVR4 では <code>atexit()</code> ルーチンによって登録されたすべての関数を呼び出す。 <code>on_exit()</code> (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 の <code>exit()</code> (3) と SunOS 5.7、ABI、SVID、または SVR4 の <code>exit()</code> とは互換性がある。	C	C	C	N

表 C-5 ライブラリルーチンのリファレンス: ecb_crypt() ~ extended_to_decimal() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
exp() (3M)	C	SunOS 5.7、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 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 では HUGE が + (プラス) の無限大として定義されるため、exp() (HUGE) および exp() (-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、errno は生成されない。SunOS 5.7、SVID または SVR4 では errno に ERANGE が設定される。	N	C	C	N
exp10() (3M)	N		N	N	N	N
exp2() (3M)	N		N	N	N	N
expm1() (3)	N		N	N	N	N
exportent() (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
extended_to_decimal() (3)	S		N	N	N	N

表 C-6 ライブラリルーチンのリファレンス: fabs() ~ fwrite()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
fabs() (3M)	S		N	S	S	N
fclose() (3S)	S		S	S	S	N
fconvert() (3)	S		N	N	N	N

表 C-6 ライブラリルーチンのリファレンス: fabs() ~ fwrite() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
fcvt() (3)	S		N	N	S	N
fdopen() (3V)	S		S	S	S	N
feof() (3V)	S		S	S	S	N
ferror() (3V)	S		S	S	S	N
fetch() (3X)	A	SunOS 5.7 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 5.7 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

表 C-6 ライブラリルーチンのリファレンス: fabs() ~ fwrite() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
flash() (3V) -SysV	S		N	S	S	N
floatingpoint() (3)	S		N	N	N	N
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 5.7、SVID、または SVR4 では x を返し errno に EDOM を設定する。SunOS 4 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

表 C-6 ライブラリルーチンのリファレンス: fabs() ~ fwrite() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
ftime() (3V)	S	現在は ftime() (3C)。	A	A	A	S
ftok() (3)	S		S	S	S	N
ftw() (3)	S		S	S	S	N
func_to_decimal() (3)	N		N	N	N	N
fwrite() (3S)	S		S	S	S	N

表 C-7 ライブラリルーチンのリファレンス: gamma() ~ gtty()

SunOS 4	SunOS 5.7	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-7 ライブラリルーチンのリファレンス: gamma() ~ gttty() 続く

SunOS 4	SunOS 5.7	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 5.7、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 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 5.7 では SunOS 4 と互換性がある。SunOS 4 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) のマニュアルページを参照。	A	A	A	N

表 C-7 ライブラリルーチンのリファレンス: gamma() ~ gtty() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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) のマニュアルページを参照。	A	A	A	N
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-7 ライブラリルーチンのリファレンス: gamma() ~ gttty() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
getmntent() (3)	C	SunOS 4 と SunOS 5.7、SVID または SVR4 の互換性はない。SunOS 4 では mntent 型のオブジェクトへのポインタを返すが、SunOS 5.7、SVID または SVR4 の getmntent() は int 型を返す。また SunOS 5.7、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-7 ライブラリルーチンのリファレンス: gamma() ~ gttty() 続く

SunOS 4	SunOS 5.7	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() (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-7 ライブラリルーチンのリファレンス: gamma() ~ gttty() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
gettmode() (3V)	C	SunOS 5.7 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 5.7 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 5.7 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-8 ライブラリルーチンのリファレンス: halfdelay() ~ hypot()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
HUGE() (3M)	C	HUGE は SunOS 4 では infinity() (3M) として <math.h> で定義され、IEEE の無限大を生成。SunOS 5.7、SVID、または SVR4 では <math.h> で定義されるマシン依存型定数。	N	C	C	N
HUGE_VAL() (3M)	C	SunOS 4 では infinity() (3M) として <math.h> で定義され、IEEE の無限大を生成。SunOS 5.7、SVID、または SVR4 では <math.h> で定義されるマシン依存型定数。	N	C	C	N
hypot() (3M)	S		N	S	S	N

表 C-9 ライブラリルーチンのリファレンス: idlok() ~ itom()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
idlok() (3V)	C	SunOS 4 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 5.7、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-9 ライブラリルーチンのリファレンス: idlok() ~ itom() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
initscr() (3V)	C	SunOS 4 では関数。一方、SunOS 5.7、SVID または SVR4 では initscr32() を呼び出すマクロ。エラーが発生すると、SunOS 4 では ERR を返すが、SunOS 5.7、SVID または SVR4 では適当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
initscr() (3V) -SysV	S		N	S	S	N
initstate() (3)	S	現在は initstate() (3)。	N	A	A	S
innetgr() (3N)	S		N	N	N	N
insch() (3V) -SysV	S		N	S	S	S
insertln() (3V) -SysV	S		N	S	S	S
insque() (3)	S		N	N	S	N
intrflush() (3V) -SysV	S		N	S	S	N
ipalloc() (3R)	N		N	N	N	N
irint() (3M)	N	rint() に置換された。	N	N	N	N
isalnum() (3V)	S		S	S	S	N
isalpha() (3V)	S		S	S	S	N
isascii() (3V)	S		S	S	S	N
isatty() (3V)	S		S	S	S	N
iscntrl() (3V)	S		S	S	S	N
isdigit() (3V)	S		S	S	S	N

表 C-9 ライブラリルーチンのリファレンス: idlok() ~ itom() 続く

SunOS 4	SunOS 5.7	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

表 C-10 ライブラリルーチンのリファレンス: j0() ~ jrand48()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
j0() (3M)	C	j0(HUGE)(), j1(HUGE)(), jn(4, HUGE)() と指定した場合、SunOS 4 ではエラーを表示せずに 0(ゼロ) を返す。 SunOS 5.7、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
j1() (3M)	C		N	C	C	N
jn() (3M)	C		N	C	C	N
jrand48() (3)	S		N	S	S	N

表 C-11 ライブラリルーチンのリファレンス: key_decryptsession() ~ kvm_write()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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

表 C-11 ライブラリルーチンのリファレンス: key_decryptsession() ~ kvm_write() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
kvm_setproc() (3K)	S		N	N	S	N
kvm_write() (3K)	S		N	N	N	N

表 C-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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

表 C-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
ldaopen() (3X)	N		N	N	N	N
ldclose() (3X)	N		N	N	N	N
ldexp() (3M)	C	SunOS 4 と、SunOS 5.7、ABI、SVID、または SVR4 とはオーバーフロー処理のみ異なる。SunOS 4 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 5.7、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

表 C-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
ldrseek() (3X)	N		N	N	N	N
ldshread() (3X)	N		N	N	N	N
ldsseek() (3X)	N		N	N	N	N
ldtbindex() (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 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 5.7、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-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
log10() (3M)	C	SunOS 4 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 5.7、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() (3)	C	現在は _longjmp() (3C)。siglongjmp() (3) ルーチンが同様の機能を提供。	A	A	A	S
longjmp() (3V)	S		S	S	S	S
longname() (3V)	C	SunOS 4 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 5.7、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 5.7、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 5.7、SVID または SVR4 へ移植するには、SunOS 4 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

表 C-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
ltol3() (3C)	N		N	N	N	N
lwp_checkstkset() (3L)	N		N	N	N	N
lwp_create() (3L)	N		N	N	N	N
lwp_ctxinit() (3L)	N		N	N	N	N
lwp_ctxmemget() (3L)	N		N	N	N	N
lwp_ctxmemset() (3L)	N		N	N	N	N
lwp_ctxremove() (3L)	N		N	N	N	N
lwp_ctxset() (3L)	N		N	N	N	N
lwp_datastk() (3L)	N		N	N	N	N
lwp_destroy() (3L)	N		N	N	N	N
lwp_enumerate() (3L)	N		N	N	N	N
lwp_errstr() (3L)	N		N	N	N	N
lwp_fpset() (3L)	N		N	N	N	N
lwp_geterr() (3L)	N		N	N	N	N
lwp_getregs() (3L)	N		N	N	N	N
lwp_getstate() (3L)	N		N	N	N	N
lwp_join() (3L)	N		N	N	N	N
lwp_libcset() (3L)	N		N	N	N	N
lwp_newstk() (3L)	N		N	N	N	N
lwp_perror() (3L)	N		N	N	N	N

表 C-12 ライブラリルーチンのリファレンス: l3tol() ~ lwp_yield() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>lwp_ping()</code> (3L)	N		N	N	N	N
<code>lwp_resched()</code> (3L)	N		N	N	N	N
<code>lwp_resume()</code> (3L)	N		N	N	N	N
<code>lwp_self()</code> (3L)	N		N	N	N	N
<code>lwp_setpri()</code> (3L)	N		N	N	N	N
<code>lwp_setregs()</code> (3L)	N		N	N	N	N
<code>lwp_setstkcache()</code> (3L)	N		N	N	N	N
<code>lwp_sleep()</code> (3L)	N		N	N	N	N
<code>lwp_stkcswwset()</code> (3L)	N		N	N	N	N
<code>lwp_suspend()</code> (3L)	N		N	N	N	N
<code>lwp_yield()</code> (3L)	N		N	N	N	N

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>madd()</code> (3X)	S		N	N	N	N
<code>madvise()</code> (3)	S		N	N	N	N
<code>malloc()</code> (3)	S		S	S	S	N
<code>malloc_debug()</code> (3)	S		N	N	N	N
<code>malloc_verify()</code> (3)	S		N	N	N	N
<code>malloccmap()</code> (3)	S		N	N	N	N

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
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

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mktemp() (3)	C	SunOS 4 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 5.7、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 と、SunOS 5.7、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
mon_waiters() (3L)	N		N	N	N	N
moncontrol() (3)	A	profil() (2) に置換。	A	A	A	N
MONITOR() (3L)	N		N	N	N	N

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
monitor() (3)	C	<p>SunOS 4 と SunOS 5.7、ABI、SVID、または SVR4 との相違点は以下のとおり。SunOS 4 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 5.7、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 において、以前に定義済みの buf 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 5.7、ABI、SVID、または SVR4 monitor 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 prof(1) コマンドを使用すれば、結果が確認できる。</p>	C	C	C	N
monstartup() (3)	A	profil() (2) に置換。	A	A	A	N
mout() (3X)	S		N	N	N	N
move() (3V) -SysV	S		N	S	S	S
mrnd48() (3)	S		N	S	S	N

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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_RECVALL() (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 では有効、SunOS 5.7、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 では EPERM、SunOS 5.7、ABI、SVID、または SVR4 では EBUSY。	C	C	C	N
mtx() (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
mvcur() (3V)	C	SunOS 4 での戻り値は不定。SunOS 5.7、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-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mvcur() (3V) -SysV	S		N	S	S	N
mvdelch() (3V) -SysV	S		N	S	S	N
mvgetch() (3V) -SysV	C	SunOS 5.7、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 では、このような状況で wrefresh() は呼び出されない。	N	C	C	N
mvgetstr() (3V) -SysV	C	getstr() (3V) -SysV を参照。	N	C	C	N
mvinch() (3V) -SysV	S		N	S	S	N
mvinsch() (3V) -SysV	S		N	S	S	N
mvprintw() (3V)	C	wprintw() (3V) を参照。	N	C	C	S
mvprintw() (3V) -SysV	S		N	S	S	N
mvscanw() (3V)	C	wscanw() (3V) を参照。	N	C	C	S
mvscanw() (3V) -SysV	S		N	S	S	N
mvwaddch() (3V) -SysV	S		N	S	S	N
mvwaddstr() (3V) -SysV	S		N	S	S	N
mvwdelch() (3V) -SysV	S		N	S	S	N

表 C-13 ライブラリルーチンのリファレンス: madd() ~ mvwscanw() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
mvwgetch() (3V) -SysV	C	SunOS 5.7、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 では、このような状況で wrefresh() は呼び出されない。	N	C	C	N
mvwgetstr() (3V) -SysV	C	getstr() (3V) -SysV を参照。	N	C	C	N
mvwin() (3V)	C	SunOS 4 を使用すれば、サブウィンドウは移動可能。SunOS 5.7、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 mvderwin() を使用すること。	N	C	C	S
mvwin() (3V) -SysV	S		N	S	S	N
mvwinch() (3V) -SysV	S		N	S	S	N
mvwinsch() (3V) -SysV	S		N	S	S	N
mvwprintw() (3V)	C		N	C	C	S
mvwprintw() (3V) -SysV	S		N	S	S	N
mvwscanw() (3V)	C	wscanw() (3V) を参照。	N	C	C	S
mvwscanw() (3V) -SysV	S		N	S	S	N

表 C-14 ライブラリルーチンのリファレンス: napms() ~ ntohs()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
napms(3V) () (3V) -SysV	S		N	S	S	N
net_addr() (3N)	S		N	N	S	N
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 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 5.7、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 5.7 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

表 C-14 ライブラリルーチンのリファレンス: napms() ~ ntohs() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
nocbreak() (3V) -SysV	S		N	S	S	S
nocrmode() (3X)	S		N	N	S	N
nodelay() (3V) -SysV	S		N	S	S	N
noecho() (3V) -SysV	S		N	S	S	S
nonl() (3V) -SysV	S		N	S	S	S
nonstandard _arithmetic() (3M)	N		N	N	N	N
noraw() (3V) -SysV	S		N	S	S	S
notimeout() (3V) -SysV	S		N	S	S	N
nrand48() (3)	S		N	S	S	N
ntohl() (3N)	S		N	N	S	N
ntohs() (3N)	S		N	N	S	N

表 C-15 ライブラリルーチンのリファレンス: `on_exit()` ~ `overwrite()`

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>on_exit()</code> (3)		<code>atexit()</code> (3C) に置換。 <code>atexit()</code> 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
<code>opendir()</code> (3V)	C	SunOS 5.7、ABI、SVID、または SVR4 の DIR 構造体には、SunOS 4 の <code>dd_bsize</code> および <code>dd_off</code> フィールドが含まれない。また、SunOS 5.7、ABI、SVID または SVR4 の <code>dd_loc</code> および <code>dd_size</code> フィールドは、SunOS 4 の場合の long 型と違い int 型。 SunOS 5.7、ABI、SVID または SVR4 では、 <code><sys/types.h></code> に含まれるが、SunOS 4 では含まれない。SunOS 5.7、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 <code>errno</code> に <code>ENOENT</code> を設定。	C	C	C	N
<code>opendir()</code> (3V) -SysV	S		S	S	S	N
<code>openlog()</code> (3)	S		N	N	N	N
<code>openpl()</code> (3X)	N		N	N	N	N
<code>optarg()</code> (3)	S		N	N	N	N
<code>optind()</code> (3)	S		N	N	N	N
<code>overlay()</code> (3V)	C	SunOS 4 の <code>overlay()</code> は関数。SunOS 5.7、SVID または SVR4 の <code>overlay(srcwin, dstwin)</code> () は <code>_overlap(srcwin, (dstwin), TRUE)</code> () を呼び出すマクロ。 SunOS 4 では戻り値は不定。SunOS 5.7、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-15 ライブラリルーチンのリファレンス: `on_exit()` ~ `overwrite()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>overlay()</code> (3V) -SysV	S		N	S	S	N
<code>overwrite()</code> (3V)	C	SunOS 4 の <code>overwrite()</code> は関数。SunOS 5.7、SVID または SVR4 の <code>overwrite(srcwin, dstwin)()</code> は <code>_overlap((srcwin), (dstwin), FALSE)()</code> を呼び出すマクロ。 SunOS 4 では戻り値は不定。SunOS 5.7、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>overwrite()</code> (3V) -SysV	S		N	S	S	N

表 C-16 ライブラリルーチンのリファレンス: `passwd2des()` ~ `pwdauth()`

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>passwd2des()</code> (3R)	S		N	N	N	N
<code>pause()</code> (3V)	S		S	S	S	N
<code>pclose()</code> (3S)	S		S	S	S	N
<code>pechochar()</code> (3V) -SysV	S		N	S	S	N
<code>perror()</code> (3)	S		S	S	S	N
<code>plock()</code> (3)	S		S	S	S	N
<code>plot()</code> (3X)	S		N	N	N	N
<code>pmap_getmaps()</code> (3N)	S	利用可能。ただし、なるべく <code>rpcb_getmaps()</code> (3N) を使用する。	A	A	S	N

表 C-16 ライブラリルーチンのリファレンス: passwd2des() ~ pwdauth() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
pmap_set() (3N)	S	利用可能。ただし、なるべく rpcb_set() (3N) を使用す る。	A	A	S	N
pmap_unset() (3N)	S	利用可能。ただし、なるべく rpcb_unset() (3N) を使用す る。	A	A	S	N
pnoutrefresh() (3V) -SysV	S		N	S	S	N
pnp() (3R)	N		N	N	N	N
pod_getexit() (3L)	N		N	N	N	N
pod_getmaxpri() (3L)	N		N	N	N	N
pod_getmaxsize() (3L)	N		N	N	N	N
pod_setexit() (3L)	N		N	N	N	N
pod_setmaxpri() (3L)	N		N	N	N	N
point() (3X)	S		N	N	N	N
popen() (3S)	S		S	S	S	N

表 C-16 ライブラリルーチンのリファレンス: passwd2des() ~ pwdauth() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
pow() (3M)	C	SunOS 5.7、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 では 1 (SunOS 5.7、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 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 5.7、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
prefresh() (3V) -SysV	S		N	S	S	N
printf() (3V)	S	fprintf(3) を参照。	S	S	S	S

表 C-16 ライブラリルーチンのリファレンス: passwd2des() ~ pwdauth() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
printw() (3V)	C	SunOS 5.7、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 では void を返す。SunOS 5.7、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 では unsigned int 型、SVR4 および SunOS 5.7 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-17 ライブラリルーチンのリファレンス: qsort() ~ quiet_nan()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
qsort() (3)	S		S	S	S	N
quiet_nan() (3M)	N		N	N	N	N

表 C-18 ライブラリルーチンのリファレンス: rand() ~ rwall()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
rand() (3V)	S		S	S	S	S
random() (3)	A	現在は random() (3C)。SunOS 5.7、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

表 C-18 ライブラリルーチンのリファレンス: rand() ~ rwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
readdir() (3V)	C	SunOS 4、および ABI と SVID での dirent 構造体にも、共通に d_name フィールドがある。SunOS 4 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 5.7、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 5.7、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、SVR4、および SunOS 5.7 の dirent 構造体にも、共通に d_name および d_reclen フィールドがある。また、SunOS 5.7 の dd_loc および dd_size フィールドは、SunOS 4 の場合の long 型と違い int 型。	C	C	C	S
readdir() (3V) -SysV	C	SunOS 4、SVR4、および SunOS 5.7 の dirent 構造体にも、共通に d_name および d_reclen フィールドがある。また、SunOS 5.7 の dd_loc および dd_size フィールドは、SunOS 4 ソフトウェアの場合の long 型と違い int 型。SunOS 5.7、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、および ABI と SVID の dirent 構造体にも、共通に d_name フィールドがある。	C	C	C	N

表 C-18 ライブラリルーチンのリファレンス: rand() ~ rwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
realloc() (3)	C	SunOS 4 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 5.7、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 5.7 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 5.7 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
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) のマニュアルページを参照。	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

表 C-18 ライブラリルーチンのリファレンス: rand() ~ rwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
riporffline() (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

表 C-18 ライブラリルーチンのリファレンス: rand() ~ rwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
SAMECV() (3L)	N		N	N	N	N
SAMEMON() (3L)	N		N	N	N	N
SAMETHREAD() (3L)	N		N	N	N	N
saveterm() (3V)	A	def_prog_mode() (3X) に置換。	N	A	A	N
savetty() (3V) -SysV	S		N	S	S	S
scalb() (3M)	C	SunOS 5.7、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、ldexp() (3M) ルーチンと同等。オーバフローの場合は +HUGE または -HUGE (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 errno を設定する。SunOS 4 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

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
scalbn() (3M)	S		N	N	N	N
scandir() (3)	N		N	N	N	S
scanf() (3V)	S		S	S	S	N
scanw() (3V)	C	SunOS 5.7、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 では、異常終了すると ERR、正常終了すると OK (0) を返す。SunOS 5.7、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

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
scroll() (3V) -SysV	S		N	S	S	N
scrollok() (3V) -SysV	S		N	S	S	S
seconvert() (3)	S		N	N	N	N
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

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>_setjmp()</code> (3)	C	現在は <code>setjmp()</code> (3C)。 <code>sigsetjmp()</code> (3) <i>savemask</i> 引数がゼロのとき <code>sigsetjmp()</code> (3) 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、 <i>signalmask</i> は 保存しない。	A	A	A	S
<code>setjmp()</code> (3V)	S		S	S	S	S
<code>setkey()</code> (3)	S		N	S	S	N
<code>setlinebuf()</code> (3V)	S		N	N	N	S
<code>setlocale()</code> (3V)	C		S	S	S	N
<code>setlogmask()</code> (3)	S		N	N	N	N
<code>setmntent()</code> (3)	A	<code>fopen()</code> (3) に続けて <code>lockf()</code> (3) ルーチンを使用 するのと同様の機能を提供。	A	A	A	N
<code>setnetent()</code> (3N)	S		N	N	S	N
<code>setnetgrent()</code> (3N)	S		N	N	N	N
<code>setprotoent()</code> (3N)	S		N	N	S	N
<code>setpwaent()</code> (3)	N		N	N	N	N
<code>setpwent()</code> (3V)	S		S	S	S	N
<code>setpwfile()</code> (3V)	N		N	N	N	N
<code>setrgid()</code> (3V)	A	<code>setgid()</code> (2) に置換。	A	A	A	N
<code>setrpcent()</code> (3N)	S		N	N	S	N
<code>setruuid()</code> (3V)	A	<code>setuid()</code> (2) に置換。	A	A	A	N

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
setscrreg() (3V) -SysV	S		N	S	S	N
setservent() (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
setterm() (3V)	C	SunOS 4 および SunOS 5.7 の両方のリリースで setupterm() に置換される、古い呼び出し形式。 curs_terminfo() (3X) を参照。 setupterm(term, 1, (int *) 0) () は setterm(term) () 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 5.7 の tty システムに関する情報は、ttymon(1) および ttydefs(4) 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

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
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 のシグナルである SIGLOST は、SVR4 および SunOS 5.7 の signal() (2) ルーチンでは未定義。 SunOS 4 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-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
sin() (3M)	C	SunOS 5.7、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 (matherr() (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 errno に ERANGE が設定される。SunOS 4 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。<math.h> で定義されている fp_pi 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でも IEEE 754 に基づき生成される。	N	C	C	N
sinh() (3M)	S		N	S	S	N
single_precision() (3M)	N		N	N	N	N
single_to_decimal() (3)	S		N	N	N	N
sleep() (3V)	S		S	S	S	S
slk_clear() (3V) -SysV	S		N	S	S	N
slk_init() (3V) -SysV	S		N	S	S	N
slk_label() (3V) -SysV	S		N	S	S	N
slk_noutrefresh() (3V) -SysV	S		N	S	S	N
slk_refresh() (3V) -SysV	S		N	S	S	N
slk_restore() (3V) -SysV	S		N	S	S	N

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	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 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 5.7、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 ソフトウェアでは int 型、SunOS 5.7、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 5.7、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-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	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
STKTOP() (3L)	N		N	N	N	N
store() (3X)	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-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
strftime() (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。</p> <p>%k と %l は SunOS 5.7 ではサポートされない。%s は、秒を SunOS 4 では 0~59 の範囲内で指定。SunOS 5.7 では 0~61 の範囲内 (うるう秒を許容する) で定義。%U と %W は、週番号を指定する場合、SunOS 4 における 01 は 4 日以上を含む 1 月の最初の週を示す。SunOS 5.7 における 01 は、%U は日曜日から、%W は月曜日から始まる 1 月の最初の週を示す。</p> <p>SunOS 4.1 の tm 構造体には、SunOS 5.7 の tm 構造体に存在しない 2 つのフィールド、tm_zone および tm_gmtoff がある。そのかわり SunOS 5.7 では、GMT と地方標準時間との差 (単位は秒) が外部変数 <i>timezone</i>、夏時間を適用するかどうかを外部変数 <i>daylight</i> 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 <i>tzname</i> に格納される。これらの外部変数 (<i>timezone</i>、<i>daylight</i>、および <i>tzname</i>) は、SunOS 4 の System V インストールオプションである <i>ctime() (3V)</i> ライブラリルーチンによりサポートされる。</p>	C	C	C	N
string_to_decimal() (3)	N		N	N	N	N
strlen() (3)	S		S	S	S	N
strncasecmp() (3)	S		N	N	N	N

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
strncat() (3)	S		S	S	S	N
strncmp() (3)	S		S	S	S	N
strncpy() (3)	S		S	S	S	N
strpbrk() (3)	S		S	S	S	N
strptime() (3V)	S		A	A	A	N
strrchr() (3)	S		S	S	S	N
strspn() (3)	S		S	S	S	N
strtod() (3)	C	SunOS 4 の strtod() および atof() 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 5.7、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
strtok() (3)	S		S	S	S	N
strtol() (3)	S		S	S	S	N
strxfrm() (3)	S		S	S	S	N
stty() (3C)	A	termio(7) インタフェースが同様の機能を提供。	A	A	A	N
subpad() (3V) -SysV	S		N	S	S	N

表 C-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
subwin() (3V)	C	SunOS 5.7、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 5.7、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_getrpccaller() (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-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	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-19 ライブラリルーチンのリファレンス: SAMECV() ~ system() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
svcudp_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

表 C-20 ライブラリルーチンのリファレンス: t_accept() ~ tzsetwall()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
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

表 C-20 ライブラリルーチンのリファレンス: `t_accept()` ~ `tzsetwall()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>t_listen()</code> (3N)	S		S	S	S	N
<code>t_look()</code> (3N)	S		S	S	S	N
<code>t_open()</code> (3N)	S		S	S	S	N
<code>t_optmgmt()</code> (3N)	S		S	S	S	N
<code>t_rcv()</code> (3N)	S		S	S	S	N
<code>t_rcvconnect()</code> (3N)	S		S	S	S	N
<code>t_rcvdis()</code> (3N)	S		S	S	S	N
<code>t_rcvrel()</code> (3N)	S		S	S	S	N
<code>t_rcvudata()</code> (3N)	S		S	S	S	N
<code>t_rcvuderr()</code> (3N)	S		S	S	S	N
<code>t_snd()</code> (3N)	S		S	S	S	N
<code>t_snddis()</code> (3N)	S		S	S	S	N
<code>t_sndrel()</code> (3N)	S		S	S	S	N
<code>t_sndudata()</code> (3N)	S		S	S	S	N
<code>t_sync()</code> (3N)	S		S	S	S	N
<code>t_unbind()</code> (3N)	S		S	S	S	N
<code>tan()</code> (3M)	S		N	S	S	N
<code>tanh()</code> (3M)	S		N	S	S	N
<code>tcdrain()</code> (3V)	S		S	S	S	N
<code>tcflow()</code> (3V)	S		S	S	S	N

表 C-20 ライブラリルーチンのリファレンス: t_accept() ~ tzsetwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 5.7、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 5.7 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
tgetent() (3V) -SysV	S		N	S	S	N

表 C-20 ライブラリルーチンのリファレンス: t_accept() ~ tzsetwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
tgetflag() (3X)	C	SunOS 5.7、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 5.7 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 5.7、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 5.7 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 5.7、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 5.7 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-20 ライブラリルーチンのリファレンス: t_accept() ~ tzsetwall() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
tgoto() (3X)	C	SunOS 5.7、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 5.7 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 は、1/HZ 秒の単位で時間値を返す (HZ は 60)。SunOS 5.7、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) -SysV	S		S	S	S	N

表 C-20 ライブラリルーチンのリファレンス: `t_accept()` ~ `tzsetwall()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>tolower()</code> (3V)	S		S	S	S	N
<code>tolower()</code> (3V) -SysV	C	SunOS 5.7、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 では影響なし。	C	C	C	N
<code>touchline()</code> (3V)	C	SunOS 5.7、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 5.7、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) -SysV	S		S	S	S	N
<code>toupper()</code> (3V)	S		S	S	S	N
<code>toupper()</code> (3V) -SysV	C	SunOS 5.7、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 では影響なし。	C	C	C	N
<code>tparam()</code> (3V) -SysV	S		N	S	S	N

表 C-20 ライブラリルーチンのリファレンス: `t_accept()` ~ `tzsetwall()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>tputs()</code> (3V)	C	SunOS 5.7、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 5.7 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
<code>traceoff()</code> (3V) -SysV	S		N	N	S	N
<code>traceon()</code> (3V) -SysV	S		N	N	S	N
<code>tsearch()</code> (3)	S		S	S	S	N
<code>ttynname()</code> (3V)	S		S	S	S	N
<code>ttyslot()</code> (3V)	S		N	N	S	N
<code>twalk()</code> (3)	S		S	S	S	N
<code>typeahead()</code> (3V) -SysV	S		N	S	S	N
<code>tzset()</code> (3V)	C	<code>ctime()</code> (3V) を参照。	C	C	C	N
<code>tzsetwall()</code> (3V)	A	<code>tzset()</code> (3C) に置換。	A	A	A	N

表 C-21 ライブラリルーチンのリファレンス: ualarm() ~ utime()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
ualarm() (3)	S	現在は ualarm() (3C)。ITIMER_REAL に引数 <i>which</i> を設定した setitimer() (2) システムコールが同様の機能を提供。	N	A	A	S
ulimit() (3C)	S	SVR4 および SunOS 5.7 の ulimit() は、SunOS 4 の ulimit() と互換性がある。SunOS 4 の ulimit() ルーチンの整数 <i>cmd</i> の値 1 および 2 は同等な SVID の ulimit() ルーチンのシンボリック定数 <i>cmd</i> の値 UL_GETFSIZE および UL_SETFSIZE と互換性のない場合がある。また、SVID の ulimit() 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
unctrl() (3V) -SysV	S		N	S	S	S
ungetc() (3S)	S	SVR4 および SunOS 5.7 の ungetc() は必ず 4 文字をバッファにプッシュするため、SunOS 4 の ungetc() と互換性がある。SunOS 4 の ungetc() は、前に read 文がなくても必ず標準入力に 1 文字をバッファにプッシュするが、ABI および SVID の ungetc() はこの属性をサポートしていない。	C	C	S	N
ungetch() (3V) -SysV	S		N	S	S	N
user2netname() (3N)	S		S	S	S	N

表 C-21 ライブラリルーチンのリファレンス: ualarm() ~ utime() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
usleep() (3)	S	現在は usleep() (3C)。 setitimer() (2) または select() (3C) が同様の機能 を提供。	N	A	A	S
utime() (3V)	C	SunOS 4 と、SunOS 5.7、 ABI、SVID、または SVR4 で は、第 2 引数の型が異なる。 SunOS 4 では、引数 <i>timep</i> は 2 つの <i>time_t</i> 値の配列を指す が、SunOS 5.7、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

表 C-22 ライブラリルーチンのリファレンス: valloc() ~ vwscanw()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
valloc() (3)	S		N	N	S	N
varargs() (3)	S		N	N	N	N
vfprintf() (3V)	C	vprintf() (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	getrlimit() (2) に置換。	A	A	A	N

表 C-22 ライブラリルーチンのリファレンス: `valloc()` ~ `vwscanw()` 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
<code>vprintf()</code> (3V)	C	<code>vprintf()</code> 、 <code>vfprintf()</code> 、および <code>vsprintf()</code> 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 と SunOS 5.7、ABI、SVID、または SVR4 との互換性はない。SunOS 4 では、関数ヘッダで <code>va_list</code> (<code><varargs.h></code> で定義されている) を使用し、可変引数リスト (たとえば <code>void function(va_list)</code> など) を宣言する。SunOS 5.7、ABI、SVID、または SVR4 では、 <code><stdarg.h></code> 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 <code>void function(int arg1)</code> など) を宣言する。	C	C	C	S
<code>vsprintf()</code> (3V)	C	<code>vprintf()</code> (3V) を参照。	C	C	C	S
<code>syslog()</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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout()

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
waddch() (3V)	C	CHTYPE が < curses.h> で別に定義されていない場合は、SunOS 5.7 における文字は chtype (long)。SunOS 5.7、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。SunOS 5.7、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
waddch() (3V) -SysV	S		N	S	S	N
waddstr() (3V)	C	SunOS 5.7、SVID または SVR4 では、異常終了すると ERR (-1) を返す。SunOS 5.7、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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
wclear() (3V)	C	SunOS 5.7、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4 は void を返す。SunOS 5.7、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
wclrtoobot() (3V)	C	SunOS 5.7、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4 では void を返す。SunOS 5.7、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
wclrtoobot() (3V) -SysV	S		N	S	S	N
wclrtoeol() (3V)	C	SunOS 5.7、SVID または SVR4 では、正常終了すると常に (OK = 0) を返すが、SunOS 4 では void 返す。SunOS 5.7、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-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
wclrtoeol() (3V) -SysV	S		N	S	S	N
wcstombs() (3)	S	wchar_t のサイズは SunOS 4 は短く、SunOS 5.7 は長い。	S	S	S	N
wctomb() (3)	S	wchar_t のサイズは SunOS 4 は短く、SunOS 5.7 は長い。	S	S	S	N
wdelch() (3V)	C	SunOS 5.7、SVID または SVR4 ではマクロの場合があるが、SunOS 4 では常にマクロ。 SunOS 5.7、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 5.7、SVID または SVR4 ではマクロの場合があるが、SunOS 4 では常にマクロ。 SunOS 5.7、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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
werase() (3V)	C	SunOS 5.7、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 5.7、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
wgetch() (3V)	C	SunOS 5.7、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 5.7、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 5.7、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 5.7、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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
winsch() (3V)	C	SunOS 5.7、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 5.7、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 5.7、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
wmove() (3V)	C	SunOS 5.7、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 5.7、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 では void を返す。SunOS 5.7、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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
wrefresh() (3V)	C	SunOS 5.7、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 では void を返す。SunOS 5.7、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
wrefresh() (3V) -SysV	S		N	S	S	N
wscanw() (3V)	C	SunOS 5.7、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 では void を返す。SunOS 5.7、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

表 C-23 ライブラリルーチンのリファレンス: waddch() ~ wstandout() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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 では常に不定値を返す。一方、SunOS 5.7、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
wstandout() (3V)	C	このルーチンは、ウィンドウに対して最適な standout モードの端末を有効にするため A_STANDOUT 属性を設定する curses() (3V) 関数。SunOS 4 では、この関数に attron(A_STANDOUT)() を使用し、不定値を返す。SunOS 5.7、SVID または SVR4 の standout() ルーチンは attron(A_STANDOUT)() 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

表 C-24 ライブラリルーチンのリファレンス: xcrypt() ~ xtom()

SunOS 4	SunOS 5.7	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

表 C-24 ライブラリルーチンのリファレンス: xcrypt() ~ xtom() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
xdr_authunix_parms() (3N)	S	利用可能。 ただし SunOS 5.7、ABI、 SVID または SVR4 ではなるべく xdr_authsys_parms() (3N) を 使用する	A	A	A	N
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

表 C-24 ライブラリルーチンのリファレンス: xcrypt() ~ xtom() 続く

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
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
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

表 C-25 ライブラリルーチンのリファレンス: $y_0()$ ~ $y_n()$

SunOS 4	SunOS 5.7	注	ABI	SVID	SVR4	BSD
$y_0()$ (3M)	C	SunOS 4 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 5.7、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 において $y_0(\text{HUGE}())$, $y_1(\text{HUGE}())$, $y_n(9, \text{HUGE}())$ 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
$y_1()$ (3M)	C	SunOS 4 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 5.7、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 において $y_0(\text{HUGE}())$, $y_1(\text{HUGE}())$, $y_n(9, \text{HUGE}())$ 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
$y_n()$ (3M)	C	SunOS 4 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 5.7、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 において $y_0(\text{HUGE}())$, $y_1(\text{HUGE}())$, $y_n(9, \text{HUGE}())$ 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 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 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 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 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、『*man pages section 4: 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: a.out ~ auto.master

SunOS 4	SunOS 5.7	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	実行アカウントティングファイル
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

表 D-2 ファイルフォーマットのリファレンス: bar ~ bootparams

SunOS 4	SunOS 5.7	注
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

表 D-3 ファイルフォーマットのリファレンス: cpio ~ crontab

SunOS 4	SunOS 5.7	注
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

表 D-4 ファイルフォーマットのリファレンス: dir ~ dump

SunOS 4	SunOS 5.7	注
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

表 D-5 ファイルフォーマットのリファレンス: environ ~ exports

SunOS 4	SunOS 5.7	注
environ(5V)	C	ユーザ環境
ethers(5)	S	<i>hostname</i> 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

表 D-6 ファイルフォーマットのリファレンス: fbtab ~ ftpusers

SunOS 4	SunOS 5.7	注
fbtab(5)	C	フレームバッファテーブル
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
ftpusers(5)	S	FTP によって禁止されたユーザのリスト

表 D-7 ファイルフォーマットのリファレンス: gettytab ~ group.adjunct

SunOS 4	SunOS 5.7	注
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

表 D-8 ファイルフォーマットのリファレンス: holidays ~ hosts.equiv

SunOS 4	SunOS 5.7	注
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

表 D-9 ファイルフォーマットのリファレンス: indent.pro ~ internat

SunOS 4	SunOS 5.7	注
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

表 D-10 ファイルフォーマットのリファレンス: keytables

SunOS 4	SunOS 5.7	注
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

表 D-11 ファイルフォーマットのリファレンス: link ~ locale

SunOS 4	SunOS 5.7	注
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

表 D-12 ファイルフォーマットのリファレンス: magic ~ mtab

SunOS 4	SunOS 5.7	注
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

表 D-13 ファイルフォーマットのリファレンス: netgroup ~ networks

SunOS 4	SunOS 5.7	注
netgroup (5)	S	ネットワークグループのリスト
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

表 D-14 ファイルフォーマットのリファレンス: passwd ~ publickey

SunOS 4	SunOS 5.7	注
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

表 D-14 ファイルフォーマットのリファレンス: passwd ~ publickey 続く

SunOS 4	SunOS 5.7	注
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

表 D-15 ファイルフォーマットのリファレンス: queuedefs

SunOS 4	SunOS 5.7	注
queuedefs (5)	S	at、batch、および cron 用の待ち行列記述ファイル

表 D-16 ファイルフォーマットのリファレンス: rasterfile ~ rpc

SunOS 4	SunOS 5.7	注
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

表 D-16 ファイルフォーマットのリファレンス: rasterfile ~ rpc 続く

SunOS 4	SunOS 5.7	注
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-17 ファイルフォーマットのリファレンス: sccsfile ~ systems

SunOS 4	SunOS 5.7	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

表 D-18 ファイルフォーマットのリファレンス: tar ~ tzfile

SunOS 4	SunOS 5.7	注
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

表 D-18 ファイルフォーマットのリファレンス: tar ~ tzfile 続く

SunOS 4	SunOS 5.7	注
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

表 D-19 ファイルフォーマットのリファレンス: updaters ~ uuencode

SunOS 4	SunOS 5.7	注
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
uuencode(5)	S	コード化された uuencode ファイルのフォーマット

表 D-20 ファイルフォーマットのリファレンス: vfont ~ vgrindefs

SunOS 4	SunOS 5.7	注
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

表 D-21 ファイルフォーマットのリファレンス: xtab

SunOS 4	SunOS 5.7	注
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

表 D-22 ファイルフォーマットのリファレンス: ypaliases ~ ypprintcap

SunOS 4	SunOS 5.7	注
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 5.7 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的マウントポイント
/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リング用ディレクトリ
/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/nterm	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/ucb/lib	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	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 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	Solaris 7	説明
<code>ps -ax</code>	<code>ps -ef</code>	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
<code>pstat -s</code>	<code>swap -s</code>	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	Solaris 7	説明
<code>dump</code>	<code>ufsdump</code>	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
<code>exportfs</code>	<code>share resources</code>	<code>/etc/dfs/dfstab</code> 内に登録された特定のリソースのリモートマウントを可能にする
<code>exportfs -a</code>	<code>shareall</code>	<code>/etc/dfs/dfstab</code> 内に登録されたすべてのリソースのマウントを可能にするオプション
<code>exportfs -u</code>	<code>unshare resource</code>	リソースを利用できないようにする
<code>mount -a</code>	<code>mountall</code>	<code>mountall</code> オプションが設定された <code>/etc/vfstab</code> 内のすべてのファイルシステムをマウントする
<code>restore</code>	<code>ufsrestore</code>	バックアップ媒体にダンプしたファイルを復元する
<code>showmount -d</code>	<code>dfmounts option</code>	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
<code>showmount -e</code>	<code>dfshares option</code>	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
<code>umount -a</code>	<code>umountall</code>	<code>/etc/vfstab</code> 内のすべてのファイルシステム (ルート、 <code>/proc</code> 、 <code>/var</code> 、 <code>/usr</code> 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	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/mntab	/etc/mnttab	/etc/mount コマンドが読み取る現在マウントされているリソースのリスト
/etc/xtab	/etc/dfs/sharetab	共用可能なリソースのリスト
/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	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	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	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
dket1	利用できない
dkinfo	prvtoc
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	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	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	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.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
update	fsflush

表 F-5 ファイルおよびコマンドの相違 続く

SunOS 4	Solaris 7
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 5.7 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 5.7 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 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 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 5.7 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 5.7 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 419
- /5bin ディレクトリ 116
- /5include ディレクトリ 116
- /5lib ディレクトリ 116, 191 - 193
- 32 ビットの実行環境 30
- 64 ビットの実行環境 30
- 64 ビットライブラリ 172

A

- a641 ライブラリルーチン 317
- abort ライブラリルーチン 317
- abs ライブラリルーチン 317
- accept システムコール 285
- access システムコール 285
- accounting システム
 - 構成情報ディレクトリ 416
- accounting ファイル
 - Service Access Facility (SAF) 418
 - システム 417
 - スクリプトとバイナリ 420
- acctcms コマンド 230
- acctcom コマンド 230
- acctcon1 コマンド 230
- acctcon2 コマンド 230
- acctdisk コマンド 230
- acctdusg コマンド 230, 242
- acctmerg コマンド 230
- accton コマンド 230
- acctprc1 コマンド 230
- acctprc2 コマンド 230
- acctwtmp コマンド 230
- acct コマンド 269
- acct システムコール 285
- acct システムファイル 406
- /acct ディレクトリ 416, 420
- acosh ライブラリルーチン 317
- acos ライブラリルーチン 317
- ac コマンド 230
- adbgen コマンド 230
- adb コマンド 180, 182, 230, 231
- addbib コマンド 231
- addch ライブラリルーチン 317
- add_client コマンド 230
- add_drv コマンド 94, 212, 214, 225
- addexportent ライブラリルーチン 317
- addmntent ライブラリルーチン 317
- add_services コマンド 231
- addstr ライブラリルーチン 317
- adjacentscreens コマンド 231
- adjtime システムコール 285
- Administration ツール
 - files ディレクトリ 422
- admintool
 - Serial Port Manager 145
 - 描写 29, 44
 - ユーザアカウント 76
 - ユーザとグループの管理 75
- admintool コマンド 44
 - add_client コマンド vs. 230
 - files ディレクトリ 422
 - rm_client コマンド vs. 267
- admin コマンド 198, 231
- /.Admin ディレクトリ 154

/adm ディレクトリ 417
adv コマンド 231
aedplot コマンド 231
aging パスワード 67
agt_create ライブラリルーチン 317
agt_enumerate ライブラリルーチン 317
agt_trap ライブラリルーチン 318
aint ライブラリルーチン 318
aiocancel ライブラリルーチン 318
aioread ライブラリルーチン 318
aiowait ライブラリルーチン 318
aiowrite ライブラリルーチン 318
alarm ライブラリルーチン 318
aliasadm コマンド 43
aliases システムファイル 406, 425
align_equals コマンド 231
alloca ライブラリルーチン 318
alphasort ライブラリルーチン 318
analyze コマンド 231
anint ライブラリルーチン 318
annuity ライブラリルーチン 318
ANSI C コンパイラ 171
AnswerBook2 29
a.out システムファイル 117, 406
apropos コマンド 231
arch コマンド 232
arc ライブラリルーチン 318
arp コマンド 232
ar コマンド 198, 231, 265
ar システムファイル 406
asctime ライブラリルーチン 318
ASET (自動セキュリティ拡張ツール) 29, 69
asinh ライブラリルーチン 318
asin ライブラリルーチン 318
assert ライブラリルーチン 319
async_daemon システムコール 285
as コマンド 198, 232
AT&T システム
 Solaris オペレーティング環境に含まれない SVR4 の機能 31
 sysadm メニューユーティリティ 31
 デバイスの命名 37
atan2 ライブラリルーチン 319
atanh ライブラリルーチン 319
atan ライブラリルーチン 319
atexit ライブラリルーチン 362
atof ライブラリルーチン 319

atoi ライブラリルーチン 319
atol ライブラリルーチン 319
atoplot コマンド 232
atq コマンド 232
atrm コマンド 232
attroff ライブラリルーチン 319
attron ライブラリルーチン 319
attrset ライブラリルーチン 319
at コマンド 232, 253, 418
audioio ioctl 188
audit_args ライブラリルーチン 319
audit_control システムファイル 406
audit_data システムファイル 406
auditd コマンド 233
audit.log システムファイル 406
auditon システムコール 285
auditsvc システムコール 285
audit_text ライブラリルーチン 319
audit_warn コマンド 233
audit コマンド 233
audit システムコール 285
authdes_create ライブラリルーチン 319
authdes_getucrcd ライブラリルーチン 320
authdes_seccreate ライブラリルーチン 319
auth_destroy ライブラリルーチン 320
authnone_create ライブラリルーチン 320
authsys_create_default ライブラリルーチン 320
authsys_seccreate ライブラリルーチン 320
authunix_create_default ライブラリルーチン 320
authunix_create ライブラリルーチン 320
AutoFS 42
autofs プログラム 41, 119, 120
auto.home システムファイル 233, 406
auto.master システムファイル 233, 406
automount コマンド 233
autopush コマンド 224
awk コマンド 233

B

Backup CoPilot 51, 125
banner コマンド 233
bar コマンド 125, 233
bar システムファイル 406
basename コマンド 234

batch コマンド 234
baudrate ライブラリルーチン 320
bcmp ライブラリルーチン 320
bcopy ライブラリルーチン 320
bc コマンド 234
beep ライブラリルーチン 320
Berkeley 互換パッケージ 422
BFS (ブートファイルシステム) 106
bgplot コマンド 234
biff コマンド 234
bindresvport ライブラリルーチン 320
bindtextdomain ライブラリルーチン 206
bind システムコール 286
bin-mail コマンド 234
biod コマンド 234, 426
block I/O デーモン 426
boards.pc システムファイル 406
bootblk コマンド 212
bootparamd コマンド 234
bootparams データベース 406
bootparam ライブラリルーチン 320
bootsd コマンド 93
boot.sun4c.sunos.4.1 コマンド 94, 212
boot コマンド
 再構成ブート 213
 デバイスの管理 88
 デバイスの追加と 92
 デバイスを追加 87
 の変更 92, 93, 211, 212, 234
Bourne シェル 71 - 73, 271
 制限つき 268
 制限付きシェル 67
box ライブラリルーチン 321
brk システムコール 286
BSD ソース互換パッケージ 61, 62
bsearch ライブラリルーチン 321
byteorder ライブラリルーチン 321
bzero ライブラリルーチン 321

C

C2conv コマンド 234
C2unconv コマンド 234
CACHEFS (キャッシュファイルシステム) 100, 105
calendar コマンド 235
calloc ライブラリルーチン 321

callrpc ライブラリルーチン 321
cal コマンド 234
cancel コマンド 46, 143, 235, 255, 423
capitalize コマンド 235
captaininfo コマンド 235
catclose ライブラリルーチン 322
catgetmsg ライブラリルーチン 207, 322
catgets ライブラリルーチン 206, 322
catman コマンド 235
catopen ライブラリルーチン 322
cat コマンド 235
cbc_crypt ライブラリルーチン 322
cbreak ライブラリルーチン 322, 325
cbrt ライブラリルーチン 322
cb コマンド 201, 235
cc コマンド 201, 235
cdc コマンド 198, 235
/cdrom ディレクトリ 89, 121
CD-ROM デバイス
 install4x プログラムと 133, 135
 Solaris 7 要求 50
 管理 88, 90
 性能の改善 100, 105
 マネージャ 48
 命名規則 85
CD-ROM ファイルシステム (HSFS) 88, 104, 121
cd コマンド 235
ceil ライブラリルーチン 322
cfgetispeed ライブラリルーチン 322
cfgetospeed ライブラリルーチン 322
cflow コマンド 201, 235
cfree ライブラリルーチン 322
cfsetispeed ライブラリルーチン 322
cfsetospeed ライブラリルーチン 322
chargefee コマンド 235
chdir システムコール 286
checkeq コマンド 236
checknr コマンド 236
CHECK ライブラリルーチン 322
chfn コマンド 236
chgrp コマンド 236
chkey コマンド 236
chmod コマンド 234, 236
chmod システムコール 286
chown コマンド 236

chown システムコール 286
chroot システムコール 287
chroot ライブラリルーチン 206, 236
chrtbl コマンド 236
chsh コマンド 236
circle ライブラリルーチン 322
ckpacct コマンド 236
/class ディレクトリ 420
clear_colormap コマンド 237
clearerr ライブラリルーチン 322
clear_function コマンド 237
clearok ライブラリルーチン 323
clear コマンド 237
clear ライブラリルーチン 322
click コマンド 237
clnt_broadcast ライブラリルーチン 323
clnt_call ライブラリルーチン 323
clnt_control ライブラリルーチン 323
clnt_create ライブラリルーチン 323, 324
clnt_destroy ライブラリルーチン 323
clnt_dg_create ライブラリルーチン 324
clnt_freeres ライブラリルーチン 323
clnt_geterr ライブラリルーチン 323
clnt_pcreateerror ライブラリルーチン 323
clnt_perrno ライブラリルーチン 323
clnt_perror ライブラリルーチン 323
clntraw_create ライブラリルーチン 323
clnt_screateerror ライブラリルーチン 323
clnt_spermo ライブラリルーチン 323
clnt_sperror ライブラリルーチン 323
clnttcp_create ライブラリルーチン 324
clnt_tli_create ライブラリルーチン 324
clntudp_bufcreate ライブラリルーチン 324
clntudp_create ライブラリルーチン 324
clnt_vc_create ライブラリルーチン 324
clock コマンド 237
clock ライブラリルーチン 324
closedir ライブラリルーチン 324
closelog ライブラリルーチン 324
closepl ライブラリルーチン 324
close システムコール 287
clri コマンド 107, 237
clrtobot ライブラリルーチン 324
clrtoeol ライブラリルーチン 324
cmdtool コマンド 237
cmp コマンド 237
COFF 200, 201

colcrt コマンド 237
colldef コマンド 237
colltbl コマンド 237
coloredit コマンド 237
colrm コマンド 237
col コマンド 237
comb コマンド 198, 237
comm コマンド 237
compound ライブラリルーチン 324
compress コマンド 238
config コマンド 94, 210, 212, 238
Config ファイル 154
connect システムコール 287
cont ライブラリルーチン 324
convert4x プログラム 132, 138
copyright ファイル 196
copysign ライブラリルーチン 325
copywin ライブラリルーチン 325
cosh ライブラリルーチン 325
cos ライブラリルーチン 325
cpio コマンド 127
 bar コマンド vs. 233
 paxcpio コマンド vs. 262
 pax コマンド vs. 262
 説明 126, 127, 129
 のサポート 125
 の変更 128, 129, 238
cpio システムファイル 406
cpp コマンド 198, 238
cp コマンド 238
crash コマンド 238
/crash ディレクトリ 417
creat システムコール 287
crmode ライブラリルーチン 325
/cron.d ディレクトリ 416
crontab コマンド 238
crontab システムファイル 406
cron コマンド 238, 253, 418
/cron ディレクトリ 418
crtplot コマンド 238
crypt コマンド 238
_crypt ライブラリルーチン 325, 326
cscope コマンド 201
.cshrc ファイル 73
csh コマンド 71 - 73, 238
csplit コマンド 238

ctags コマンド 238
 ctermid ライブラリルーチン 326
 ctime ライブラリルーチン 327
 ctrace コマンド 201, 239
 curs_set ライブラリルーチン 327
 cuserid ライブラリルーチン 327
 cut コマンド 239
 cu コマンド 239
 cv_broadcast ライブラリルーチン 327
 cv_create ライブラリルーチン 327
 cv_destroy ライブラリルーチン 327
 cv_enumerate ライブラリルーチン 327
 cv_notify ライブラリルーチン 328
 cv_send ライブラリルーチン 328
 cv_waiters ライブラリルーチン 328
 cv_wait ライブラリルーチン 328
 cxref コマンド 201, 239
 C コンパイラ 171, 201
 C コンパイルシステム 419
 C コンパイルシステム, ディレクトリ 116
 C シェル 71 - 73, 238
 C プログラムツール 201

D

Data Link Provider Interface (DLPI) 204
 date コマンド 206, 239
 dbconfig コマンド 239
 dbm_clearerr ライブラリルーチン 328
 dbm_close ライブラリルーチン 328
 dbm_delete ライブラリルーチン 328
 dbm_error ライブラリルーチン 328
 dbm_fetch ライブラリルーチン 328, 335
 dbm_firstkey ライブラリルーチン 328, 335
 dbm_init ライブラリルーチン 328
 dbm_nextkey ライブラリルーチン 328, 360
 dbm_open ライブラリルーチン 328
 dbm_store ライブラリルーチン 328, 379
 dbxtool コマンド 180, 201, 240
 dbx コマンド 180, 201, 239
 -dcheck コマンド 240
 dc コマンド 240
 ddi_create_minor_node コマンド 213
 DDI/DKI 準拠
 定義 219
 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) 30, 218, 219, 225

DDI-準拠デバイスドライバ 219
 DDI (デバイスドライバインタフェース) 30,
 218, 224
 dd コマンド 124, 126, 240
 decimal_to_double ライブラリルーチン 328
 decimal_to_extended ライブラリルーチ
 ン 329
 decimal_to_floating ライブラリルーチン 329
 decimal_to_single ライブラリルーチン 329
 defaultsedit コマンド 240
 defaults_from_input コマンド 240
 defaults_merge コマンド 240
 defaults_to_indentpro コマンド 240
 defaults_to_mailrc コマンド 240
 def_prog_mode ライブラリルーチン 329, 371
 def_shell_mode ライブラリルーチン 329
 delay_output ライブラリルーチン 329
 delch ライブラリルーチン 329
 del_curterm ライブラリルーチン 329
 deleteln ライブラリルーチン 329
 delete ライブラリルーチン 329
 delta コマンド 198, 240
 delwin ライブラリルーチン 329
 /demo ディレクトリ 419
 deroff コマンド 241
 des_crypt ライブラリルーチン 329
 DeskSet 24, 27
 des_setparity ライブラリルーチン 329
 des コマンド 241
 /dev/dsk ディレクトリ 110, 215, 415
 /dev/devices ディレクトリ 213, 214
 devinfo コマンド 85, 87, 220, 221, 241
 devinfo ツリー 224
 dev_info ノード 224
 /dev/ksyms ファイル 182
 devnm コマンド 241
 /dev/pts ディレクトリ 110, 415
 /dev/rdisk ディレクトリ 110, 215, 415
 /dev/rmt ディレクトリ 110, 415
 /dev/sad ディレクトリ 110, 415
 /dev/sd1g コマンド 94
 /dev/term ディレクトリ 110, 416
 /dev ディレクトリ
 /devices ディレクトリと 214
 記述 215, 415
 変更点 38, 83, 94, 110, 415

dfmounts コマンド 120, 424
dfshares コマンド 120, 424
DFS (分散ファイルシステム) の管理 150
df コマンド 86, 107, 121, 241
dgettext ライブラリルーチン 206
/dict ディレクトリ 420
diff3 コマンド 241
diffmk コマンド 241
diff コマンド 241
dircmp コマンド 242
directories
 クイックリファレンス 425, 426
dirname コマンド 242
dir システムファイル 407
discover4x プログラム 132, 133
diskusg コマンド 242
dispadmin コマンド 420
dis コマンド 200, 242
dkctl コマンド 242
dkinfo コマンド 52, 85, 86, 242
dkio ioctl 185, 187
DKI (ドライバカーネルインタフェース) 30,
 218
dlclose ライブラリルーチン 329
dlerror ライブラリルーチン 329
dlopen ライブラリルーチン 329
DLPI (Data Link Provider Interface) 204
dlsym ライブラリルーチン 329
dmesg コマンド 242
dname コマンド 242
dn_comp ライブラリルーチン 329
dn_expand ライブラリルーチン 329
DNS (ドメインネームシステム)
 NIS+ (Network Information Services Plus)
 vs. 156, 157
 説明 156
dodisk コマンド 242
domainname コマンド 242
dorfs コマンド 242
dos2unix コマンド 242
double_to_decimal ライブラリルーチン 330
doupdate ライブラリルーチン 330
draino ライブラリルーチン 330
drand48 ライブラリルーチン 330, 345, 367,
 375
drvconfig プログラム 213
/drv ディレクトリ 117

/dsk ディレクトリ 110, 215, 415
dumbplot コマンド 242
dumpadm コマンド 243
dumpfs コマンド 243
dumpkeys コマンド 243
dump コマンド
 新しいコマンド 200
 クイックリファレンステーブル 424
 変更点 125, 243
 リンクのチェックを使用する 178, 179
dump システムファイル 407
dup2 システムコール 288
dup システムコール 288
du コマンド 86, 121, 242
dysize ライブラリルーチン 330

E

ecb_crypt ライブラリルーチン 330
echochar ライブラリルーチン 330
echo コマンド 243
echo ライブラリルーチン 330
econvert ライブラリルーチン 330
ecvt ライブラリルーチン 330
edata ライブラリルーチン 330
edit コマンド 244
edquota コマンド 244
ed コマンド 244
eeprom コマンド 244
EFT (Extended Fundamental Types) 51
egrep コマンド 244
eject コマンド 244
ELF (executable and linking format) ファイル
 コンパイラ 171
 実行するためのカーネルモジュール 117
 リンカ 172
encrypt ライブラリルーチン 330
endac ライブラリルーチン 330
endexportent ライブラリルーチン 330
endfsent ライブラリルーチン 331
endgraent ライブラリルーチン 331
endgrent ライブラリルーチン 331
endhostent ライブラリルーチン 331
endmntent ライブラリルーチン 331
endnetent ライブラリルーチン 331
endnetgrent ライブラリルーチン 331

endprotoent ライブラリルーチン 331
 endpwaent ライブラリルーチン 331
 endpwent ライブラリルーチン 331
 endrpcent ライブラリルーチン 331
 endservent ライブラリルーチン 331
 endttyent ライブラリルーチン 331
 endusershell ライブラリルーチン 331
 endwin ライブラリルーチン 331
 end ライブラリルーチン 330
 enroll コマンド 244
 environ システムファイル 407
 env コマンド 244, 263
 eqn コマンド 244
 erand48 ライブラリルーチン 331
 erasechar ライブラリルーチン 332
 erase ライブラリルーチン 332
 erfc ライブラリルーチン 332
 erf ライブラリルーチン 332
 errno 値 284
 errno ライブラリルーチン 332
 error コマンド 198, 244
 /etc/acct ディレクトリ 416
 /etc/aliases ファイル 406, 425
 /etc/config コマンド 94, 210, 212, 238
 /etc/cron.d ディレクトリ 416
 /etc/default/fs ファイル 109
 /etc/default/login ファイル 67
 /etc/default/passwd ファイル 67
 /etc/default/su ファイル 67
 /etc/default ディレクトリ 66, 111, 416
 /etc/dfs/dfstab ファイル 59, 122
 /etc/dfs/fstype ファイル 109
 /etc/dfs/sharetab ファイル 317, 425
 /etc/dfs ディレクトリ 416
 /etc/exports ファイル 59, 122, 317, 407, 425
 /etc/fstab ファイル
 /etc/vfstab vs. 58, 59, 112, 425
 記述 407, 425
 描写 52
 /etc/fs ディレクトリ 109, 416
 /etc/group ファイル 59, 408
 /etc/inet ディレクトリ 111, 416
 /etc/init.d スクリプト 265, 416
 /etc/inittab ファイル 93, 95
 /etc/lib ディレクトリ 416
 /etc/.login ファイル 73
 /etc/lp/printers ディレクトリ 143
 /etc/lp ディレクトリ 46, 100, 111, 142, 416
 /etc/mail/aliases ファイル 425
 /etc/mail/sendmail.cf ファイル 60
 /etc/mail ディレクトリ 416
 /etc/mnttab ファイル 425
 /etc/mntab ファイル 409, 425
 /etc/netgroup ファイル 59, 409
 /etc/net ディレクトリ 416
 /etc/opt ディレクトリ 111, 118, 416
 /etc/passwd ファイル 60, 66, 409, 425
 /etc/printcap データベース、置換 100, 142, 143, 410
 /etc/printcap データベース、復元 60
 /etc/printcap データベース、変更 46, 425
 /etc/profile ファイル 73
 /etc/rc.boot スクリプト 94, 95, 111, 212, 265
 /etc/rc.d スクリプト 111, 212, 416, 426
 /etc/rc.local スクリプト 94, 95, 111, 212, 265, 426
 /etc/rc.single スクリプト 94, 111, 212
 /etc/rcS.d スクリプト 111, 416, 426
 /etc/rcS スクリプト 94, 111, 212
 /etc/rc スクリプト 94, 95, 111, 212, 265, 416, 426
 /etc/rmmount.conf ファイル 90
 /etc/saf ディレクトリ 112, 417
 /etc/sendmail.cf ファイル 60
 /etc/shadow ファイル 60, 66, 115, 425
 /etc/skel ディレクトリ 74, 417
 /etc/sm.bak ディレクトリ 417
 /etc/sm ディレクトリ 417
 /etc/system ファイル
 moddir 変数 41, 210
 記述 94, 210, 212
 描写 88
 /etc/tm ディレクトリ 417
 /etc/ttytab ファイル 60, 412
 /etc/uucp/Config ファイル 152, 154
 /etc/uucp/Grades ファイル 152
 /etc/uucp/Limits ファイル 152, 154
 /etc/uucp ディレクトリ 59, 152, 417
 /etc/vfstab ファイル
 fields 112, 115
 merging /etc/fstab ファイル内 58, 59
 記述 425
 説明 112

のファイルシステムを指定する 121
例 114
/etc/vold.conf ファイル 90
/etc/xtab ファイル 413, 425
/etc ディレクトリ
記述 416
説明 102, 110
の変更 100, 109, 110, 112, 416, 417
etext ライブラリルーチン 332
ether_aton ライブラリルーチン 332
etherd コマンド 244
etherfind コマンド 244
ether_hostton ライブラリルーチン 332
ether_line ライブラリルーチン 332
Ethernet ドライバ 204
ether_ntoa ライブラリルーチン 332
ether_ntohost ライブラリルーチン 332
ethers システムファイル 407
ether ライブラリルーチン 332
EUC (拡張 UNIX コード) 205
exc_bound ライブラリルーチン 332
exc_handle ライブラリルーチン 332
exc_notify ライブラリルーチン 332
exc_on_exit ライブラリルーチン 332
exc_raise ライブラリルーチン 332
exc_unhandle ライブラリルーチン 332
exc_uniqpatt ライブラリルーチン 332
execle ライブラリルーチン 333
execlp ライブラリルーチン 333
execl ライブラリルーチン 332
execve システムコール 288
execvp ライブラリルーチン 333
execv ライブラリルーチン 333
/exec ディレクトリ 117
exit ライブラリルーチン 333
exp10 ライブラリルーチン 334
exp2 ライブラリルーチン 334
expand コマンド 244
expm1 ライブラリルーチン 334
exportent ライブラリルーチン 334
exportfs コマンド 120, 122, 244, 424
/export/home ディレクトリ 233
exports ファイル 59, 122, 317, 407, 425
/export ディレクトリ 132, 417
expr コマンド 244
exp ライブラリルーチン 334
exstr コマンド 200

Extended Fundamental Types (EFT) 51
extended_to_decimal ライブラリルーチン 334
extract_files コマンド 244
extract_patch コマンド 245
extract_unbundled コマンド 245
ex エディタ用のバックアップファイルディレクトリ 418
ex コマンド 243, 244
e コマンド 243

F

fabs ライブラリルーチン 334
FACE パッケージ 421, 422
false コマンド 245
fastboot コマンド 92, 98, 245
fasthalt コマンド 94, 96, 98, 245
fctab システムファイル 407
fchdir システムコール 288
fchmod システムコール 288
fchown システムコール 288
fchroot システムコール 289
fclose ライブラリルーチン 331, 334
fcntl システムコール 289, 301
fcntl システムファイル 407
fconvert ライブラリルーチン 334
fcvt ライブラリルーチン 335
fdformat コマンド 245
FDFS (ファイル記述子ファイルシステム) 100, 105
fdopen ライブラリルーチン 335
feof ライブラリルーチン 335
ferror ライブラリルーチン 335
fetch ライブラリルーチン 335
fflush ライブラリルーチン 335
ffs ライブラリルーチン 335
ff コマンド 107
fgetc ライブラリルーチン 335
fgetgraent ライブラリルーチン 335
fgetgrent ライブラリルーチン 335
fgetpwaent ライブラリルーチン 335
fgetpwent ライブラリルーチン 335
fgets ライブラリルーチン 335
fgrep コマンド 245
FIFOFS (FIFO/パイプファイルシステム) 100, 105

fileno ライブラリルーチン 335
file_to_decimal ライブラリルーチン 335
file コマンド 245
filio ioctl 185, 187
filter ライブラリルーチン 335
find コマンド 122, 245
fingerd コマンド 245
finger コマンド 245
finite ライブラリルーチン 335
firstkey ライブラリルーチン 335
fixterm ライブラリルーチン 335
flash ライブラリルーチン 336
floatingpoint ライブラリルーチン 336
flock システムコール 289
floor ライブラリルーチン 336
/floppy ディレクトリ 89, 121
flushingp ライブラリルーチン 336
flusok ライブラリルーチン 336
FMLI スクリプトが使用するバイナリ 421
fmod ライブラリルーチン 336
fmt_mail コマンド 246
fmt コマンド 245
fold コマンド 246
fontedit コマンド 246
fonts
 troff 用のフォント記述ファイル 420
/font ディレクトリ 420
fopen ライブラリルーチン 336, 373, 374
foption コマンド 246
fork システムコール 289
format コマンド 51, 246
fpa_download コマンド 246
fparel コマンド 246
fpathconf システムコール 289
fpaversion コマンド 246
fp_class ライブラリルーチン 336
fprintf ライブラリルーチン 336
fpurel コマンド 246
fputc ライブラリルーチン 336
fputs ライブラリルーチン 336
fpuversion コマンド 246
fread ライブラリルーチン 336
free ライブラリルーチン 336
freopen ライブラリルーチン 336
frexp ライブラリルーチン 336
from コマンド 246
fscanf ライブラリルーチン 336

fsck_cdrom コマンド 246
fsck コマンド 107, 124, 240, 246
fsdb コマンド 107, 249
fseek ライブラリルーチン 336
fsflush コマンド 278
fsirand コマンド 246
fspec システムファイル 407
fstatfs システムコール 289
fstatvfs システムコール 289
fstat システムコール 289
fstyp コマンド 107, 243
fsync システムコール 289
fs システムファイル 407
/fs ディレクトリ 109, 110, 117, 416, 420
ftell ライブラリルーチン 336
ftime ライブラリルーチン 337
ftok ライブラリルーチン 337
ftpd コマンド 246
ftputers システムファイル 407
ftp コマンド 246
ftruncate システムコール 289
ftw ライブラリルーチン 337
fumount コマンド 246
func_to_decimal ライブラリルーチン 337
fusage コマンド 246
fuser コマンド 247
fwrite ライブラリルーチン 337
fwtmp コマンド 247

G

/games ディレクトリ 419
gamma ライブラリルーチン 337
garbagedlines ライブラリルーチン 337
gcd ライブラリルーチン 337
gconvert ライブラリルーチン 337
gcore コマンド 247
gcvt ライブラリルーチン 337
generic_args コマンド 247
getacdir ライブラリルーチン 337
getacflg ライブラリルーチン 337
getacinfo ライブラリルーチン 337
getacmin ライブラリルーチン 337
get_alarm コマンド 247
getauditflagsbin ライブラリルーチン 337
getauditflagschar ライブラリルーチン 337

getuid システムコール 290
getbegyx ライブラリルーチン 338
getcap ライブラリルーチン 338
getchar ライブラリルーチン 338
getch ライブラリルーチン 338
getcwd ライブラリルーチン 207, 338, 342
getc ライブラリルーチン 338
getdate ライブラリルーチン 381
getdents システムコール 290
getdirentries システムコール 290
getdomainname システムコール 290
getdtablesize システムコール 290
getegid システムコール 290
getenv ライブラリルーチン 338
geteuid システムコール 290
getexportent ライブラリルーチン 338
getexportopt ライブラリルーチン 339
getfauditflags ライブラリルーチン 339
getfsent ライブラリルーチン 339
getfsfile ライブラリルーチン 339
getfsspec ライブラリルーチン 339
getfstype ライブラリルーチン 339
getgid システムコール 291
getgraent ライブラリルーチン 339
getgranam ライブラリルーチン 339
getgrent ライブラリルーチン 339
getgrgid ライブラリルーチン 339
getgrnam ライブラリルーチン 339
getgroups システムコール 291
gethostbyaddr ライブラリルーチン 339
gethostbyname ライブラリルーチン 339
gethostent ライブラリルーチン 339
gethostid システムコール 291
gethostname システムコール 291
getitimer システムコール 291
getlogin ライブラリルーチン 339
getmaxyx ライブラリルーチン 339
getmntent ライブラリルーチン 340
getmsg システムコール 291
get_myaddress ライブラリルーチン 340
getnetbyaddr ライブラリルーチン 340
getnetbyname ライブラリルーチン 340
getnetent ライブラリルーチン 340
getnetgrent ライブラリルーチン 340
getnetname ライブラリルーチン 340

getoptcvd コマンド 247
getopts コマンド 247
getopt コマンド 247
getopt ライブラリルーチン 340
getpagesize システムコール 291
getpass ライブラリルーチン 340
getpeername システムコール 291
getpgid システムコール 292
getpgrp システムコール 292
getpid システムコール 292
getppid システムコール 292
getpriority システムコール 292
getprotobyname ライブラリルーチン 340
getprotobynumber ライブラリルーチン 340
getprotoent ライブラリルーチン 340
getpublickey ライブラリルーチン 340
getpwaent ライブラリルーチン 341
getpwanam ライブラリルーチン 341
getpwent ライブラリルーチン 341
getpwnam ライブラリルーチン 341
getpwuid ライブラリルーチン 341
getpw ライブラリルーチン 340
getrlimit システムコール 290, 293, 305, 392
getrpcbyname ライブラリルーチン 341
getrpcbynumber ライブラリルーチン 341
getrpcent ライブラリルーチン 341
getrpcport ライブラリルーチン 341
getrusage システムコール 293, 393
getsecretkey ライブラリルーチン 341
get_selection コマンド 247
getservbyname ライブラリルーチン 341
getservbyport ライブラリルーチン 341
getservent ライブラリルーチン 341
getsockname システムコール 293
getsockopt システムコール 293
getstr ライブラリルーチン 341
getsubopt ライブラリルーチン 341
getsyx ライブラリルーチン 341
gets ライブラリルーチン 341
gettable コマンド 247
gettext ライブラリルーチン 207, 341
gettimeofday システムコール 293
gettmode ライブラリルーチン 342
getttyent ライブラリルーチン 342

gettynam ライブラリルーチン 342
gettytab データベース 408
getty コマンド 247
getuid システムコール 293
getusershell ライブラリルーチン 342
getut ライブラリルーチン 207
getvfsany ライブラリルーチン 339
getvfsent ライブラリルーチン 339
getvfsfile ライブラリルーチン 339
getwd ライブラリルーチン 342
getw ライブラリルーチン 342
getyx ライブラリルーチン 342
get コマンド 198, 247
gfxtool コマンド 247
gigipolot コマンド 247
glob コマンド 248
gmtime ライブラリルーチン 342
goto コマンド 248
gpconfig コマンド 248
gprof コマンド 201, 248
graphical user interface (GUI)
 インストール 29
graph コマンド 248
grep コマンド 248
group.adjunct システムファイル 408
groups コマンド 248
group システムファイル 59, 408
grpauth ライブラリルーチン 342
grpck コマンド 248
gsignal ライブラリルーチン 342
GSS-API 66
gtty ライブラリルーチン 342
gxttest コマンド 248

H

halfdelay ライブラリルーチン 343
halt コマンド 92, 96, 98, 248
hashcheck コマンド 248
hashmake コマンド 248
hashstat コマンド 248
has_ic ライブラリルーチン 343
has_il ライブラリルーチン 343
hasmntopt ライブラリルーチン 343
hcreate ライブラリルーチン 343
hdestroy ライブラリルーチン 343

head コマンド 248
help_open コマンド 249
help コマンド 198, 248
holidays system ファイル 408
\$HOME/.cshrc ファイル 73
\$HOME/.login ファイル 73
\$HOME/.profile ファイル 73
/home ディレクトリ 417
\$HOME/ ファイル 73
/home ファイルシステム 103
host2netname ライブラリルーチン 343
hostid コマンド 249
hostname コマンド 249
hostrfs コマンド 249
hosts.equiv システムファイル 408
hosts データベース 408
hp7221plot コマンド 249
hpplot コマンド 249
hsearch ライブラリルーチン 343
HSFS (CD-ROM ファイルシステム) 88, 104,
 121
htable コマンド 249
HUGE_VAL ライブラリルーチン 343
HUGE ライブラリルーチン 343
hypot ライブラリルーチン 343

I

i386 コマンド 249
iAPX286 コマンド 249
icheck コマンド 249
Icon Edit tool (OpenWindows) 249
iconedit コマンド 249
iconv コマンド用変換テーブル 420
/iconv ディレクトリ 420
iconv 用変換テーブル 420
idload コマンド 249
idlok ライブラリルーチン 344
id コマンド 249, 280
ID データ型、拡張された 30
ieee_flags ライブラリルーチン 344
ieee_functions ライブラリルーチン 344
ieee_handler ライブラリルーチン 344
ieee_retrospective ライブラリルーチン 344
ifconfig コマンド 249
IIIMP (internet intranet input method
 protocol) 151

- ilogb ライブラリルーチン 344
- imemtest コマンド 249
- implot コマンド 249
- in.comsat コマンド 250
- in.fingerd コマンド 250
- in.ftpd コマンド 250
- in.named コマンド 250, 258
- in.rexcd コマンド 266
- in.rexd コマンド 266
- in.rexecd コマンド 250
- in.rlogind コマンド 250
- in.routed コマンド 250
- in.rshd コマンド 250
- in.rwhod コマンド 250
- in.talkd コマンド 250
- in.telnetd コマンド 250
- in.tftpd コマンド 250
- in.tnamed コマンド 250
- in.uucpd コマンド 250
- inch ライブラリルーチン 344
- /include ディレクトリ 116, 419
- indent.pro システムファイル 408
- indentpro_to_defaults コマンド 250
- indent コマンド 201, 250
- index ライブラリルーチン 344
- indxbib コマンド 250
- inetboot コマンド 94, 211, 212
- inetd.conf データベース 408
- inetd コマンド 250
- inet_lnaof ライブラリルーチン 344
- inet_makeaddr ライブラリルーチン 344
- inet_netof ライブラリルーチン 344
- inet_network ライブラリルーチン 344
- inet_ntoa ライブラリルーチン 344
- /inet ディレクトリ 111, 416
- infinity ライブラリルーチン 344
- infocmp コマンド 250
- init.d スクリプト 265, 416
- initgroups ライブラリルーチン 344
- initscr ライブラリルーチン 345
- initstate ライブラリルーチン 345
- inittab ファイル 93, 95
- init コマンド
 - 使用 94
 - 使用する 96
 - 説明 93
 - 代替コマンド 245
- によって置き換えられたコマンド 92
の変更 94, 251
- inline コマンド 201, 251
- innetgr ライブラリルーチン 345
- inodes、クリアする 107
- input_from_defaults コマンド 251
- insch ライブラリルーチン 345
- insert_brackets コマンド 251
- insertln ライブラリルーチン 345
- insque ライブラリルーチン 345
- install4x プログラム 132, 133
- installboot コマンド 93, 211, 251
- installtxt コマンド 206, 251
- install コマンド 251
- internat システムファイル 408
- internet intranet input method protocol
(IIIMP) 151
- intrflush ライブラリルーチン 345
- intr コマンド 251
- ioctl 要求 185, 188, 294
STREAMS 223, 224
- iostat コマンド 251
- ipallocd コマンド 251
- ipalloc ライブラリルーチン 345
- ipcrm コマンド 251
- ipcs コマンド 252
- irint ライブラリルーチン 345
- isainfo コマンド 252
- isalnum ライブラリルーチン 345
- isalpha ライブラリルーチン 345
- isascii ライブラリルーチン 345
- isatty ライブラリルーチン 345
- iscntrl ライブラリルーチン 345
- isdigit ライブラリルーチン 345
- isendwin ライブラリルーチン 346
- isgraph ライブラリルーチン 346
- isinf ライブラリルーチン 346
- islower ライブラリルーチン 346
- isnan ライブラリルーチン 346
- isnormal ライブラリルーチン 346
- isprint ライブラリルーチン 346
- ispunct ライブラリルーチン 346
- issecure ライブラリルーチン 346
- isspace ライブラリルーチン 346
- issubnormal ライブラリルーチン 346
- isupper ライブラリルーチン 346

isxdigit ライブラリルーチン 346
iszero ライブラリルーチン 346
itom ライブラリルーチン 346

J

j0 ライブラリルーチン 347
j1 ライブラリルーチン 347
jn ライブラリルーチン 347
join コマンド 252
jrand48 ライブラリルーチン 347

K

kadb コマンド 180 - 182, 252
Kerberos セキュリティ 69
/kernel 39, 101
 crash dump ディレクトリ 417
 モジュール、ディレクトリ検索パス 417,
 419
/kernel/drv ディレクトリ 117
/kernel/exec ディレクトリ 117
/kernel/fs ディレクトリ 117
/kernel/misc ディレクトリ 117
/kernel/sched ディレクトリ 117
/kernel/strmod ディレクトリ 117
/kernel/sys ディレクトリ 118
/kernel/unix ディレクトリ 50, 94, 118, 210
/kernel ディレクトリ 41, 101, 117, 210, 417,
 419
key_decryptsession ライブラリルーチン 347
key_encryptsession ライブラリルーチン 347
keyenvoy コマンド 252
key_gendes ライブラリルーチン 347
keylogin コマンド 252
keylogout コマンド 252
keyname ライブラリルーチン 347
keypad ライブラリルーチン 347
keyserv コマンド 252
key_setsecret ライブラリルーチン 347
keytables システムファイル 408
/keytables ディレクトリ 421
kgmon コマンド 252
killchar ライブラリルーチン 347
killpg システムコール 294
kill コマンド 252
kill システムコール 294

klm_prot ライブラリルーチン 347
Korn シェル 71 - 73
 制限付きシェル 67
ksh コマンド 71 - 73
ksyms ファイル 182
kvm_close ライブラリルーチン 348
kvm_getcmd ライブラリルーチン 348
kvm_getproc ライブラリルーチン 348
kvm_getu ライブラリルーチン 348
kvm_nextproc ライブラリルーチン 348
kvm_nlist ライブラリルーチン 348
kvm_open ライブラリルーチン 348
kvm_read ライブラリルーチン 348
kvm_setproc ライブラリルーチン 348
kvm_write ライブラリルーチン 348
/kvm ディレクトリ 419

L

l3tol ライブラリルーチン 207, 348
l64a ライブラリルーチン 348
labelit コマンド 107, 125, 253
label ライブラリルーチン 348
langinfo ライブラリルーチン 207
lastcomm コマンド 253
lastlogin コマンド 253
last コマンド 253
lcong48 ライブラリルーチン 348
ldaclose ライブラリルーチン 348
ldahread ライブラリルーチン 348
ldaopen ライブラリルーチン 349
LDAP (lightweight directory access
 protocol) 151
ldclose ライブラリルーチン 349
ldconfig コマンド 253
ldd コマンド 253
ldexp ライブラリルーチン 349
ldfcn ライブラリルーチン 349
ldfhread ライブラリルーチン 349
ldgetname ライブラリルーチン 349
ldlinit ライブラリルーチン 349
ldlitem ライブラリルーチン 349
ldlread ライブラリルーチン 349
ldlseek ライブラリルーチン 349
ldnlseek ライブラリルーチン 349
ldnrseek ライブラリルーチン 349

ldnshread ライブラリルーチン 349
ldnsseek ライブラリルーチン 349
ldohseek ライブラリルーチン 349
ldopen ライブラリルーチン 349
ldrseek ライブラリルーチン 350
ldshread ライブラリルーチン 350
ldsseek ライブラリルーチン 350
ldtbindex ライブラリルーチン 350
ldtbread ライブラリルーチン 350
ldtbseek ライブラリルーチン 350
ld コマンド 198, 253
leaveok ライブラリルーチン 350
leave コマンド 253
lex コマンド 199, 253
lfind ライブラリルーチン 350
lgamma ライブラリルーチン 350
libc ディレクトリ 206
libdevinfo 30
libintl ディレクトリ 206
/libp ディレクトリ 192, 420
libsocket ディレクトリ 204
libw ディレクトリ 205
/libxpg2.a ライブラリ 206
/libxpg ディレクトリ 206
lightweight directory access protocol
(LDAP) 151
linemod ライブラリルーチン 350
line コマンド 253
line ライブラリルーチン 350
link コマンド 253
link システムコール 295
link システムファイル 409
lint コマンド 171, 201, 253
lint ライブラリ 171
listen システムコール 295
listen ポートモニタ 47, 146, 253
ln コマンド 254
loadkeys コマンド 254
/local.cshrc ファイル 73
/local.login ファイル 74
/local.profile ファイル 74
localdtconv ライブラリルーチン 350
localeconv ライブラリルーチン 350
locale データベース 409, 420
localization データベース 420
localtime ライブラリルーチン 350, 388
/local ディレクトリ 421

lockd コマンド 254
lockf ライブラリルーチン 350
locks
 network lock デーモン 425
 ロックファイルのスーパーリング 418
lockscreen コマンド 254
/locks ディレクトリ 418
LOFS (ループバックファイルシステム) 105
log10 ライブラリルーチン 351
log1p ライブラリルーチン 351
log2 ライブラリルーチン 351
logb ライブラリルーチン 351
logger コマンド 254
login コマンド 254
login ファイル 67, 73
logname コマンド 254
logname ライブラリルーチン 207
log ファイル
 cron 418
 Service Access Facility (SAF) 418
 uucp 419
 システム 417
log ライブラリルーチン 350
_longjmp ライブラリルーチン 351
longname ライブラリルーチン 351
looklib コマンド 254
look コマンド 254
lorder コマンド 199, 254
lpadmin コマンド 143, 255
lpc コマンド 142, 143, 254
lpd コマンド 142, 255, 425
lpmove コマンド 143
lpq コマンド 46, 142, 255, 423
lprm コマンド 46, 142, 143, 255, 423
lprprof コマンド 275
lpr コマンド 46, 142, 255, 423
lpsched コマンド 425
lpstat コマンド 46, 142, 255, 423
lpssystem コマンド 143
lptest コマンド 255
lp コマンド 46, 142, 144, 254, 423
lrand48 ライブラリルーチン 351
lsearch ライブラリルーチン 351
lseek システムコール 295
lstat システムコール 295

lsw コマンド 255
 ls コマンド 121, 255
 ltol3 ライブラリルーチン 352
 lwp_checkstkset ライブラリルーチン 352
 lwp_create ライブラリルーチン 352
 lwp_ctxinit ライブラリルーチン 352
 lwp_ctxmemget ライブラリルーチン 352
 lwp_ctxmemset ライブラリルーチン 352
 lwp_ctxremove ライブラリルーチン 352
 lwp_ctxset ライブラリルーチン 352
 lwp_datastk ライブラリルーチン 352
 lwp_destroy ライブラリルーチン 352
 lwp_enumerate ライブラリルーチン 352
 lwp_errstr ライブラリルーチン 352
 lwp_fpset ライブラリルーチン 352
 lwp_geterr ライブラリルーチン 352
 lwp_getregs ライブラリルーチン 352
 lwp_getstate ライブラリルーチン 352
 lwp_join ライブラリルーチン 352
 lwp_libcset ライブラリルーチン 352
 lwp_newstk ライブラリルーチン 352
 lwp_perror ライブラリルーチン 352
 lwp_ping ライブラリルーチン 353
 lwp_resched ライブラリルーチン 353
 lwp_resume ライブラリルーチン 353
 lwp_self ライブラリルーチン 353
 lwp_setpri ライブラリルーチン 353
 lwp_setregs ライブラリルーチン 353
 lwp_setstkcache ライブラリルーチン 353
 lwp_sleep ライブラリルーチン 353
 lwp_stkcswset ライブラリルーチン 353
 lwp_suspend ライブラリルーチン 353
 lwp_yield ライブラリルーチン 353

M

m4 コマンド 199, 255
 m68k コマンド 255
 mach コマンド 255
 madd ライブラリルーチン 353
 madvise ライブラリルーチン 353
 magic システムファイル 409
 mail
 構成情報ディレクトリ 416
 ディレクトリ 418, 425
 補助プログラムのディレクトリ 420

 待ち行列のディレクトリ 418
 mailrc_to_defaults コマンド 256
 mailstat コマンド 256
 Mail Tool (OpenWindows) 256
 mailtool インタフェース 77, 256
 mailx コマンド 77, 255, 421
 /mailx ディレクトリ 421
 mailx のヘルプファイルコマンド 421
 mailx プログラム 43
 mail コマンド 76, 77, 234, 255
 makedbm コマンド 256
 MAKEDEV 環境変数 94
 makedev コマンド 256
 Makefiles 193, 194
 makekey コマンド 256
 make コマンド 193, 194, 199, 256
 malloc_debug ライブラリルーチン 353
 mallocmap ライブラリルーチン 353
 malloc_verify ライブラリルーチン 353
 malloc ライブラリルーチン 207, 353
 man.cf ファイル 79
 man68010 コマンド 257
 man68020 コマンド 257
 man68881version コマンド 257
 MANPATH 環境変数 80
 MANSECTS 環境変数 80
 man コマンド 79 - 81, 257
 /man ディレクトリ 78, 80
 man ページ
 windex データベース 80, 231
 ディレクトリ構成の変更 79
 matherr ライブラリルーチン 354
 max_normal ライブラリルーチン 354
 max_subnormal ライブラリルーチン 354
 mblen ライブラリルーチン 354
 mbstowcs ライブラリルーチン 354
 mbtowc ライブラリルーチン 354
 mcmp ライブラリルーチン 354
 mconnect コマンド 257
 mcs コマンド 200
 mctl システムコール 295
 mdiv ライブラリルーチン 354
 memalign ライブラリルーチン 354
 memccpy ライブラリルーチン 354
 memchr ライブラリルーチン 354
 memcmp ライブラリルーチン 320, 354

memcntl システムコール 295
 memcpy ライブラリルーチン 320, 354
 memset ライブラリルーチン 321, 354
 merging ファイル 57, 58
 mesg コマンド 257
 metadb コマンド 53
 metastat コマンド 53
 meta ライブラリルーチン 354
 mfree ライブラリルーチン 354
 mincore システムコール 295
 min_normal ライブラリルーチン 354
 min_subnormal ライブラリルーチン 354
 min ライブラリルーチン 354
 /misc ディレクトリ 117
 mkdir コマンド 257
 mkdir システムコール 296
 mkfifo システムコール 296
 mkfile コマンド 257
 mkfs コマンド 107, 123, 124, 257
 mknod コマンド 257
 mknod システムコール 297
 mkproto コマンド 257
 mkstemp ライブラリルーチン 354
 mkstr コマンド 257
 mktemp ライブラリルーチン 355
 mktime ライブラリルーチン 388
 mlockall ライブラリルーチン 355
 mlock ライブラリルーチン 355
 mmap システムコール 297
 mnttab ファイル 425
 /mnt ディレクトリ 417
 moddebug マクロ 182
 moddir 変数 41, 210
 modf ライブラリルーチン 355
 modinfo コマンド 209, 258
 modload コマンド 40, 210, 212, 225, 257
 modstat コマンド 258
 modunload コマンド 40, 210, 225, 258
 monacct コマンド 258
 mon_break ライブラリルーチン 355
 mon_cond_enter ライブラリルーチン 355
 moncontrol ライブラリルーチン 355
 mon_create ライブラリルーチン 355
 mon_destroy ライブラリルーチン 355
 mon_enter ライブラリルーチン 355
 mon_enumerate ライブラリルーチン 355
 mon_exit ライブラリルーチン 355
 MONITOR ライブラリルーチン 355
 monitor ライブラリルーチン 356
 monstartup ライブラリルーチン 356
 mon_waiters ライブラリルーチン 355
 more コマンド 258
 Motif admin ツール 24
 mountall コマンド 107, 424
 mountd コマンド 258
 mount_tfs コマンド 258
 mount コマンド 107, 120, 258, 424
 mount システムコール 298
 mout ライブラリルーチン 356
 move ライブラリルーチン 356
 mprotect システムコール 298
 /mqueue ディレクトリ 418
 mrand48 ライブラリルーチン 356
 msgctl システムコール 299
 msg_enumrecv ライブラリルーチン 357
 msg_enumsend ライブラリルーチン 357
 msgfmt コマンド 206, 251
 msgget システムコール 299
 msgrcv システムコール 299
 MSG_RECVALL ライブラリルーチン 357
 msg_recv ライブラリルーチン 357
 msg_reply ライブラリルーチン 357
 msg_send ライブラリルーチン 357
 msgsnd システムコール 299
 msub ライブラリルーチン 357
 msync システムコール 299
 msync ライブラリルーチン 357
 mtab システムファイル 409, 425
 mtio ioctl 185, 187
 mtox ライブラリルーチン 357
 mt コマンド 258
 MT スタイルのロック 222
 MT (マルチスレッド) カーネル 30
 mult ライブラリルーチン 357
 munlockall ライブラリルーチン 357
 munlock ライブラリルーチン 357
 munmap システムコール 299
 mutexes 181, 222
 mutex マクロ 181
 mvaddch ライブラリルーチン 357
 mvaddstr ライブラリルーチン 357
 mvcur ライブラリルーチン 357
 mvdelch ライブラリルーチン 358

mvgetch ライブラリルーチン 358
mvgetstr ライブラリルーチン 358
mvinch ライブラリルーチン 358
mvinsch ライブラリルーチン 358
mvprintw ライブラリルーチン 358
mvscanw ライブラリルーチン 358
mvwaddch ライブラリルーチン 358
mvwaddstr ライブラリルーチン 358
mvwdelch ライブラリルーチン 358
mvwgetch ライブラリルーチン 359
mvwgetstr ライブラリルーチン 359
mvwinch ライブラリルーチン 359
mvwinsch ライブラリルーチン 359
mvwin ライブラリルーチン 359
mvwprintw ライブラリルーチン 359
mvwscanw ライブラリルーチン 359
mv コマンド 258

N

named コマンド 258
NAMEFS (ネームファイルシステム) 101, 105
Name Service Switch 156, 204
napms ライブラリルーチン 360
nawk コマンド 258
ncheck コマンド 107, 240, 258
ndbootd コマンド 258
neqn コマンド 258
net_addr ライブラリルーチン 360
netdir_getbyname ライブラリルーチン 340
netgroup システムファイル 59, 409
netmasks データベース 409
netname2host ライブラリルーチン 360
netname2user ライブラリルーチン 360
netrc システムファイル 409
netstat コマンド 259
/netsvc ディレクトリ 268, 420
Network Interface Tap (NIT) 204
network lock デーモン 425
networks データベース 409
newaliases コマンド 259
newfs コマンド 123, 259
newgrp コマンド 259
newkey コマンド 259
newpad ライブラリルーチン 360
/news ディレクトリ 418
newterm ライブラリルーチン 360

newwin ライブラリルーチン 360
nextafter ライブラリルーチン 360
nextkey ライブラリルーチン 360
nexus ドライバ 224
nexus ノード 224
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 424

コマンドの変更 150

セキュリティ 65

デーモン 420, 425

に対するサポート 104

バインダプロセス 425

ファイルシステムの自動マウント 41

補助プログラムとデーモンのディレクト
リ 420

マウントされた NFS ファイルシステムの
リスト 424

マウント要求 サーバ 425

nfsd コマンド 259, 425

nfsstat コマンド 259

nfssvc システムコール 299

nfssys システムコール 299

/nfs ディレクトリ 420, 425

nice コマンド 259

nice ライブラリルーチン 360

nint ライブラリルーチン 360

NIS+ (Network Information Services Plus)

database ディレクトリ 418

DNS vs. 156, 157

NIS vs. 156, 157

TCP/IP と 149

アップグレードの計画 159

説明 156, 203

へのNIS の移行 158, 159

NIS+ (ネットワーク情報サービスプラス)

検索 45

更新 45

描写 25, 29, 45

NIS (Network Information Services)

NIS+ vs. 156, 157

NIS+ の移行 158, 159

nispasswd コマンド 68

/nis ディレクトリ 418

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 57

マスタサーバ、システムデータの変換 54

マスタサーバ、システムデータを保存 57
 NIT (Network Interface Tap) 204
 nl_init ライブラリルーチン 360
 nlist ライブラリルーチン 360
 nl_langinfo ライブラリルーチン 360
 nlm_prot ライブラリルーチン 360
 nlsadmin コマンド 47, 259
 nl コマンド 259
 nl ライブラリルーチン 360
 nm コマンド 199, 260
 nocbreak ライブラリルーチン 361
 nocrmode ライブラリルーチン 361
 nodelay ライブラリルーチン 361
 noecho ライブラリルーチン 361
 nohup コマンド 260
 nonl ライブラリルーチン 361
 nonstandard_arithmetic ライブラリルーチン 361
 noraw ライブラリルーチン 361
 notimeout ライブラリルーチン 361
 nrand84 ライブラリルーチン 361
 nroff コマンド 260, 420 - 422
 nroff 用端末テーブル 421
 nslookup コマンド 260
 nsquery コマンド 260
 nsswitch.conf ファイル 204
 /nterm ディレクトリ 421
 ntohl ライブラリルーチン 361
 ntohs ライブラリルーチン 361
 nulladm コマンド 260

O

/oasys ディレクトリ 421
 objdump コマンド 201
 od コマンド 260
 old-analyze コマンド 260
 old-cat コマンド 260
 old-clocktool コマンド 260
 old-compact コマンド 260
 old-eyacc コマンド 260
 old-filemerge コマンド 260
 old-make コマンド 260
 old-perfmon コマンド 261
 old-prmail コマンド 261
 old-pti コマンド 261
 old-setkeys コマンド 261

old-sun3cvt コマンド 261
 old-syslog コマンド 261
 old-uncompact コマンド 261
 old-vc コマンド 261
 /old ディレクトリ 116, 200, 421
 OLIT (OPEN LOOK Intrinsics Toolkit) 197
 on_exit ライブラリルーチン 362
 on コマンド 261
 opendir ライブラリルーチン 362
 openlog ライブラリルーチン 362
 OPEN LOOK Intrinsics Toolkit (OLIT) 197
 openpl ライブラリルーチン 362
 OpenWindows
 clock コマンド 237
 Developer's Guide File Chooser 対 XView
 File Chooser 75
 Icon Edit tool 249
 indent コマンド 231
 Mail Tool 256
 mail インタフェース 77
 SunView replaced by 273
 Text Edit tool 275
 version differences 74
 xlock コマンド 254
 xset コマンド 270
 コマンドツール 237
 シェルツール 271
 パフォーマンスメータツール 262
 ファイルマネージャの変更 48, 88
 複数のディスプレイ 231
 プロパティウィンド 237, 240, 256, 270
 プロパティシート 250
 マウントまたはインストールポイン
 ト 421
 /openwin ディレクトリ 421
 open システムコール 300
 "Operation not applicable for FSType" メッ
 セージ 107
 /opt 101
 optarg ライブラリルーチン 362
 optind ライブラリルーチン 362
 /options ディレクトリ 418
 /opt/sunwspro ディレクトリ 417
 /opt ディレクトリ 111, 118, 417
 /opt ファイルシステム 101, 103
 OSF/Motif 162

CDE に準拠した 162
overlay ライブラリルーチン 362
overview コマンド 261
overwrite ライブラリルーチン 363

P

pack コマンド 261, 278
pac コマンド 261
pagesize コマンド 261
page コマンド 261
passwd2des ライブラリルーチン 363
passwd.adjunct システムファイル 409
passwd コマンド 68, 262
passwd ファイル 60, 66, 67, 409, 425
paste コマンド 262
pathconf システムコール 300
pause ライブラリルーチン 363
paxcpio コマンド 262
pax コマンド 262
pcat コマンド 262
PCFS (PC ファイルシステム) 104
pclose ライブラリルーチン 363
pdpll コマンド 262
pechochar ライブラリルーチン 363
perfmeter コマンド 262
perror ライブラリルーチン 363
pgrep コマンド 262
pg コマンド 262
phones データベース 409
/pics ディレクトリ 420
PIC アーカイブ 420
ping コマンド 262
pipe システムコール 300
pkgadd コマンド 34, 35, 197, 244
pkgask コマンド 197
pkgchk コマンド 35, 197
pkginfo コマンド 35, 197
pkginfo ファイル 195
pkgmk コマンド 196
pkgparam コマンド 197
pkgproto コマンド 196
pkgrm コマンド 34, 35, 197
pkgtrans コマンド 196
/pkg ディレクトリ 418
pkill コマンド 262

plock ライブラリルーチン 363
plottoa コマンド 262
plot コマンド 262
plot システムファイル 410
plot ライブラリルーチン 363
Pluggable Authentication Module (PAM) 70
pmadm コマンド 47, 48, 145, 146
pmap_getmaps ライブラリルーチン 363
pmap_getport ライブラリルーチン 341, 364
pmap_rmtcall ライブラリルーチン 364
pmap_set ライブラリルーチン 364
pmap_unset ライブラリルーチン 364
pnoutrefresh ライブラリルーチン 364
pnp ライブラリルーチン 364
pod_getexit ライブラリルーチン 364
pod_getmaxpri ライブラリルーチン 364
pod_getmaxsize ライブラリルーチン 364
pod_setexit ライブラリルーチン 364
pod_setmaxpri ライブラリルーチン 364
point ライブラリルーチン 364
poll システムコール 300
popen ライブラリルーチン 364
porting デバイスドライバを移行する 221
portmap コマンド 262
PostScript フィルタ 77
pow ライブラリルーチン 365
PPP (point-to-point protocol) 151
praudit コマンド 263
prctmp コマンド 263
prdaily コマンド 263
prefresh ライブラリルーチン 365
/preserve ディレクトリ 418
printenv コマンド 263
printf ライブラリルーチン 365
printw ライブラリルーチン 366
priocntl コマンド 265, 420
priocntl システムコール 292, 305
PROCFS (プロセスアクセスファイルシステム) 100, 105
/proc ディレクトリ 100
/proc ファイルシステム 103
.profile ファイル 73
profil システムコール 301
profil ライブラリルーチン 355, 356
prof コマンド 199, 263

prof ライブラリルーチン 366
PROM, からブートする 93
protocols データベース 410
prototype ファイル 196
proto システムファイル 410
prs コマンド 199, 263
prtacct コマンド 263
prtconf コマンド 87, 220, 221, 246
prtvoc コマンド 85 - 87, 242
prt コマンド 199, 263
pr コマンド 263
pstat (C/A/T) フィルタ 46, 77
psignal ライブラリルーチン 366
psrinfo コマンド 264
pstat コマンド 264, 424
ps コマンド 263, 424
ptrace システムコール 301
ptrace 要求値 188, 189
/pts ディレクトリ 110, 415
ptx コマンド 264
publickey データベース 410
/pub ディレクトリ 421
putchar ライブラリルーチン 366
putc ライブラリルーチン 366
putenv ライブラリルーチン 366
putmntent ライブラリルーチン 317
putmsg システムコール 301
putpwent ライブラリルーチン 366
putp ライブラリルーチン 366
puts ライブラリルーチン 366
putw ライブラリルーチン 366
pwck コマンド 264
pwdauthd コマンド 264
pwdauth ライブラリルーチン 366
pwd コマンド 264

Q

qsort ライブラリルーチン 367
queuedefs システムファイル 410
QuickCheck 51
quiet_nan ライブラリルーチン 367
quotacheck コマンド 264
quotactl システムコール 301
quotaoff コマンド 264
quotaon コマンド 264

quota コマンド 264
quot コマンド 122, 264

R

random ライブラリルーチン 367
rand ライブラリルーチン 367
ranlib コマンド 199, 265
rarpd コマンド 265
rasfilter8tol コマンド 265
rasterfile システムファイル 410
raster image フィルタ 77
rastrepl コマンド 265
raw ディスクデバイス 415
raw ディスクデバイス用ディレクトリ 110, 215
raw テープデバイス 415
raw テープデバイス, ディレクトリ用の 110
raw ライブラリルーチン 367
rc.boot スクリプト 94, 95, 111, 212, 265
rc.d スクリプト 111, 416, 426
rc.local スクリプト 94, 95, 111, 212, 265, 426
rc.single スクリプト 94, 111, 212
rcmd ライブラリルーチン 367
rcp コマンド 265
rcS.d スクリプト 111, 416, 426
rcS スクリプト 94, 96, 100, 111, 212
rc スクリプト 94 - 96, 100, 111, 115, 212, 265, 416, 426
rdate コマンド 265
rdist コマンド 265
/rdsk ディレクトリ 110, 215, 415
rdump コマンド 265
readdir ライブラリルーチン 368
readlink システムコール 302
readv システムコール 302
read システムコール 302
realloc ライブラリルーチン 369
realpath ライブラリルーチン 369
reboot コマンド 92, 96, 98, 265
reboot システムコール 302
re_comp ライブラリルーチン 369
recvfrom システムコール 303
recvmsg システムコール 303
recv システムコール 302
red コマンド 265
re_exec ライブラリルーチン 369

refer コマンド 265
/refer ディレクトリ 420
refresh ライブラリルーチン 369
regexp ライブラリルーチン 369
registerrpc ライブラリルーチン 369
rehash コマンド 265
reject コマンド 143
remainder ライブラリルーチン 369
rem_drv コマンド 212, 225
remexportent ライブラリルーチン 369
remote システムファイル 410
remove_brackets コマンド 265
remque ライブラリルーチン 369
rename システムコール 303
renice コマンド 265
repquota コマンド 266
reset_prog_mode ライブラリルーチン 335,
370
reset_shell_mode ライブラリルーチン 370
resetterm ライブラリルーチン 369
resetty ライブラリルーチン 370
reset コマンド 266
res_init ライブラリルーチン 370
res_mkquery ライブラリルーチン 370
resolv.conf システムファイル 410
res_send ライブラリルーチン 370
restartterm ライブラリルーチン 370
restore コマンド 125, 266, 424
rev コマンド 266
rewinddir ライブラリルーチン 370
rewind ライブラリルーチン 370
rexd コマンド 266
rexeed コマンド 266
rexec ライブラリルーチン 370
rex ライブラリルーチン 370
rfadmin コマンド 266
rfmaster システムファイル 410
rfpasswd コマンド 266
rfstart コマンド 266
rfstop コマンド 266
/RFS ファイル システム 100
rfuadmin コマンド 266
rfudaemon コマンド 266
rgb システムファイル 410
.rhosts.equiv ファイル 65
rhosts システムファイル 410
.rhosts ファイル 65
rindex ライブラリルーチン 370
ring_alarm コマンド 266
rint ライブラリルーチン 370
ripoffline ライブラリルーチン 370
rksh コマンド 67
rlogind コマンド 266
rlogin コマンド 266
rmail コマンド 267
rm_client コマンド 267
rmdel コマンド 199, 267
rmdir コマンド 267
rmdir システムコール 303
rmmount.conf ファイル 90
rmmount コマンド 90
rmnstat コマンド 267
rm_services コマンド 267
rmtab システムファイル 410
rmt コマンド 267
/rmt ディレクトリ 110, 415
rm コマンド 266
rnusers ライブラリルーチン 370
roffbib コマンド 267
rootmenu システムファイル 411
root アクセス、デフォルト 67
root ファイルシステム (/) 53, 415, 419
routed コマンド 267
route コマンド 267
RPC, セキュリティ 66, 69
rpcb_getaddr ライブラリルーチン 364
rpcb_getmaps ライブラリルーチン 363
rpcbbind コマンド 262
rpc.bootparamd コマンド 267
rpcb_rmtcall ライブラリルーチン 364
rpc_broadcast ライブラリルーチン 323
rpcb_set ライブラリルーチン 364
rpcb_unset ライブラリルーチン 364
rpc_call ライブラリルーチン 321
rpc_createerr ライブラリルーチン 370
rpc.etherd コマンド 267
rpcgen コマンド 268
rpcinfo コマンド 268
rpc.lockd コマンド 267, 425
rpc.mountd コマンド 268, 425
rpc.rexd コマンド 268
rpc.rquotad コマンド 268
rpc.rstatd コマンド 268

rpc.rusersd コマンド 268
rpc.rwalld コマンド 268
rpc.sprayd コマンド 268
rpc.statd コマンド 268
rpc.user_agentd コマンド 268
rpc.yppasswdd コマンド 268
rpc.yppupdated コマンド 268
rpc データベース 411
rpow ライブラリルーチン 370
rquota ライブラリルーチン 370
rrestore コマンド 268
rresvport ライブラリルーチン 370
rsh コマンド 67, 268
rstat ライブラリルーチン 371
rtime ライブラリルーチン 371
runacct コマンド 268
ruptime コマンド 268
rup コマンド 268
rusage コマンド 269
ruserok ライブラリルーチン 371
rusers コマンド 269
rusers ライブラリルーチン 371
rwall コマンド 269
rwall ライブラリルーチン 371
rwho コマンド 269

S

s. ファイル 194
S5 (System V ファイルシステム) 106
sacadm コマンド 47, 145, 146
sact コマンド 199, 269
SAC (サービスアクセス管理) 47
/sad ディレクトリ 110, 415
SAMECV ライブラリルーチン 371
SAMEMON ライブラリルーチン 371
SAMETHREAD ライブラリルーチン 371
sar コマンド 230, 264
savecore コマンド 269
saveterm ライブラリルーチン 371
savetty ライブラリルーチン 371
sa コマンド 269
/sa ディレクトリ 420
/sbin/init コマンド 92 - 94, 96
/sbin/rcS スクリプト 96, 100, 111, 115
/sbin/rc スクリプト 95, 96, 100, 111, 115
/sbin ディレクトリ

記述 417
説明 103, 109, 115
の変更 100

sbrk システムコール 303
sbrk ライブラリルーチン 207
scalbn ライブラリルーチン 372
scalb ライブラリルーチン 371
scandir ライブラリルーチン 372
scanf ライブラリルーチン 372
scanw ライブラリルーチン 372
sccs-admin コマンド 269
sccs-cdc コマンド 269
sccs-comb コマンド 269
sccs-delta コマンド 269
sccsdiff コマンド 199, 270
sccsfile システムファイル 411
sccs-get コマンド 269
sccs-help コマンド 269
sccs-prs コマンド 269
sccs-prt コマンド 270
sccs-rmdel コマンド 270
sccs-sact コマンド 270
sccs-sccsdiff コマンド 270
sccs-unget コマンド 270
sccs-val コマンド 270
sccs コマンド 199, 269
SCCS (ソースコード管理システム) 194
/sccs ディレクトリ 194, 198, 200
/sched ディレクトリ 117
scr_dump ライブラリルーチン 372
screenblank コマンド 270
screndump コマンド 270
screenload コマンド 270
scr_init ライブラリルーチン 372
script コマンド 270
scrolldefaults コマンド 270
scrollok ライブラリルーチン 373
scroll ライブラリルーチン 372
scr_restore ライブラリルーチン 372
SCSA (Sun common SCSI architecture), DDI 専用インタフェース 219
SCSI ディスク
Sun common SCSI architecture (SCSA) 219
マルチスレッド (MT) カーネル 222
命名規則 84

sd1g コマンド 94
 sdiff コマンド 270
 seconvert ライブラリルーチン 373
 sed コマンド 270
 seed48 ライブラリルーチン 373
 seekdir ライブラリルーチン 373
 selection_svc コマンド 270
 select システムコール 303, 392
 semctl システムコール 303
 semget システムコール 303
 semop システムコール 303
 sendmail.cf ファイル 60
 sendmail コマンド 43, 267, 270
 sendmsg システムコール 304
 sendto システムコール 304
 send システムコール 304
 Serial Port Manager (Admintool) 145
 /servers ディレクトリ 421
 Service Access Facility (SAF) 145, 147
 Service Access Controller (SAC) と 145
 コマンド 146
 説明 145, 147
 ディレクトリ 417
 補助プログラムとデーモンのディレクト
 リ 420
 用ディレクトリ 112
 ログアカウントファイルディレクト
 リ 418
 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 117
 services システムファイル 411
 set4 コマンド 270
 setac ライブラリルーチン 373
 set_alarm コマンド 271
 setaudit システムコール 304
 setaudit システムコール 304
 setbuffer ライブラリルーチン 373
 setbuf ライブラリルーチン 373
 set_curterm ライブラリルーチン 373
 setdomainname システムコール 304
 setegid システムコール 305
 setegid ライブラリルーチン 373
 seteuid システムコール 305
 seteuid ライブラリルーチン 373
 setexportent ライブラリルーチン 373
 setfsent ライブラリルーチン 373
 setgid システムコール 305
 setgid ライブラリルーチン 373, 374
 setgraent ライブラリルーチン 373
 setgrent ライブラリルーチン 373
 setgroups システムコール 304
 sethostent ライブラリルーチン 373
 sethostname システムコール 304
 setitimer システムコール 304, 391, 392
 _setjmp ライブラリルーチン 374
 setkeys コマンド 271
 setkey ライブラリルーチン 374
 setlinebuf ライブラリルーチン 374
 setlocale コマンド 207
 setlocale ライブラリルーチン 374
 setlogmask ライブラリルーチン 374
 setmntent ライブラリルーチン 374
 setnetent ライブラリルーチン 374
 setnetgrent ライブラリルーチン 374
 setpgid システムコール 305
 setpgrp システムコール 305
 setpriority システムコール 305
 setprotoent ライブラリルーチン 374
 setpwaent ライブラリルーチン 374
 setpwent ライブラリルーチン 374
 setpwfile ライブラリルーチン 374
 setregid システムコール 305
 setreuid システムコール 305
 setrgid ライブラリルーチン 374
 setrlimit システムコール 305
 setrpcent ライブラリルーチン 374
 setruuid ライブラリルーチン 374
 setscreg ライブラリルーチン 375
 setservent ライブラリルーチン 375
 setsid コマンド 271
 setsid システムコール 305
 setsockopt システムコール 305
 setstate ライブラリルーチン 375
 setsyx ライブラリルーチン 375
 set_term ライブラリルーチン 375
 settimeofday システムコール 306
 setttyent ライブラリルーチン 375
 setuid システムコール 305, 374
 setuid ライブラリルーチン 375
 setup_client コマンド 271
 setup_exec コマンド 271
 setupterm ライブラリルーチン 375

setuseraudit システムコール 306
 setusershell ライブラリルーチン 375
 setvbuf ライブラリルーチン 375
 sfconvert ライブラリルーチン 375
 sgconvert ライブラリルーチン 375
 sgetl システムコール 306
 shadow ファイル 60, 66, 425
 shareall コマンド 122, 424
 sharetab ファイル 317, 425
 share コマンド 120, 122, 244, 424
 /share ファイルシステム 119
 shelltool コマンド 271
 shift_lines コマンド 271
 shmat システムコール 306
 shmctl システムコール 306
 shmdt システムコール 306
 shmget システムコール 306
 showfh コマンド 271
 showmount コマンド 120, 271, 424
 shutacct コマンド 271
 shutdown コマンド 92, 96, 97, 271
 shutdown システムコール 306
 sh コマンド 71 - 73, 271
 sigaction システムコール 306, 307
 sigaction ライブラリルーチン 376
 sigaddset ライブラリルーチン 376
 sigaltstack システムコール 307
 sigblock システムコール 306
 sigdelset ライブラリルーチン 376
 sigemptyset ライブラリルーチン 376
 sigfillset ライブラリルーチン 376
 sigfpeset ライブラリルーチン 376
 siginterrupt ライブラリルーチン 376
 sigismember ライブラリルーチン 376
 siglongjmp ライブラリルーチン 351, 376
 sigmask システムコール 306
 signaling_nan ライブラリルーチン 376
 signal ライブラリルーチン 376
 signbit ライブラリルーチン 376
 significand ライブラリルーチン 376
 sigpause システムコール 306
 sigpending システムコール 306
 sigprocmask システムコール 307
 sigprocmask ルーチン 307
 sigsetjmp ライブラリルーチン 374, 376
 sigsetmask システムコール 307
 sigsetops ルーチン 306
 sigstack システムコール 307
 sigsuspend システムコール 307
 sigvec システムコール 307
 single_precision ライブラリルーチン 377
 single_to_decimal ライブラリルーチン 377
 sinh ライブラリルーチン 377
 sin ライブラリルーチン 377
 size コマンド 199, 272
 /skel ディレクトリ 74, 417
 skyversion コマンド 272
 sleep コマンド 222, 272
 sleep ライブラリルーチン 377
 slk_clear ライブラリルーチン 377
 slk_init ライブラリルーチン 377
 slk_label ライブラリルーチン 377
 slk_noutrefresh ライブラリルーチン 377
 slk_refresh ライブラリルーチン 377
 slk_restore ライブラリルーチン 377
 slk_set ライブラリルーチン 378
 slk_touch ライブラリルーチン 378
 /sm.bak ディレクトリ 417
 sm_inter ライブラリルーチン 378
 sm システムファイル 411
 /sm ディレクトリ 417
 /snadm ディレクトリ 116, 422
 snoop コマンド 244
 socket 307
 socketpair システムコール 307
 socket システムコール 307
 sockio ioctl 185, 187
 soelim コマンド 272
 Solaris 2.x ドライバアーキテクチャ 224, 225
 Solaris 7
 アドミンツール 44
 主な変更 48
 Solaris 7 DDI/DKI 30, 218, 219, 225
 Solaris 7 DDI/DKI 準拠, 定義 219
 Solaris 7 SPARC DDI/DKI 30, 218, 219, 225
 Solaris 7 サーバ、SunOS リリース 4 ディスク
 レスククライアントサポ
 ート 131
 Solaris 7 動作環境
 インストール機能 49
 主な変更 34
 Solaris CDE

OpenWindowsから移動する 165
 スタイル・マネージャ 163
 ファイル・マネージャ 164
 フロントパネル 162
 Solaris 動作環境
 SVR4 vs. 24, 26, 31
 移植性 25
 開発者機能 29, 30
 機能 23, 26, 30
 互換性 24, 26
 システム管理者用の機能 27, 29
 スケラビリティ 25, 30
 相互運用性 26
 大規模な組織 26
 メリット 23, 26
 ユーザ機能 27
 Solstice DiskSuite, インストール 52, 53
 sortbib コマンド 272
 sort コマンド 272
 source code ディレクトリ 422
 space ライブラリルーチン 378
 SPARC DDI/DKI 30, 218 - 220, 225
 SPARCserver Manager, インストール 52, 53
 sparc コマンド 272
 SPARC プラットフォーム 30
 SPECFS (特殊デバイスファイルシステム) 105
 spellin コマンド 272
 spell コマンド 272, 420, 421
 /spell ディレクトリ 421
 spline コマンド 272
 split コマンド 272
 splN/splr ペア 222
 spoken メッセージ 25
 spray コマンド 272
 spray ライブラリルーチン 378
 sprint ライブラリルーチン 378
 sputl システムコール 307
 sqrt ライブラリルーチン 378
 srand48 ライブラリルーチン 378
 srandom ライブラリルーチン 378
 srand ライブラリルーチン 378
 /src ディレクトリ 422
 sscanf ライブラリルーチン 378
 ssignal ライブラリルーチン 376, 378
 standard_arithmetic ライブラリルーチン 379
 standend ライブラリルーチン 379
 standout ライブラリルーチン 379
 startup コマンド 272
 statfs システムコール 307
 statmon システムファイル 411
 statvfs システムコール 307
 stat システムコール 307
 store ライブラリルーチン 379
 strcasecmp ライブラリルーチン 379
 strcat ライブラリルーチン 379
 strchr ライブラリルーチン 344, 379
 strcmp ライブラリルーチン 379
 strcoll ライブラリルーチン 379
 strcpy ライブラリルーチン 379
 strcspn ライブラリルーチン 379
 strdup ライブラリルーチン 379
 streamio ioctl 185, 187
 STREAMS 223, 227
 Ethernet ドライバと 204
 ioctl の透過性 223, 224
 Solaris 2.x ドライバアーキテクチャ 224,
 225
 の変更 223
 描写 30
 モジュールの位置 117
 モジュールをブッシュする 224
 用の疑似ファイルシステム 100
 STREAMS 管理ドライバのエントリポイン
 ト 110, 415
 streamtab 構造 224
 strftime コマンド 206
 strftime ライブラリルーチン 380
 strings コマンド 272
 string_to_decimal ライブラリルーチン 380
 strioctl 構造 223
 strip コマンド 199, 273
 strlen ライブラリルーチン 380
 /strmod ディレクトリ 117
 strncasecmp ライブラリルーチン 380
 strncat ライブラリルーチン 381
 strncmp ライブラリルーチン 381
 strncpy ライブラリルーチン 381
 strpbrk ライブラリルーチン 381
 strptime ライブラリルーチン 381
 strrchr ライブラリルーチン 370, 381
 strspn ライブラリルーチン 381
 strtod ライブラリルーチン 381
 strtok ライブラリルーチン 381

strtol ライブラリルーチン 381
 strxfrm ライブラリルーチン 381
 stty_from_defaults コマンド 273
 stty コマンド 266, 273
 stty ライブラリルーチン 381
 subpad ライブラリルーチン 381
 subwin ライブラリルーチン 382
 sum コマンド 273
 Sun common SCSI architecture (SCSA), DDI 専用インタフェース 219
 Sun C コンパイラ 171
 Sun DDI (Sun デバイスドライバインタフェース) 30, 218, 224
 sundiag コマンド 273
 suninstall コマンド 273
 SunOS 4
 Backup Copilot 51
 QuickCheck 51
 Solaris 7 主な変更 34, 48
 Solaris 7 の作業環境 74
 SunOS 5.7 との互換性 61, 74
 コマンドの変更 61
 作業環境、Solaris 7 を使用した 61
 システムデータを復元 60
 ツール、検索 198
 ディスクパーティション情報の保存 51, 52
 ディスクレスクライアント、用の Solaris 7 サーバサポート 131
 デバイスの命名 37
 バックアップファイルシステム 51, 55
 ファイルシステム情報の保存 52
 メタデバイス構成情報 52, 53
 SunOS 5.7
 Admintool 29
 アドミンツール 44
 印刷サブシステム 46
 カーネルの構成 40
 カーネル名 39
 互換パッケージ 61, 74
 コマンドの変更 61
 自動マウント 41
 セキュリティ 65
 相互互換性 24
 デバイスの命名 37
 ファイルシステム 38, 40
 プリントサブシステム 142
 SunOS/BSD ソース互換パッケージ 61, 62
 SunShield 基本セキュリティモジュール (BSM) 70
 SunView 75, 419
 /sunview1 ディレクトリ 419
 sunview コマンド 273
 sunview システムファイル 411
 SUNWhinst 132, 133
 Sun WorkShop Debugger 180
 sun コマンド 273
 su コマンド 273
 su ファイル 67
 sv_acquire コマンド 273
 svc_create ライブラリルーチン 383
 svc_destroy ライブラリルーチン 382
 svc_dg_create ライブラリルーチン 383
 svcerr_auth ライブラリルーチン 383
 svcerr_decode ライブラリルーチン 383
 svcerr_noproc ライブラリルーチン 383
 svcerr_noprog ライブラリルーチン 383
 svcerr_progers ライブラリルーチン 383
 svcerr_systemerr ライブラリルーチン 383
 svcerr_weakauth ライブラリルーチン 383
 svc_fd_create ライブラリルーチン 383
 svc_fdset ライブラリルーチン 382
 svc_fds ライブラリルーチン 382
 svc_freeargs ライブラリルーチン 382
 svc_getargs ライブラリルーチン 382
 svc_getcaller ライブラリルーチン 382
 svc_getreqset ライブラリルーチン 382
 svc_getreq ライブラリルーチン 382
 svc_raw_create ライブラリルーチン 383
 svc_register ライブラリルーチン 382
 svc_reg ライブラリルーチン 382
 svc_run ライブラリルーチン 383
 svc_sendreply ライブラリルーチン 383
 svctcp_create ライブラリルーチン 383
 svc_tli_create ライブラリルーチン 383
 svcdp_bufcreate ライブラリルーチン 383
 svcdp_create ライブラリルーチン 384
 svc_unregister ライブラリルーチン 383
 svc_unreg ライブラリルーチン 383
 svc_vc_create ライブラリルーチン 383
 svdtab システムファイル 411
 SVR4
 Solaris vs. 23, 24, 26, 31

サポートされないファイルシステムの形式 106
 データ交換する 126
 svr4.make コマンド 193, 194, 256
 sv_release コマンド 273
 sv_xv_sel_svc コマンド 270
 swab ライブラリルーチン 207, 384
 SWAPFS (スワップファイルシステム) 101, 105, 106
 swapon コマンド 274
 swapon システムコール 307
 swap コマンド 264, 274, 424
 swin コマンド 274
 switcher コマンド 274
 swmtool コマンド
 add_services コマンド vs. 231
 extract_unbundled コマンド vs. 245
 rm_services コマンド vs. 267
 描写 34, 35
 symlink システムコール 307
 symorder コマンド 199, 274
 sync コマンド 274
 sync システムコール 307
 /sys 39, 100
 sysadm メニューユーティリティ 31
 syscall システムコール 308
 sysconf システムコール 291, 293, 308
 sysdef コマンド 85, 241, 249
 sysinfo システムコール 290, 291, 304
 syslog.conf システムファイル 411
 syslogd コマンド 274
 syslog ライブラリルーチン 384, 393
 sys_siglist ライブラリルーチン 384
 systems システムファイル 411
 System V Interface Definition 190
 System V ファイルシステム (S5) 31, 106
 system コマンド 94
 system ライブラリルーチン 384
 sys_unconfig コマンド 274
 /sys ディレクトリ 100, 118

T

t300s コマンド 274
 t300 コマンド 274
 t4013 コマンド 274
 t450 コマンド 274

/tabset ディレクトリ 422
 tabs コマンド 274
 t_accept ライブラリルーチン 384
 tail コマンド 274
 talk コマンド 274
 t_alloc ライブラリルーチン 384
 tanh ライブラリルーチン 385
 tan ライブラリルーチン 385
 tar コマンド
 bar コマンド vs. 233
 ustar コマンド vs. 278
 影響のある変更点 127, 274
 説明 126, 127
 のサポート 124
 tar システムファイル 411
 t_bind ライブラリルーチン 384
 tbl コマンド 274
 tcdrain ライブラリルーチン 385
 tcflow ライブラリルーチン 385
 tcflush ライブラリルーチン 386
 tcgetattr ライブラリルーチン 386
 tcgetpgrp ライブラリルーチン 386
 t_close ライブラリルーチン 384
 t_connect ライブラリルーチン 384
 tcopy コマンド 275
 tcov コマンド 201, 275
 TCP/IP 149
 TCP 選択肯定応答 (SACK) 150
 tcsendbreak ライブラリルーチン 386
 tcsetattr ライブラリルーチン 386
 tcsetpgrp ライブラリルーチン 386
 tdelete ライブラリルーチン 386
 tee コマンド 275
 tektool コマンド 275
 tek コマンド 275
 telldir ライブラリルーチン 386
 tell システムコール 308
 telnet コマンド 275
 tempnam ライブラリルーチン 386
 termcap データベース 100, 411
 terminfo データベース 100, 142, 143, 412, 422, 425
 termio ioctl 185, 188
 termios ioctl 185, 188
 termio インタフェース 342, 381
 term システムファイル 411

/term ディレクトリ 110, 416
t_error ライブラリルーチン 384
test コマンド 275
textdomain ライブラリルーチン 207, 386
textedit_filters コマンド 275
Text Edit tool (OpenWindows) 275
textedit コマンド 275
TEX フィルタ 46, 77
tfind ライブラリルーチン 386
t_free ライブラリルーチン 384
tfsd コマンド 275
TFS (疑似ファイルシステム) 100, 105
/tftpboot ディレクトリ 54
tftp コマンド 275
tgetent ライブラリルーチン 386
tgetflag ライブラリルーチン 387
t_getinfo ライブラリルーチン 384
tgetnum ライブラリルーチン 387
t_getstate ライブラリルーチン 384
tgetstr ライブラリルーチン 387
tgoto ライブラリルーチン 388
threadlist マクロ 181
thread マクロ 181
tic コマンド 275
tigetflag ライブラリルーチン 388
tigetnum ライブラリルーチン 388
tigetstr ライブラリルーチン 388
timegm ライブラリルーチン 388
timelocal ライブラリルーチン 388
times ライブラリルーチン 388
timezone ライブラリルーチン 388
time コマンド 275
time ライブラリルーチン 337, 388
tip コマンド 275
t_listen ライブラリルーチン 385
t_look ライブラリルーチン 385
/tmac ディレクトリ 422
tmpfile ライブラリルーチン 388
tmpnam ライブラリルーチン 388
/tmp 一時ファイルシステム (TMPFS)
 に対するサポート 105
/tmp ディレクトリ 417
/tmp ファイルシステム(TMPFS)
 説明 103
 転送 53
/tm ディレクトリ 417
toascii ライブラリルーチン 388

toc システムファイル 412
_tolower ライブラリルーチン 388, 389
toolplaces コマンド 275
ToolTalk 27
t_open ライブラリルーチン 385
t_optmgmt ライブラリルーチン 385
touchline ライブラリルーチン 389
touchoverlap ライブラリルーチン 389
touchwin ライブラリルーチン 389
touch コマンド 276
_toupper ライブラリルーチン 389
tparm ライブラリルーチン 389
tputs ライブラリルーチン 390
tput コマンド 276
traceoff ライブラリルーチン 390
traceon ライブラリルーチン 390
traceroute ユーティリティ 150
trace コマンド 182, 276
traffic コマンド 276
translate システムファイル 412
t_rcvconnect ライブラリルーチン 385
t_rcvdis ライブラリルーチン 385
t_rcvrel ライブラリルーチン 385
t_rcvudata ライブラリルーチン 385
t_rcvuderr ライブラリルーチン 385
t_rcv ライブラリルーチン 385
troff コマンド 276
 の変更 144
 プリプロセッサ 420
 変更 46, 78
 マクロパッケージ 422
trpt コマンド 276
true コマンド 276
truncate システムコール 308
truss コマンド 182, 200, 276
tr コマンド 276
tsearch ライブラリルーチン 390
tset コマンド 276
t_snddis ライブラリルーチン 385
t_sndrel ライブラリルーチン 385
t_sndudata ライブラリルーチン 385
t_snd ライブラリルーチン 385
tsort コマンド 199, 276
t_sync ライブラリルーチン 385
ttyadm コマンド 47
ttymon ポートモニタ 47, 146

ttyname ライブラリルーチン 390
ttyslot ライブラリルーチン 390
ttysoftcar コマンド 276
ttytab システムファイル 60, 412
tty コマンド 276
TTY デバイスの管理と問題の解決 47
t_unbind ライブラリルーチン 385
tunefs コマンド 276
turnacct コマンド 276
tvconfig コマンド 276
twalk ライブラリルーチン 390
typeahead ライブラリルーチン 390
types システムファイル 412
tzfile システムファイル 412
tzsetup コマンド 276
tzsetwall ライブラリルーチン 390
tzset ライブラリルーチン 390

U

u370 コマンド 277
u3b15 コマンド 277
u3b2 コマンド 277
u3b5 コマンド 277
u3b コマンド 277
uadmin システムコール 302, 307
ualarm ライブラリルーチン 391
/ucbininclude ディレクトリ 422
/ucblib ディレクトリ 192, 193, 422
ufsbootblk コマンド 93
ufsboot コマンド 93, 211, 212
ufsdump コマンド
 説明 124 - 126
 代替コマンド 243, 265, 424
 によって置き換えられたコマンド 125
ufsrestore コマンド
 クイックリファレンス 424
 説明 126
 代替コマンド 266, 268
 によって置き換えられたコマンド 125
UFS ログ 129
ulimit ライブラリルーチン 391
ul コマンド 277
umask コマンド 277
umask システムコール 308
umountall コマンド 107, 120, 424
umount_tfs コマンド 277

umount コマンド 107, 120, 277, 424
umount システムコール 308
unadv コマンド 277
uname コマンド 232, 249, 255, 277
uname システムコール 308
uncompress コマンド 277
unconfigure コマンド 277
unctrl ライブラリルーチン 391
unexpand コマンド 277
ungetch ライブラリルーチン 391
ungetc ライブラリルーチン 391
unget コマンド 199, 277
unifdef コマンド 199, 277
uniq コマンド 277
units コマンド 278
unix 39, 101
unix2dos コマンド 278
Unix-to-Unix Copy (UUCP) 59, 152, 154
unix カーネル名 39, 101, 212
/unix ディレクトリ 50, 94, 118, 210
Unix ファイルシステム (UFS)
 作成する 123
 シリンダグループ 37
 に対するサポート 104
unlink コマンド 278
unlink システムコール 309
unmount システムコール 309
unpack コマンド 278
unshare コマンド 120, 424
unwhiteout コマンド 278
updaters システムファイル 412
update コマンド 278
uptime コマンド 278
user2netname ライブラリルーチン 391
useradd コマンド 75, 76
userdel コマンド 75, 76
usermod コマンド 75, 76
users コマンド 278
user ディレクトリ, デフォルトのルート 417
usleep ライブラリルーチン 392
/usr/4lib ディレクトリ 419
/usr/5bin ディレクトリ 116
/usr/5include ディレクトリ 116
/usr/5lib ディレクトリ 116, 191 - 193
/usr/bin/csh コマンド 71 - 73, 238
/usr/bin/ksh コマンド 71 - 73

/usr/bin/mailx コマンド 77, 255, 421
 /usr/bin/mail コマンド 76, 77, 234, 255
 /usr/bin/rksh コマンド 67
 /usr/bin/rsh コマンド 67, 268
 /usr/bin/sh コマンド 71 - 73, 271
 /usr/bin/sunview1 ディレクトリ 419
 /usr/bin ディレクトリ
 移動された SunOS 4 のファイル 116
 記述 419
 クイックリファレンス 425
 説明 106
 汎用ファイルシステム管理コマンド 106,
 107
 プログラミングツール 198, 200
 /usr/ccs/bin/make コマンド 193, 194, 199,
 256
 /usr/ccs/bin ディレクトリ 194, 198, 419
 /usr/ccs/libp ディレクトリ 192
 /usr/ccs/lib/svr4.make コマンド 193, 194,
 256
 /usr/ccs/lib ディレクトリ 192, 193, 198, 419
 /usr/ccs ディレクトリ 116, 419
 /usr/demo ディレクトリ 419
 /usr/etc/biod コマンド 234, 426
 /usr/etc/nfsd コマンド 259, 425
 /usr/etc/rpc.lockd コマンド 267, 425
 /usr/etc/rpc.mountd コマンド 268, 425
 /usr/etc/ypbind コマンド 281, 425
 /usr/etc ディレクトリ 116
 /usr/games ディレクトリ 419
 /usr/include ディレクトリ 116, 419
 /usr/kvm ディレクトリ 419
 /usr/lib/acct ディレクトリ 420
 /usr/lib/class ディレクトリ 420
 /usr/lib/dict ディレクトリ 420
 /usr/lib/font ディレクトリ 420
 /usr/lib/fs ディレクトリ 109, 110, 420
 /usr/lib/iconv ディレクトリ 420
 /usr/lib/libc ディレクトリ 206
 /usr/lib/libintl ディレクトリ 206
 /usr/lib/libp ディレクトリ 420
 /usr/lib/libw ディレクトリ 205
 /usr/lib/locale ディレクトリ 409, 420
 /usr/lib/lpd コマンド 142, 255, 425
 /usr/lib/lp/lpsched コマンド 425
 /usr/lib/lp ディレクトリ 420
 /usr/lib/mail ディレクトリ 420
 /usr/lib/netsvc/yp/ypbind プロセス 281,
 425
 /usr/lib/netsvc ディレクトリ 268, 420
 /usr/lib/nfs/lockd デーモン 425
 /usr/lib/nfs/mountd サーバ 425
 /usr/lib/nfs/nfsd デーモン 425
 /usr/lib/nfs ディレクトリ 420
 /usr/lib/pics ディレクトリ 420
 /usr/lib/refer ディレクトリ 420
 /usr/lib/rsh コマンド 67
 /usr/lib/saf ディレクトリ 420
 /usr/lib/sa ディレクトリ 420
 /usr/lib/spell ディレクトリ 421
 /usr/lib/uucp ディレクトリ 421
 /usr/lib ディレクトリ 116, 191, 192, 198,
 200, 420
 /usr/local ディレクトリ 421
 /usr/net/servers ディレクトリ 421
 /usr/oasys ディレクトリ 421
 /usr/old ディレクトリ 116, 200, 421
 /usr/openwin ディレクトリ 421
 /usr/opt 101
 /usr/opt ファイル 101
 /usr/sadm/bin ディレクトリ 421
 /usr/sadm/install ディレクトリ 421
 /usr/sadm ディレクトリ 421
 /usr/sbin/static ディレクトリ 421
 /usr/sbin ディレクトリ 109, 110, 116, 421,
 425
 /usr/sccs ディレクトリ 194, 198, 200
 /usr/share/lib/keytables ディレクトリ 421
 /usr/share/lib/mailx ディレクトリ 421
 /usr/share/lib/nterm ディレクトリ 421
 /usr/share/lib/pub ディレクトリ 421
 /usr/share/lib/spell ディレクトリ 421
 /usr/share/lib/tabset ディレクトリ 422
 /usr/share/lib/terminfo データベース 100,
 142, 143, 412, 422, 425
 /usr/share/lib/tmac ディレクトリ 422
 /usr/share/lib/zoneinfo ディレクトリ 422
 /usr/share/lib ディレクトリ 421
 /usr/share/man ディレクトリ 78, 80
 /usr/share/src ディレクトリ 422
 /usr/share ディレクトリ 421
 /usr/share ファイルシステム
 マウント 119

- /usr/snadm ディレクトリ 116, 422
- /usr/ucb/df コマンド 86, 107, 121, 241
- /usr/ucb/du コマンド 86, 121, 242
- /usr/ucbinclude ディレクトリ 422
- /usr/ucblib ディレクトリ 192, 193, 422
- /usr/ucb/mail コマンド 76, 77, 234, 255
- /usr/ucb ディレクトリ 62, 198, 422
- /usr/vmsys ディレクトリ 422
- /usr/xpg2bin ディレクトリ 116
- /usr/xpg2include ディレクトリ 116
- /usr/xpg2lib/libxpg2.a ライブラリ 206
- /usr/xpg2lib/libxpg ディレクトリ 206
- /usr/xpg2lib ディレクトリ 116, 193
- /usr ディレクトリ
 - 記述 417
 - 説明 100, 115
 - の変更 115
- /usr ファイルシステム
 - 説明 102
 - 転送 53
 - 配置の変更 419, 422
 - マウントポイント 417
- ustar コマンド 278
- ustat システムコール 309
- utimes システムコール 309
- utime ライブラリルーチン 392
- utmp システムファイル 412
- uuccheck コマンド 278
- uucico コマンド 278
- uucleanup コマンド 278
- UUCP (Unix-to-Unix Copy) 59, 152, 154
- uucp コマンド 267, 278
 - file deposit ディレクトリ 418
 - 構成情報ディレクトリ 417
 - 補助プログラムとデーモンのディレクトリ 421
 - 待ち行列に登録されたジョブのディレクトリ 418
 - ログファイルとステータスファイルのディレクトリ 154, 419
- uudecode コマンド 278
- uuencode コマンド 278
- uuencode システムファイル 412
- uuglist コマンド 152
- uulog コマンド 278
- uuname コマンド 279
- uupick コマンド 279

- uusched コマンド 279
- uusend コマンド 279
- uustat コマンド 279
- uuto コマンド 279
- uuxqt コマンド 279
- uux コマンド 279

V

- vacation コマンド 279
- vadvise システムコール 309
- valloc ライブラリルーチン 392
- val コマンド 200, 279
- /var/adm ディレクトリ 417
- varargs ライブラリルーチン 392
- /var/crash ディレクトリ 417
- /var/cron ディレクトリ 418
- /var/lp ディレクトリ 418
- /var/mail ディレクトリ 100, 117, 418, 425
- /var/news ディレクトリ 418
- /var/nis ディレクトリ 418
- /var/options ディレクトリ 418
- /var/opt/ ディレクトリ 117, 418
- /var/preserve ディレクトリ 418
- /var/sadm ディレクトリ 117, 418
- /var/saf ディレクトリ 117, 418
- /var/spool/cron ディレクトリ 418
- /var/spool/locks ディレクトリ 418
- /var/spool/lp ディレクトリ 143, 418
- /var/spool/mail ディレクトリ 100, 117, 425
- /var/spool/mqueue ディレクトリ 418
- /var/spool/pkg ディレクトリ 418
- /var/spool/uucppublic ディレクトリ 418
- /var/spool/uucp ディレクトリ 418
- /var/spool ディレクトリ 143, 418
- /var/tmp ディレクトリ 418
- /var/uucp/.Admin/perflog ファイル 154
- /var/uucp/.Admin/security ファイル 154
- /var/uucp/.Admin/アカウントファイル 154
- /var/uucp/.Admin/コマンドファイル 154
- /var/uucp ディレクトリ 419
- /var/yp ディレクトリ 419
- /var ディレクトリ 143, 417
 - 記述 417
 - 説明 100, 103, 117
 - の変更 100, 117, 417, 419

vax コマンド 279
vc コマンド 200
vedit コマンド 279
vfontinfo コマンド 279
vfont システムファイル 412
vfork システムコール 309
vfprintf ライブラリルーチン 392
VFS (仮想ファイルシステムアーキテク
チャ) 103, 109
vgrindefs データベース 412
vgrind コマンド 279
vhangup システムコール 309
vidattr ライブラリルーチン 392
vidputs ライブラリルーチン 392
view コマンド 279
vipw コマンド 279
vi エディタ 279, 418
vi と ex エディタ用のバックアップファイ
ル 418
vlimit ライブラリルーチン 392
vmstat コマンド 280
/vmsys ディレクトリ 422
/vmunix file 210, 212
/vmunix ファイル 41, 50, 94
volcancel コマンド 90
volcheck コマンド 90
volcopy コマンド 107, 125
vold.conf ファイル 90
/vol/dev/aliases/cdrom0 ファイルシステ
ム 89
/vol/dev/aliases/floppy0 ファイルシステ
ム 89
vold コマンド 90, 101
volmissing コマンド 90
/vol ファイルシステム 101, 103
vplot コマンド 280
vprintf ライブラリルーチン 393
vsprintf ライブラリルーチン 393
vswap コマンド 280
vsyslog ライブラリルーチン 393
vtimes ライブラリルーチン 393
vtroff コマンド 280
vwidth コマンド 280
vwprintw ライブラリルーチン 393
vwscanw ライブラリルーチン 393

W

waddch ライブラリルーチン 394
waddstr ライブラリルーチン 394
wait3 システムコール 310
wait4 システムコール 310
waitpid システムコール 310, 311
wait コマンド 280
wait システムコール 310
wakeup コマンド 222
wall コマンド 280
wattroff ライブラリルーチン 394
wattron ライブラリルーチン 394
wattrset ライブラリルーチン 394
wclear ライブラリルーチン 395
wclrtobot ライブラリルーチン 395
wclrtoeol ライブラリルーチン 395
wcstombs ライブラリルーチン 396
wctomb ライブラリルーチン 396
wc コマンド 280
wdelch ライブラリルーチン 396
wdeletefn ライブラリルーチン 396
WebNFS
SDK 31
wechochar ライブラリルーチン 396
werase ライブラリルーチン 397
wgetch ライブラリルーチン 397
wgetstr ライブラリルーチン 397
whatis コマンド 280
what コマンド 200, 280
whereis コマンド 280
which コマンド 280
whoami コマンド 280
whois コマンド 281
who コマンド 278, 280
winch ライブラリルーチン 397
windex データベース 80, 231
winsch ライブラリルーチン 398
winsertln ライブラリルーチン 398
wmove ライブラリルーチン 398
wnoutrefresh ライブラリルーチン 398
wprintw ライブラリルーチン 398
wrefresh ライブラリルーチン 399
writev システムコール 313
write コマンド 281
write システムコール 312
wscanw ライブラリルーチン 399

wsetscrreg ライブラリルーチン 399
wstandend ライブラリルーチン 400
wstandout ライブラリルーチン 400
w コマンド 280

X

xargs コマンド 281
xcrypt ライブラリルーチン 400
xdecrypt ライブラリルーチン 400
xdr_accepted_reply ライブラリルーチン 400
xdr_array ライブラリルーチン 400
xdr_authsys_parms ライブラリルーチン 401
xdr_authunix_parms ライブラリルーチン 401
xdr_bool ライブラリルーチン 401
xdr_bytes ライブラリルーチン 401
xdr_callhdr ライブラリルーチン 401
xdr_callmsg ライブラリルーチン 401
xdr_char ライブラリルーチン 401
xdr_destroy ライブラリルーチン 401
xdr_double ライブラリルーチン 401
xdr_enum ライブラリルーチン 401
xdr_float ライブラリルーチン 401
xdr_free ライブラリルーチン 401
xdr_getpos ライブラリルーチン 401
xdr_inline ライブラリルーチン 401
xdr_int ライブラリルーチン 401
xdr_long ライブラリルーチン 401
xdrmem_create ライブラリルーチン 402
xdr_opaque ライブラリルーチン 401
xdr_pointer ライブラリルーチン 401
xdrrec_create ライブラリルーチン 402
xdrrec_endofrecord ライブラリルーチン 402
xdrrec_eof ライブラリルーチン 402
xdrrec_skiprecord ライブラリルーチン 402
xdr_reference ライブラリルーチン 401
xdr_setpos ライブラリルーチン 401
xdr_short ライブラリルーチン 401
xdrstdio_create ライブラリルーチン 402
xdr_string ライブラリルーチン 401
xdr_u_char ライブラリルーチン 402
xdr_u_int ライブラリルーチン 402
xdr_u_long ライブラリルーチン 402
xdr_union ライブラリルーチン 402
xdr_u_short ライブラリルーチン 402
xdr_vector ライブラリルーチン 402

xdr_void ライブラリルーチン 402
xdr_wrapstring ライブラリルーチン 402
XENIX, Solaris SPARC リリース 31
XENIX セマフォファイルシステム
(xnamefs) 106
xgettext コマンド 206
xget コマンド 281
xlock コマンド (OpenWindows) 254
xnamefs 106
/xpg2bin ディレクトリ 116
/xpg2include ディレクトリ 116
xsend コマンド 281
xset コマンド (OpenWindows) 270
xstr コマンド 281
xtab システムファイル 413, 425
xtom ライブラリルーチン 402
xv_get_sel コマンド 247
XView File Chooser 75
XView Window Toolkit 197

Y

y0 ライブラリルーチン 403
y1 ライブラリルーチン 403
yaccpar コマンド 200
yacc コマンド 200, 281
yes コマンド 281
yn ライブラリルーチン 403
ypaliases システムファイル 413
ypbatchupd コマンド 281
ypbind コマンド 281, 425
ypcat コマンド 281
ypfiles データベース 413
ypgroup システムファイル 413
ypinit コマンド 281
ypmatch コマンド 281
yppasswd コマンド 282
yppasswd システムファイル 413
yppoll コマンド 282
ypprintcap データベース 413
yppush コマンド 282
ypserv コマンド 282
ypset コマンド 282
ypupdated コマンド 282
ypwhich コマンド 282
ypxfrd コマンド 282

ypxfr コマンド 282
yp データベースディレクトリ 419

Z

zcat コマンド 282
zdump コマンド 282
zic コマンド 282
/zoneinfo ディレクトリ 422

あ

アーカイブ, ランダムライブラリに変換する 199
アーキテクチャ特有のディレクトリ情報 419
アーキテクチャ独立型データベース 421
アカウントिंगファイル
UUCP 154
アジア系言語の文字セット 205
アドミンツール
描写 44
アドミンツール コマンド 44
アプリケーション
移植、理由 62
互換性の判断 194, 195
アプリケーションの多重度 29
アプリケーションの並列化 222
アプリケーションを移植する、理由 62
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
ト 417
アンバンドルのソフトウェア 101
アンバンドルのソフトウェア用のファイルシ
ステム 101, 103
アンロード, デバイスドライバ 210
アンロードする, デバイスドライバ 225

い

移植性 25
一時ファイルシステム
転送 54
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 418
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 418

一覧表示

ソフトウェアパッケージのインストー
ル 35
ファイル名と統計 107
イメージコピー, ファイルシステム 107
イメージツール 24, 27
イメージフィルタ、サポートされない 46
インクルードヘッダファイル 419
印刷サブシステム
描写 46
インスタンスのデバイス名 215, 216
インストール 49, 59, 60
インストール後の段階 49, 56, 60
インストールの段階 49, 56
インストール前の段階 49, 51 - 53, 55
概要 29, 49
新機能 49, 50
ソフトウェアパッケージ 34, 35, 420
インターネットサービス
構成情報ディレクトリ 111, 416
セキュリティ 66
ネットワークサービスのディレクト
リ 420

う

ウィンドウ管理サービス 27
ウィンドウシステム、デフォルト 74

え

エラーメッセージ
ファイルシステムコマンド 107
ボリュームマネージャ 90
遠隔手続き呼び出し、管理 48
エンドユーザシステムサポート 34

お

オブジェクトコード逆アセンブラ,
COFF用 200
オプションパッケージ 416, 418
オペレーティング環境
32 ビット 30
64 ビット 30

か

- カーネル 39
 - MT (マルチスレッド) 29
 - アーキテクチャ固有 50
 - カスタマイズ 94
 - 構成 40
 - デバッグする 182
 - 動的 29, 40, 209, 210
 - 名前 39
 - ブートと 94
 - マルチスレッド (MT) 222
 - モジュール 39 - 41, 50, 209, 210
 - レイアウト 41, 210
- 開発者 169, 225
 - STREAMS 223, 225
 - Solaris 機能 29, 30
 - 移行情報の概要 169
 - 国際化 204, 207
 - コンパイラ 171
 - システム構成 209, 212
 - ツールとリソース 185, 200
 - デバイスドライバ 217, 225
 - デバイスの構成 213, 216
 - デバッガ 107, 180
 - ネットワーク機能 203, 204
 - バイナリ互換パッケージ 63, 194, 195
 - リンカ 172, 180
- 開発者システムサポート 34
- 外部ネームサービス要求用エントリポイント 421
- 拡張 UNIX コード (EUC) 205
- カスタマイズ
 - man コマンド検索パス 79, 80
 - カーネル 94
 - ユーザ環境 73, 74
- カスタマイズする
 - ポートモニタ 146
- 仮想ファイルシステムアーキテクチャ (VFS) 103, 109
- カット & ペースト 27
- カレンダーマネージャ 24, 27
- 環境
 - SVR4 vs. 23
- 環境初期設定ファイル
 - カスタマイズユーザ環境 74
- 管理ツール
 - ファイル ディレクトリ 116

き

- キーボード配置の記述テーブル 421
- 疑似端末 (pty) スレーブデバイス, ディレクトリ 110
- 疑似端末スレーブデバイス 415
- 疑似デバイスドライバモジュール用のディレクトリ 117
- 疑似ファイルシステム 100, 101, 105
- 起動 91, 95
- 共通デスクトップ環境 (CDE)
 - セッションマネージャ 161
- 共用オブジェクト 191
- 共用可能なりソースのリスト 425
- 共用ファイルシステム
 - SunOS リリース 4 ディスクレスクライア
ント用の Solaris 7 サー
バ 131
 - 構成情報ディレクトリ 416
 - 説明 122
 - デフォルトのルートの 417
 - 転送 54, 59
 - マウント 119
 - リスト 425
- 共用ライブラリ
 - 作成する 175, 178
 - の変更 190
 - バージョン番号 177, 178, 190

く

- クイックリファレンステーブル 426, 423
- クライアント
 - ディスクレス 131, 139
- クラスタ 34, 50
- グラフィカルユーザインタフェース (GUI)
 - Admintool 44
 - XView Windows Toolkit 197
 - アドミンツール 44
 - ソフトウェアマネージャ 34, 35
 - の採用 24
- グラフィックライブラリ 27
- グループ
 - 管理 76
- グループ管理 29, 75
- グループ生産性ツール 24

け

ゲームのバイナリとデータ 419

検査

ポートモニタの状態 146

検索

NIS+ テーブル 45

用のファイル 122

検索パス

カーネルモジュール 41, 210

リンカ 176, 177

検査する

ファイルシステム 107

プリンタの状態 142

こ

コアシステムサポート 34

構成

カーネル 40

自動構成 29

新機能 49, 50

構文, ファイルシステムコマンド 108, 109

互換性

SunOS 4 と SunOS 5.7 61, 74

アプリケーションの判断 194, 195

相互ハードウェア 24, 26

互換パッケージ 61

国際化 204, 207, 420

個人生産性ツール 24, 27

コマンド

local 421

NFS 150

Service Access Facility (SAF) 145, 146

クイックリファレンス 424

クイックリファレンステーブル 423

サービスアクセス機能 (SAF) 47, 48

デバイスドライバ 225

表 227

ファイルシステム 106, 109, 118

プリンタ 143, 144

変更 61

ボリュームマネージャ 90

メッセージ化 206

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 35

コマンドツール (OpenWindows) 237

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 75

コマンドリファレンステーブル 227, 282

コマンドログファイル (UUCP) 154

コマンドを確認 143

コミュニティサービスメッセージ 418

コメントセクションオブジェクトファイル 200

コンパイラ 171, 200

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 101, 103

サービスアクセス管理 (SAC) 47

サービスアクセス機能 (SAF) 47

コマンド 47

サービスアクセス管理 (SAC) 47, 48

描写 47

再構成ブート 213

サイト固有のコマンド 421

削除

NIS+ 情報 45

ソフトウェアパッケージ 34, 35

ネットワークデバイス 47

削除する

ソフトウェアパッケージ 197

デバイス 212, 225

ポートモニタ 146

ポートモニタサービス 146

作成する

ファイルシステム 107, 123

し

シェル

機能 72, 79

初期設定 73

制限付きシェル 67

デフォルトの選択 71, 72

デフォルトホームディレクトリの起動
ファイル 74, 79

シェルツール (OpenWindows) 271

時間帯情報 422

システムアカウント

ユーティリティ概要 28

システム管理関連のディレクトリ 421

システム管理者

Solaris 機能 27, 29

- セキュリティ手続き 65
- システム構成
 - データを復元 60
 - デフォルト 111
 - の変更 209, 212
- システム構成情報のアカウント用のディレクトリ 416, 417
- システムコールのリファレンステーブル 313
- システムコールリファレンステーブル 283
- システム情報ユーティリティ 28
- システム動作レポートパッケージ 420
- システムファイルリファレンステーブル 405, 413
- システムヘッダファイル, 用ディレクトリ 223
- 実行可能ファイル作成 175
- 実行状態の変更操作のディレクトリ 112
- 自動構成 29
- 自動セキュリティ拡張ツール (ASET) 29, 69
- 自動マウント 41, 119 - 121, 233
- 自動ロード、デバイスドライバ 30
- シャットダウン
 - fastboot コマンド 92, 98, 245
 - fasthalt コマンド 96, 98, 245
 - halt コマンド 92, 96, 98, 248
 - reboot コマンド 92, 96, 98, 265
 - shutdown コマンド 92, 96, 97
 - shutdown システムコール 306
- シャドウファイル 115
- シャットダウン
 - fasthalt コマンド 94
- 順序、Solaris 7 ネットワークのインストール 55
- 状態監視情報 417
- 状態ファイル 419
- 商標ファイル 417
- 情報の報告
 - ソフトウェアパッケージ 35
 - ディスクデバイス 85, 87
 - デバイスの管理 28, 85, 87
- 情報報告
- 初期設定状態、変更 94, 96
- 初期設定ファイル
 - ユーザ環境のカスタマイズ 73
- ジョブの処理順序 152, 153
- シリアルポート管理 47
- シリアルポート、制御 60
- シリンダグループ 37

- シリンダグループマップ 37
- シングルユーザシステム, シャットダウン 94
- シングルユーザの実行レベル 95, 96

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 421
 - rc 94 - 96, 100, 111, 115, 212, 265, 416, 426
 - spell コマンド 421
 - アカウント 420
 - システム動作レポートパッケージ 420
 - 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 417
 - ソフトウェアパッケージ 197
 - パッケージ管理 421
- スケーラビリティ 25, 30
- スタイル・マネージャ 166
- スタイル・マネージャのカスタマイズ 166
- スワップデバイス, デフォルト 101, 106

せ

- 正規表現コマンド 200
- 制限
 - リソース 190, 192
- 生成する
 - ソフトウェアパッケージ 196
- 性能ログファイル (uucp) 154
- セキュリティ 65
 - ASET 29, 69
 - Kerberos 69
 - PAM 70
 - SunShield 70
 - アンバンドル 69
 - 概要 65
 - 機能 65
 - 制限付きシェル 67
 - 変更点 65, 232
 - レベル 29, 69
- セキュリティログファイル (uucp) 154
- 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 421
- 全体ディストリビューション 34

そ

- 相互運用性 26
- 相互互換性 24
- 装置の速度, 改善 105
- ソースコード管理システム (SCCS) 194
- ソケット 204
- ソフトウェア
 - サードパーティとアンバンドル 101, 103
- ソフトウェアグループ 50

た

- 大規模な組織、Solaris にとってのメリット 26
- 多重度 29
- タブ設定用エスケープシーケンス 422
- 端末管理 47, 144, 147
- 端末記述ファイル 422
- 端末装置, 用の ディレクトリ 110
- 端末デバイス 416

ち

- チェック
 - ソフトウェアパッケージのインストール 35, 197
 - ファイルシステム 124
 - プリンタの状態 423

つ

- 追加
 - NIS+ 情報 45
 - ソフトウェアパッケージ 34, 197
 - デバイス 87, 88, 213, 214, 225
 - ネットワークデバイス 47
 - ポートモニタ 146
 - ポートモニタサービス 146
- ツール (開発者用) 185, 200
 - Binary 互換パッケージ 61
 - ioctl 要求 185, 188
 - make コマンド 193, 194
 - OLIT 197
 - ptrace 要求値 188, 189
 - SCCS 194
 - XView 197
 - 検索 198, 200
 - ソフトウェアパッケージ 195

- バイナリ互換パッケージ 62, 194, 195
- ライブラリ 190, 193

- ツール (開発者)
 - ソフトウェアパッケージ 197
- ツールキット 197

て

- ディスクスライス 36
- 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 51
 - 情報の報告 85, 87
 - パーティション情報、保存 51, 52
 - 命名規則 84, 85, 215
 - レポートする情報 107, 121
- ディスクドライブ
 - ディレクトリ 110
- ディスクレスクライアント、SunOS 4 をサポートする Solaris 7 サーバ 131
- ディスクレスファイルシステム 121
- 低速装置, 性能の改善 100, 105
- ディレクトリ 101
 - カーネルモジュール 41, 210
 - 監視する 121
 - デフォルト 102, 103
 - 名前 99
 - の変更 99, 110, 112
 - ファイルシステムに対するパス名の作成 107
 - モジュール 101
 - リモート自動マウント 41
 - リンク時に検索する 176, 177
- データアクセス, 共通 100
- データ型, 拡張された 30
- データの転送
 - 復元 56, 61
 - 保存 51
- データベース
 - bootparams 406
 - gettytab 408
 - hosts 408
 - inetd.conf 408
 - locale 409, 420
 - netmasks 409
 - networks 409

- NIS+ 418
- phones 409
- printcap 60, 100, 142, 143, 410, 425
- protocols 410
- publickey 410
- rpc 411
- spell コマンド 420, 421
- termcap 100, 411
- terminfo 100, 142, 143, 412, 422, 425
- vgrindefs 412
- whatis 80, 231
- windex 80, 231
- yp 413, 419
 - アーキテクチャ独立型 421
 - ソフトウェアパッケージ管理ユーティリティ 418
 - 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 117
 - プリンタサブシステム 420
 - プリントキャップ 46
- テープデバイス
 - raw 用のディレクトリ 110
 - リモートへのバックアップ 126
- デーモンクイックリファレンス 425, 426
- デスクトップ
 - 概要 162
- デスクトップ統合サービス 27
- デバイス管理
 - 自動ロード 30
 - 情報の報告 28
 - デバイスのアンロード 40
 - デバイスロード 30
 - デバイスをロードする 40
 - ボリュームマネージャ 48
 - 命名規則 37
- デバイス構成 60
- デバイス構成ライブラリ 30
- デバイスツリー 224, 225
- デバイスドライバ 217, 225
 - DDI 準拠 219
 - nexus ドライバ 224
 - STREAMS 223, 225
 - 移行に関する留意点 221, 223
 - インタフェース互換性 30
 - インタフェースの種類 30, 218, 219
 - コマンド 225
 - ディレクトリ 117
 - の変更 217
 - リーフドライバ 224, 225
- デバイスドライバインタフェース (DDI) 30
- 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) 30
- デバイスドライバのアンロード 40
- デバイスドライバの移行 223
- デバイスドライバのロード 30, 40, 210, 225
- デバイスの管理 83, 90
 - 情報の報告 85, 87
 - 性能の改善 100, 105
 - デバイスの削除 225
 - デバイスの追加 88, 214, 225
 - デバイスのロード 210, 225
 - デバイスをアンロード 210, 225
 - デバイスを削除する 212
 - デバイスを追加 87
 - ボリュームマネージャ 88, 90
 - 命名規則 83, 85, 213
 - 用のボリューム管理 101
 - レポートする情報 220, 221
- デバイスの構成 111, 213, 216
 - デバイスの追加 213
 - 命名規則 216
- デバイスの速度, 改善する 100
- デバイスのツリー 213
- デバイスの物理名 214
- デバイスの命名
 - インスタンス名 215, 216
 - 開発者に関する 213, 216
 - ディスク 215
 - 物理名 214
 - 変更 37
 - 論理名 215
- デバイスの論理名 215
- デバイス名
 - CD-ROMS 85
 - ディスク 84, 85
 - 変更 83, 85
- デバッグ 107, 180, 201
- デバッグ (Sun WorkShop) 180
- デフォルト
 - root アクセス 67
 - ウィンドウシステム 74
 - シェル 71, 72
 - 自動マウント 103

スワップデバイス 101, 106
ディレクトリ 66, 102, 103, 416
のディレクトリ 111
ファイルシステム 102, 103, 109
ボリューム管理ファイルシステム 101
テンプレートファイル、シェルの初期設定 74

と

動的カーネル
定義された 40, 209, 210
の定義 29
動的リンク 30
特殊デバイスファイルシステム(SPECFS) 105
ドライバカーネルインタフェース (DKI) 30
ドラッグ&ドロップ 27

な

名前
カーネル名 39
システムに接続しているディスク 51
ディレクトリ 99
デバイス 37, 83 - 85, 213 - 216
ファイル 57, 99, 107
明示的, ファイル記述子を使用してファイル
をオープン 100
ライブラリ 191 - 193

ね

ネームサービススイッチ 45
ネットワーク
Solaris 7 の機能 203, 204
Solaris 7 インストールの注文 55
サービス管理 47, 149, 154
トランスポート独立ネットワークサービスの構成情報 416
バックアップ 126
ブート 94, 211, 212
ネットワークデバイス管理 47
ネットワークライブラリ 191

の

ノードデバイスツリー 224, 225

は

バージョン番号, 共用ライブラリ 177, 178, 190
バイナリ 421
バイナリ互換パッケージ 61, 62, 194, 195, 419
バイナリのディレクトリ 416, 419
パイプファイル用の疑似ファイルシステム 100
パス名
ファイルシステムに対して作成する 107
パスワード
配置 60, 66, 409, 425
変更 66, 68
有効期限 66, 67
パスワード有効期限 66
バックアップ
インストール 51, 55
変更点 124, 125
パッケージ
インストール 34, 35
オプション 416, 418
開発者と 195, 197
管理 421
互換パッケージ 61
コンポーネント 195, 196
削除 34, 35
削除する 197
情報の報告 35
スプールされたディレクトリ 418
生成する 196
追加 197
定義 34
定義された 195
メリット 34, 50
レポートする情報 197
パッチの管理 35
パフォーマンスメータツール
(OpenWindows) 262
汎用ファイルシステム管理コマンド 106
汎用ファイルシステムコマンド 109
ひ
標準化 23, 25, 26
標準システムコマンド 419
標準プロセスクイックリファレンス 425, 426

ふ

ファイル

- 環境初期設定 73, 74
- 監視する 121
- クイックリファレンス 425, 426
- 結合する 57, 58
- システム 405, 413
- 名前 57, 99, 107
- パイプ, 用の疑似ファイルシステム 100
- バックアップ 51, 55, 124
- ファイル記述子を使用してファイルをオープン 100
- ファイルを検索する 122
- 復元 126
- 変換 58
- 変更 60
- 保存するファイルのリストを作成する 53
- レポートする情報 107
- ファイル記述子ファイルシステム(FDFS) 100, 105
- ファイル記述子を使用してファイルをオープン 100
- / ファイルシステム 53, 102, 415, 419
 - CD-ROM デバイス 88
 - /etc/vfstab ファイルに指定する 121
 - System V 31, 106
 - エラーメッセージ 107
 - 仮想アーキテクチャ(VFS) 103, 109
 - 監視する 121
 - 疑似 100, 101, 105
 - 形式の判定 107
 - 検査する 107
 - コマンド 106, 108, 109, 118
 - 作成する 107, 123
 - サポートされない SVR4 106
 - サポートされる形式 104, 106
 - 実装するカーネルモジュール 117
 - 自動マウント 41, 119 - 121, 233
 - 情報の保存 52
 - シンダグループ 37
 - チェック 124
 - 追加 101
 - ディスクスライスとパーティション 36
 - デバッグ 107
 - デフォルト 102, 103, 109
 - のイメージコピー 107
 - の変更 99, 101
 - パス名のリストの生成 107
 - バックアップ 51, 55, 124
 - 復元 126
 - フロッピーディスクのデバイス 88
 - 変更 51, 57, 99
 - 保存するシステムのリスト 53
 - マウント 41, 59, 107, 119, 121, 258, 298, 424
 - マウント、自動マウント 41, 119 - 121, 233
 - マウント、マウントテーブル 425
 - マウントを解除する 107, 120
 - ユーティリティ概要 28
 - ラベル 107, 125
 - リモート、自動マウント 41
 - リモート、デフォルトの形式 109
 - リモート、マウント 107, 119
 - リモート、マウントを解除する 107
 - レイアウトの変更 415, 422
- ファイルシステムの監視 121
- ファイルとファイルシステムの復元 56
- ファイルマネージャ 24, 48, 88
- ブート 91, 95, 211, 212
 - PROM からの 93
 - 変更点 91, 93, 211, 212
- ブートする
 - 変更点 94
- ブートファイルシステム(BFS) 106
- ブートブロック 93, 211, 212
- フォント
 - デバイスに依存しない 77
- 復元 60
 - SunOS リリース 4 システムデータ 60
 - ファイルシステム 126
- プリンタ 141, 144
 - 印刷ジョブの中止 46
 - 印刷ジョブをキャンセルする 235, 423
 - 印刷ジョブを取り消す 143
 - 構成 60, 111
 - 状態のチェック 142, 423
 - スプールファイルディレクトリ 418
 - 設定 46, 142, 144, 423
 - 設定、コマンド 144
 - デーモンファイル 425
 - ネットワーク要求の管理と問題の解決 47
 - 変更 46, 142
- プリンタサブシステム

構成情報ディレクトリ 416
バックエンドの実行可能ディレクト
リ 420
ログ情報ディレクトリ 418
プリントサブシステム
database ディレクトリ 420
プログラムを静的にリンクしたディレクト
リ 421
プロセス間通信ユーティリティ 28
プロセス管理
ユーティリティ 28
プロセスの管理
疑似ファイルシステム 100
ブロックディスクデバイス 415
ブロックディスクデバイス用ディレクト
リ 110, 215
フロッピーディスク装置、管理 88
フロッピーディスクデバイス、管理 48
フロッピーディスクのファイルシステム、自
動的 88
フロッピーデバイス、管理 90
プロパティウィンドウ (OpenWindows) 237,
240, 256, 270
プロパティシート (OpenWindows) 250
プロファイルライブラリ 420
分散ファイルシステム (DFS) の管理 150
文書ツールの使用 77, 78

へ

並列化 222
ヘッダファイル、用のディレクトリ 223
変換
アーカイブ からランダムライブラリ 199
ファイル 58
変更
NIS+ 情報 45
シェル 71, 72
システム 実行レベル 95
パスワード 66, 68
ファイル 60
変更する
システム 実行レベル 96

ほ

ポイントツーポイント・プロトコル
(PPP) 151

ポートモニタ
管理 47, 145, 146
説明 146
ポートモニタサービス管理 47, 145, 146
ポートモニタサービスを使用可能にする 146
ポートモニタサービスを使用不可にする 146
保存
ディスクパーティション情報 51, 52
ファイルシステム情報 52
メタデバイス構成情報 52, 53
ボリューム管理
用のデフォルトのファイルシステム 101
ボリュームマネージャ
エラーメッセージ 90
概要 48, 88
構成ファイル 90
コマンド 90
使用 88, 90

ま

マウント
ファイルシステム 41, 59, 107, 120, 121,
233, 258, 298, 424, 425
マウントされているリソースのリス
ト 425
リモートリソース 107, 119
マウントする
ファイルシステム 119
マウントを解除する
ファイルシステム 107, 120
リモートリソース 107
マクロ
kadb 181, 182
nroff 422
troff 422
マスタロック 222
マニュアルページ
構成の変更 78
マルチ OS オペレーション 132
マルチスレッド (MT), 定義 222
マルチスレッド (MT) カーネル 29, 222
マルチメディアメール 24, 25
マルチユーザ システム, シャットダウン 96,
97
マルチユーザの実行レベル 95

め

- メール
 - 使用 76, 77
 - マルチメディアメール 24, 25
 - メールの管理 43
 - メールボックススプールディレクトリ 43
 - メタデバイス構成情報 52, 53
 - メッセージカタログ 205
 - メニュー、sysadm メニューユーティリティ 31

も

- 文字セット 205
- 文字列、ソースファイルから抽出する 200
- モデム管理 47, 144, 147
- 問題の解決
 - TTY デバイス 47
 - サービスアクセス管理 (SAC) 47
 - ネットワークから要求された印刷サービス 47

ゆ

- ユーザ、Solaris 機能 27
- ユーザアカウント、追加 76
- 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 68
- ユーザー環境の管理
 - デフォルトシェルの選択 72
- ユーザアクセス制御 76
- ユーザ環境の管理 71, 81
 - man コマンド検索パス構成 80
 - man コマンド検索パスのカスタマイズ 79
 - Solaris 7 での SunOS 4 作業環境の使用 74
 - ウィンドウシステム 74
 - デフォルトシェルの選択 71
 - 文書ツールの使用 77, 78
 - マニュアルページディレクトリ構成の変更 79
 - マニュアルページの構成の変更 78
 - メールの使用 76, 77
 - ユーザとグループの管理 75
- ユーザ管理ユーティリティ 28
- ユーザとグループ追加 75
- ユーザの環境管理

Solaris 7 での SunOS 4 作業環境の使用 61

- ユーザの管理とグループ 75
- ユーザ用の機能 27

よ

- 用語集 432

ら

- ライブラリ
 - lint 171
 - アーカイブをランダムに変換する 199
 - 共用 175, 177, 178, 190
 - 検索パスの規則 176
 - 検索パスの指定 176
 - 動的リンク 30
 - 名前と位置 191 - 193
 - ネットワーク 191
 - の変更 190, 193
 - プロファイル 420
 - リソースの制限 190, 192
 - ルーチンの表 315, 403
 - ライブラリルーチンリファレンス 315
 - ライブラリルーチンリファレンスの表 403
 - ラストイメージフィルタ 46
- り
 - リアルタイムの優先スケジュール 24
 - リーフドライバ 224, 225
 - リーフノード 224
 - リスト
 - 共用可能なリソース 425
 - 共用ファイルシステム 425
 - マウントされているリソース 425
 - リストを作成する
 - ファイルシステムを保存する 53
 - リソースの制限 190, 193
 - リモート CD-ROM デバイス、install4x プログラムと 134, 135
 - リモートシステム
 - ソフトウェアパッケージ管理 34, 35
 - リモートテープドライブにバックアップ 126
 - リモートファイルシステム
 - 自動マウント 41
 - デフォルトの形式 109

- マウント 107, 119
 - マウントを解除する 107
- リンカ 172, 180
 - 共用ライブラリを作成する 175, 178
 - 実行可能ファイルを作成 175
 - 動的リング 30
 - バージョン番号 177, 178
 - ライブラリの検索パスの規則 176
 - ライブラリの検索パスの指定 176
 - リンクエディタオプションの相違 172, 175
 - 例 178, 180

る

- ルートファイルシステム(/) 102
- ループバックファイルシステム(LOFS) 105

れ

- レポートする情報
 - カーネルモジュール 209
 - ソフトウェアパッケージ 197
 - ディスクデバイス 107, 121
 - デバイスの管理 220, 221

- ファイルシステム 107, 124

ろ

- ログイン、管理 48, 66
- ログインシェル
 - 機能 72, 79
 - 初期設定ファイル 73
 - 制限付きシェル 67
 - デフォルトの選択 71, 72
 - デフォルトホームディレクトリの起動ファイル 74, 79
- ログファイル
 - uucp 154
- ロック
 - mutexes 181, 222
 - 条件変数 222
 - マスタ 222
 - マルチスレッドスタイル 222

わ

- ワークスペースプログラムサブメニューへのアクセス 166
- ワークスペースマネージャ 27