



Solaris のシステム管理 (上級編)

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

Part No: 817-7412-10
2004 年 9 月

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 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。

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

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

Sun, Sun Microsystems, docs.sun.com, AnswerBook, AnswerBook2, SunOS, Solstice, Solstice AdminSuite, Solstice DiskSuite, Solaris Solve, Java, JavaStation, DeskSet, OpenWindows, NFS, および Sun Enterprise SyMON 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。Adobe は、米国 Adobe Systems, Inc. の登録商標です。

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、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「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) は郵政事業庁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なないものとします。

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: *System Administration Guide: Advanced Administration*

Part No: 817-6959-10

Revision A



040712@9061



目次

はじめに	19
1 印刷サービスの管理 (トピック)	23
2 印刷サービスの管理 (概要)	25
印刷の新機能	25
プリンタサポートの拡張	26
プリンタ関連作業の情報の参照箇所	31
Solaris オペレーティングシステムでの印刷	32
Solaris プリンタマネージャ	32
ネームサービススイッチでの印刷サポート	33
プリンタの管理方法の選択	33
LP 印刷サービス	35
ネットワークプリンタの管理	35
プリンタの管理	35
プリンタ定義の設定	35
文字セット、フィルタ、フォーム、およびフォントの管理	36
LP 印刷サービスのカスタマイズ	36
Solaris 印刷クライアントサーバーの処理手順	36
印刷クライアントの処理手順	36
印刷クライアントの使用	37
プリンタ構成資源	38
プリンタサーバーの使用	40

- 3 ネットワーク上のプリンタの計画方法 (概要) 43
 - プリンタ関連作業の情報の参照箇所 43
 - ネットワーク上でのプリンタの分散 44
 - プリンタサーバーと印刷クライアントを割り当てる 44
 - プリンタサーバーの要件と推奨 45
 - スプーリング空間 46
 - ディスク空間 46
 - メモリー 46
 - スワップ空間 47
 - ハードディスク 47
 - プリンタ設定の計画 47
 - プリンタの定義の設定 48
 - プリンタタイプの選択 52
 - ファイル内容形式の選択 53

- 4 プリンタの設定手順 (手順) 57
 - 印刷の設定 (作業マップ) 57
 - Solaris プリンタマネージャによる印刷の設定 58
 - Solaris プリンタマネージャによるプリンタの設定 59
 - Solaris プリンタマネージャと特殊文字 60
 - Solaris プリンタマネージャの起動 61
 - ▼ Solaris プリンタマネージャを起動する方法 61
 - プリンタサーバーの設定 64
 - ▼ Solaris プリンタマネージャを使用して、新しく接続したプリンタを追加する方法 64
 - ▼ LP 印刷サービスコマンドを使用して、新しく接続したプリンタを追加する方法 66
 - 印刷クライアントの設定 70
 - ▼ Solaris プリンタマネージャを使用してプリンタアクセスを追加する方法 70
 - .printers ファイルの設定 71
 - ▼ .printers ファイルを設定する方法 71
 - ネットワークプリンタの追加 72
 - ネットワークプリンタ用のプリンタベンダー提供のソフトウェア 74
 - Sun のネットワークプリンタのサポート 74
 - ネットワークプリンタサポートの呼び出し 74
 - プロトコルの選択 74
 - プリンタノード名の選択 75
 - 宛先 (またはネットワークプリンタアクセス) 名の選択 75

- タイムアウト値の設定 76
- ネットワークプリンタアクセスの管理 76
 - ▼ プリンタベンダー提供のツールを使用して新しいネットワークプリンタを追加する方法 77
 - ▼ Solaris プリンタマネージャを使用して新しいネットワークプリンタを追加する方法 77
 - ▼ LP 印刷サービスコマンドを使用して新しいネットワークプリンタを追加する方法 79
- プリンタ構成情報を変換する 83
- プリンタ構成情報を変換する (作業マップ) 84
 - 既存のプリンタ構成情報の変換 84
 - ▼ Solaris 2.5.1 システムのプリンタ情報を変換する方法 85
 - ▼ lpd ベースのプリンタサーバーを実行するシステムのプリンタ情報を変換する方法 85
- 5 プリンタの管理 (手順) 87
 - プリンタと印刷スケジューラの管理 88
 - プリンタとプリンタアクセスの削除 88
 - ▼ プリンタとリモートプリンタへのアクセスを削除する方法 88
 - プリンタの状態のチェック 91
 - ▼ プリンタの状態をチェックする方法 91
 - 印刷スケジューラの再起動 92
 - ▼ 印刷スケジューラを停止する方法 92
 - ▼ 印刷スケジューラを再起動する方法 93
 - その他のプリンタ定義の設定とリセット 93
 - ▼ プリンタの説明 (備考欄) を追加する方法 93
 - デフォルトプリンタの指定 94
 - ▼ システムのデフォルトプリンタを設定する方法 95
 - バナーページの印刷 95
 - ▼ バナーページを選択可能にする方法 96
 - ▼ バナーページを抑制する方法 97
 - プリンタクラスの設定 98
 - ▼ プリンタのクラスを定義する方法 99
 - 障害の通知の設定 99
 - ▼ プリンタの障害警告を設定する方法 100
 - プリンタの障害回復の設定 101
 - ▼ プリンタの障害回復を設定する方法 102
 - プリンタへのユーザーアクセスを制限する 103

▼ プリンタへのユーザーアクセスを制限する方法	104
印刷要求の管理	105
▼ 印刷要求の状態をチェックする方法	106
プリンタの印刷の有効化と無効化	107
▼ プリンタを使用可能または使用不可にする方法	108
印刷要求の受け付けまたは拒否	109
▼ プリンタへの印刷要求を受け付けるまたは拒否する方法	110
印刷要求の取り消し	111
▼ 印刷要求を取り消す方法	111
▼ 特定のユーザーからの印刷要求を取り消す方法	112
印刷要求の移動	113
▼ 印刷要求を別のプリンタに移動する方法	113
印刷要求の優先順位の変更	115
▼ 印刷要求の優先順位を変更する方法	115
6 文字セット、フィルタ、フォーム、フォントの管理 (手順)	117
文字セットの管理	118
選択可能な文字セット	118
プリンタに装着する文字セット	119
印字ホイールの確認	120
印字ホイールまたはカートリッジの装着の警告	120
▼ 印字ホイールとフォントカートリッジを定義する方法	120
▼ 印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法	121
▼ 印字ホイールまたはフォントカートリッジの装着を促す警告を設定する方法	123
▼ 選択可能文字セットの別名を設定する方法	124
印刷フィルタの管理	126
印刷フィルタの作成	126
印刷フィルタの追加、変更、削除、および復元	126
▼ 印刷フィルタを追加する方法	128
▼ 印刷フィルタを削除する方法	128
▼ 印刷フィルタに関する情報を表示する方法	129
フォームの管理	130
フォームの追加、変更、または削除	131
フォームの装着	131
フォームの追跡	131
フォームの装着に関する警告の定義	132
フォームのチェック	132

フォームへのアクセスの制限	132
▼ フォームを追加する方法	132
▼ フォームを削除する方法	133
▼ フォームを取り外し、装着する方法	134
▼ フォームの装着に関する警告を設定する方法	135
▼ フォームに関する情報を表示する方法	137
▼ フォームの現在の状態を表示する方法	137
▼ フォームへのユーザーアクセスを制限する方法	138
▼ フォームへのプリンタアクセスを制限する方法	139
フォントの管理	140
プリンタ常駐フォントの管理	141
ホスト常駐フォントのダウンロード	141
ホスト常駐フォントのインストールと管理	142
▼ ダウンロードされた PostScript フォントをインストールする方法	142
▼ ホスト常駐 PostScript フォントをインストールする方法	143
7 LP 印刷サービスのカスタマイズ (手順)	145
プリンタポート特性の調整	145
▼ プリンタポート特性を調整する方法	146
サポートされていないプリンタの terminfo エントリを追加する	147
▼ サポートされていないプリンタの terminfo エントリを追加する方法	150
プリンタインタフェースプログラムのカスタマイズ	151
標準プリンタインタフェースプログラム	151
stty モードのカスタマイズ	152
終了コード	152
障害メッセージ	153
カスタマイズされたプリンタインタフェースプログラムの使用方法	154
▼ 独自のプリンタインタフェースプログラムを設定する方法	154
新しい印刷フィルタの作成	156
印刷フィルタプログラムの作成	156
印刷フィルタ定義の作成	159
▼ 新しい印刷フィルタを作成する方法	165
新しいプリンタフォームの作成	167
▼ 新しいフォーム定義を作成する方法	169
8 LP 印刷サービス (リファレンス)	171
プリンタ関連作業の情報の参照箇所	171

LP 印刷サービス	172
LP 印刷サービスの構造	172
LP 印刷サービスのコマンド	180
LP 印刷サービスの機能	181
LP によるファイルの管理とローカル印刷要求のスケジューリングの方法	181
ネットワーク印刷要求のスケジューリング	182
印刷ファイルにフィルタを適用する	183
プリンタインタフェースプログラムの機能	183
lpsched デーモンによる印刷ジョブ状態の確認	183
ログファイルの消去	184
▼ プリンタ要求のログの交換間隔を変更する方法	184
ローカル印刷の処理スケジュール	184
リモート印刷の処理スケジュール	187
9 端末とモデムの管理 (トピック)	191
10 端末とモデムの管理 (概要)	193
端末とモデムの管理に関する新機能	193
端末、モデム、ポート、サービス	193
端末	194
モデム	194
ポート	194
サービス	195
ポートモニター	195
端末とモデムを管理するツール	196
シリアルポートツール	196
サービスアクセス機能 (SAF)	196
11 端末とモデムの設定 (手順)	197
シリアルポートツールによる端末とモデムの設定	197
端末の設定	198
モデムの設定	198
▼ 端末を設定する方法	200
▼ モデムを設定する方法	201
▼ ポートを初期化する方法	202
端末とモデムの問題を解決する方法	202

12	サービスアクセス機能によるシリアルポートの管理 (手順)	205
	サービスアクセス機能 (SAF) の概要	206
	SAF 全体の管理 (sacadm)	207
	サービスアクセスコントローラ (SAC プログラム)	207
	SAC の初期化プロセス	207
	ポートモニターサービス管理 (pmadm)	208
	ttymon ポートモニター	208
	ポートの初期化プロセス	209
	発着信両用サービス	210
	TTY モニターとネットワークリスナーポートモニター	210
	TTY ポートモニター (ttymon)	210
	ttymon とコンソールポート	211
	ttymon 固有の管理コマンド (ttyadm)	212
	ネットワークリスナーサービス (listen)	212
	listen 固有の管理コマンド (nlsadmin)	212
	ttymon ポートモニターの管理	213
	▼ ttymon ポートモニターを追加する方法	213
	▼ ttymon ポートモニターの状態を表示する方法	213
	▼ ttymon ポートモニターを停止する方法	214
	▼ ttymon ポートモニターを起動する方法	214
	▼ ttymon ポートモニターを無効にする方法	215
	▼ ttymon ポートモニターを有効にする方法	215
	▼ ttymon ポートモニターを削除する方法	215
	ttymon サービスの管理	216
	▼ サービスを追加する方法	216
	▼ TTY ポートサービスの状態を表示する方法	217
	▼ ポートモニターサービスを有効にする方法	219
	▼ ポートモニターサービスを無効にする方法	219
	サービスアクセス機能の管理 (リファレンス)	220
	SAF の関連ファイル	220
	/etc/saf/_sactab ファイル	220
	/etc/saf/pmtab/_pmtab ファイル	221
	サービスの状態	222
	ポートモニターの状態	223
	ポートの状態	223

- 13 システム資源の管理 (トピック) 225

- 14 システム資源の管理 (概要) 227
 - システム資源管理の新機能 227
 - df、du、ls の新オプション 227
 - システムパラメータの変更 228
 - Solaris 資源管理 228
 - システム資源管理の概要 (作業マップ) 229

- 15 システム情報の表示と変更 (手順) 231
 - システム情報の表示 231
 - システムが 64 ビット Solaris オペレーティングシステムを実行できるか調べる方法 232
 - 64 ビット Solaris 機能が有効になっているか調べる方法 233
 - システムのプロセッサタイプを表示する方法 234
 - Solaris リリース情報を表示する方法 234
 - 一般的なシステム情報を表示する方法 234
 - システムのホスト ID を表示する方法 235
 - システムにインストールされているメモリーを表示する方法 235
 - 日付と時刻を表示する方法 236
 - システム情報の変更 236
 - ▼ システムの日付と時刻を手作業で設定する方法 236
 - ▼ その日のメッセージを設定する方法 237
 - ▼ システムのホスト名の変更方法 237

- 16 ディスク使用の管理 (手順) 239
 - ファイルとディスク容量の情報の表示 239
 - ファイルとディスク容量の情報を表示する方法 240
 - ファイルサイズの確認 242
 - ▼ ファイルサイズを表示する方法 242
 - ▼ サイズの大きなファイルを見つける方法 243
 - 指定されたサイズ制限を超えるファイルを見つける方法 244
 - ディレクトリサイズの確認 244
 - ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する方法 245
 - ▼ ローカル UFS ファイルシステムのユーザーの所有状況を表示する方法 246
 - 古いファイルまたは使用されていないファイルの検索と削除 247
 - 最新ファイルのリストを表示する方法 247

- ▼ 古いファイルと使用されていないファイルを見つけて削除する方法 248
 - ▼ 一時ディレクトリを一度にクリアする方法 249
 - ▼ core ファイルを見つけて削除する方法 250
 - ▼ クラッシュダンプファイルを削除する方法 250
- 17 ディスク割り当て制限の管理 (手順) 253**
- ディスクの割り当て制限 253
 - ディスク割り当て制限の使用 254
 - ディスク割り当て制限の弱い制限値と強い制限値の設定 254
 - ディスクブロックとファイル制限の相違 255
 - ディスク割り当て制限の設定 255
 - 割り当て制限設定のガイドライン 256
 - 割り当て制限の設定 (作業マップ) 257
 - ▼ 割り当て制限用にファイルシステムを構成する方法 257
 - ▼ 1 ユーザーに割り当て制限を設定する方法 258
 - ▼ 複数ユーザーに割り当て制限を設定する方法 259
 - ▼ 割り当て制限の整合性を確認する方法 260
 - ▼ 割り当て制限を有効にする方法 261
 - 割り当て制限のチェック 261
 - ▼ 割り当て制限を超過したかどうかを確認する方法 262
 - ▼ ファイルシステムの割り当て制限を確認する方法 263
 - 割り当て制限の変更と削除 264
 - ▼ 弱い期間制限値のデフォルトを変更する方法 264
 - ▼ 1 ユーザーの割り当て制限を変更する方法 265
 - ▼ 1 ユーザーの割り当て制限を無効にする方法 266
 - ▼ 割り当て制限を無効にする方法 267
- 18 システムタスクのスケジュール設定 (手順) 269**
- システムタスクを自動的に実行する方法 269
 - 繰り返されるジョブのスケジュールリング (crontab) 270
 - 1つのジョブのスケジュールリング (at) 271
 - 繰り返されるシステムタスクのスケジュールリング (cron) 271
 - crontab ファイルの内容 272
 - cron デーモンのスケジュールリング 273
 - crontab ファイルエントリの構文 273
 - crontab ファイルの作成と編集 274
 - ▼ crontab ファイルを作成または編集する方法 274

crontab ファイルを確認する方法	275
crontab ファイルの表示	276
▼ crontab ファイルを表示する方法	276
crontab ファイルの削除	277
▼ crontab ファイルを削除する方法	277
crontab コマンドの使用制御	278
▼ crontab の使用を拒否する方法	279
▼ crontab の使用を特定のユーザーに限定する方法	279
crontab の使用制限を確認する方法	280
1 つのシステムタスクのスケジューリング (at)	281
at コマンドの説明	281
▼ at ジョブを作成する方法	282
at 待ち行列を表示する方法	283
at ジョブを確認する方法	283
at ジョブを表示する方法	283
▼ at ジョブを削除する方法	284
▼ at コマンドの使用を拒否する方法	285
at の使用の拒否を確認する方法	285
19 システムアカウントिंगの設定と管理 (手順)	287
システムアカウントिंग	287
システムアカウントिंगの動作	288
システムアカウントिंगのコンポーネント	288
システムアカウントिंगの設定	292
▼ システムアカウントिंगを設定する方法	293
ユーザーへの課金	295
▼ ユーザーに課金する方法	295
アカウントिंग情報の管理	296
壊れたファイルと wtmpx エラーを修復する	296
▼ 壊れた wtmpx ファイルの修復方法	296
tacct エラーを修復する	296
▼ tacct エラーを修復する方法	297
runacct スクリプトを再起動する	297
▼ runacct スクリプトを再起動する方法	298
システムアカウントिंगの停止と無効	298
▼ 一時的にシステムアカウントिंगを停止する方法	298
▼ システムアカウントिंगを永続的に無効にする方法	299

20	システムアカウントिंग (リファレンス)	301
	runacct スクリプト	301
	日次アカウントングレポート	303
	日次レポート	304
	日次利用状況レポート	305
	日次コマンド要約レポート	306
	月次コマンド要約	308
	最終ログインレポート	308
	acctcom による pacct ファイルの確認	309
	システムアカウントングファイル	311
	runacct スクリプトが生成するファイル	313
21	システムパフォーマンスの管理 (トピック)	315
22	システムパフォーマンスの管理 (概要)	317
	システムパフォーマンスの管理に関する新機能	317
	DNLC 機能の向上	317
	pargs コマンドと preap コマンド	318
	パフォーマンスツール	319
	新しい FSS スケジューラと FX スケジューラ	319
	システムパフォーマンスに関する情報の参照箇所	320
	システムパフォーマンスおよびシステム資源	321
	プロセスとシステムのパフォーマンス	321
	パフォーマンスの監視	323
	監視ツール	324
23	システムプロセスの管理 (手順)	325
	システムプロセスの管理に使用するコマンド (概要)	325
	ps コマンド	326
	プロセスを表示する方法	327
	/proc ファイルシステムとコマンド	328
	/proc プロセスコマンドを使用したプロセスの管理	329
	▼ プロセスに関する情報を表示する方法	330
	▼ プロセスを制御する方法	331
	プロセスの終了 (pkill)	332
	▼ プロセスを終了させる方法	332
	プロセスクラス情報の管理	333

	プロセスのスケジュール優先順位を変更する (priocntl)	333
	プロセスクラスに関する基本情報を表示する方法 (priocntl)	333
	プロセスのグローバル優先順位を表示する方法	334
	▼ プロセスの優先順位を指定する方法 (priocntl)	334
	▼ タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)	335
	▼ プロセスのクラスを変更する方法 (priocntl)	336
	タイムシェアリングプロセスの優先順位を変更する (nice)	337
	▼ プロセスの優先順位を変更する方法 (nice)	338
	システムのプロセスに関する問題解決方法	339
24	システムパフォーマンスの監視 (手順)	341
	システムパフォーマンスの監視に関する新機能	342
	仮想メモリーの統計情報の表示 (vmstat)	342
	仮想メモリーの統計情報を表示する方法 (vmstat)	343
	システムイベント情報を表示する方法 (vmstat -s)	344
	スワップの統計情報を表示する方法 (vmstat -S)	345
	キャッシュフラッシュの統計情報を表示する方法 (vmstat -c)	345
	各デバイス当りの割り込み数を表示する方法 (vmstat -i)	346
	ディスク使用状況の表示 (iostat <i>n</i>)	346
	ディスクの使用状況を表示する方法 (iostat)	346
	拡張ディスク統計情報を表示する方法 (iostat -xtc)	348
	ディスク容量統計情報の表示 (df)	349
	ディスク容量情報を表示する方法 (df)	349
	システム動作の監視 (sar)	350
	ファイルアクセスをチェックする方法 (sar -a)	350
	バッファ動作をチェックする方法 (sar -b)	351
	システムコールの統計情報をチェックする方法 (sar -c)	353
	ディスク動作をチェックする方法 (sar -d)	354
	ページアウトとメモリーをチェックする方法 (sar -g)	355
	カーネルメモリーの割り当てをチェックする方法 (sar -k)	356
	プロセス間通信をチェックする方法 (sar -m)	358
	ページイン動作をチェックする方法 (sar -p)	359
	待ち行列動作をチェックする方法 (sar -q)	360
	未使用のメモリーをチェックする方法 (sar -r)	361
	CPU の使用状況をチェックする方法 (sar -u)	362
	システムテーブルの状態をチェックする方法 (sar -v)	364
	スワップ動作をチェックする方法 (sar -w)	365

	端末動作をチェックする方法 (sar -y)	366
	システム全体のパフォーマンスをチェックする方法 (sar -A)	368
	システム動作データの自動収集 (sar)	368
	ブート時に sadc コマンドを実行する	368
	sa1 スクリプトを使用して sadc コマンドを定期的に行う	369
	sa2 スクリプトを使用してレポートを生成する	369
	データの自動収集を設定する (sar)	369
	自動データ収集を設定する方法	371
25	Solaris ソフトウェアで発生する問題の解決 (トピック)	373
26	ソフトウェアの問題解決 (概要)	375
	ソフトウェアの問題解決に関する新機能	375
	新しいシステムログローテーション	375
	新しい root アカウントに対するシェルのフォールバック	377
	ソフトウェアの問題の解決方法の参照先	377
	システムクラッシュの問題の解決	377
	システムがクラッシュした場合の対処方法	378
	問題の解決に使用するデータの収集	378
	システムクラッシュを解決するためのチェックリスト	379
	システムメッセージの表示	380
	▼ システムメッセージを表示する方法	381
	システムのメッセージ記録のカスタマイズ	382
	▼ システムのメッセージ記録をカスタマイズする方法	383
	リモートコンソールメッセージングを有効にする	384
	実行レベルの変更中に補助コンソールメッセージングを使用する	385
	対話型ログインセッション中に consadm コマンドを使用する	386
	▼ 補助 (リモート) コンソールを有効にする方法	386
	▼ 補助コンソールのリストを表示する方法	387
	▼ システムリブート後も補助 (リモート) コンソールを有効にする方法	387
	▼ 補助 (リモート) コンソールを無効にする方法	388
27	コアファイルの管理 (手順)	389
	コアファイルの管理 (作業マップ)	389
	コアファイルの管理の概要	390
	構成可能なコアファイルのパス	390
	拡張されたコアファイル名	391

	コアファイル名パターンの設定	391
	setuid プログラムがコアファイルを作成できるようにする	392
	現在のコアダンプ構成を表示する方法	392
	コアファイル名パターンを設定する方法	393
	▼ プロセス別コアファイルパスを有効にする方法	393
	▼ グローバルのコアファイルパスを有効にする方法	393
	コアファイルの問題解決	394
	コアファイルの検査	394
	例 — proc ツールを使用したコアファイルの検査	394
28	システムクラッシュ情報の管理 (手順)	395
	システムクラッシュ情報 (作業マップ)	395
	システムクラッシュ (概要)	396
	システムクラッシュダンプファイル	396
	クラッシュダンプの保存	397
	dumpadm コマンド	397
	ダンプデバイスとボリュームマネージャ	398
	システムクラッシュダンプ情報の管理	399
	▼ 現在のクラッシュダンプ構成を表示する方法	399
	▼ クラッシュダンプ構成を変更する方法	399
	▼ クラッシュダンプを検査する方法	401
	▼ クラッシュダンプディレクトリが一杯になった場合に復元する方法 (省略可能)	402
	▼ クラッシュダンプの保存を無効または有効にする方法	402
29	ソフトウェアで発生するさまざまな問題の解決 (手順)	405
	リブートが失敗した場合の対処	405
	SPARC: 64 ビット Solaris のブートで発生する問題の解決	406
	ルートパスワードを忘れた場合の対処	407
	SPARC: 例 — ルートパスワードを忘れた場合の対処方法	407
	x86: 例 — ルートパスワードを忘れた場合の対処方法	408
	システムがハングした場合の対処	409
	ファイルシステムが一杯になった場合の対処	410
	大規模ファイルまたはディレクトリを作成したために、ファイルシステムが一杯になる	411
	システムのメモリーが不足したために、tmpfs ファイルシステムが一杯になる	411
	コピーまたは復元後にファイルの ACL が消失した場合の対処	411

- バックアップ時の問題の解決 412
 - ファイルシステムのバックアップ中に、ルート (/) ファイルシステムが一杯になる 412
 - バックアップコマンドと復元コマンドが対応していることを確認する 412
 - 現在のディレクトリが間違っていないことを確認する 413

- 30 ファイルアクセスでの問題の解決 (手順) 415
 - 検索パスに関連する問題を解決する (コマンドが見つかりません) 415
 - ▼ 検索パスの問題を診断して解決する方法 416
 - ファイルアクセスの問題を解決する 418
 - ファイルとグループの所有権の変更 418
 - ネットワークアクセスで発生する問題の把握 418

- 31 印刷時の問題の解決 (手順) 419
 - 印刷時の問題解決のヒント 419
 - 出力されない (印刷されない) 場合の対処 420
 - 出力が正しくない場合の対処 421
 - LP コマンドがハングした場合の対処 423
 - プリンタがアイドル状態になった (ハングした) 場合の対処 423
 - 矛盾した状態メッセージの対処 424
 - 印刷時の問題の解決 425
 - プリンタに出力されない問題を解決する方法 425
 - ▼ プリンタハードウェアをチェックする方法 425
 - ▼ プリンタのネットワーク接続をチェックする方法 426
 - ▼ LP 印刷サービスの基本機能をチェックする方法 427
 - ▼ Solaris 印刷クライアントから Solaris プリンタサーバーへの印刷をチェックする方法 430
 - ▼ 出力が正しくない場合の問題を解決する方法 431
 - ▼ LP 印刷サービスのハングを解除する方法 435
 - アイドル状態になった (ハングした) プリンタの問題を解決する方法 436
 - ▼ プリンタの準備ができていないかチェックする方法 436
 - 印刷のフィルタリングをチェックする方法 436
 - ▼ プリンタ障害の後に印刷を再開する方法 437
 - ▼ ローカル待ち行列で停止している印刷要求をリモートプリンタに送信する方法 437
 - ▼ プリンタサーバーの待ち行列で停止する印刷クライアントからの印刷要求を使用可能にする方法 437
 - ▼ 矛盾したプリンタ状態メッセージを解決する方法 438

32	UFS ファイルシステムの不整合解決 (手順)	439
	fsck エラーメッセージ	439
	fsck の一般エラーメッセージ	441
	初期化フェーズでの fsck メッセージ	442
	フェーズ 1: ブロックとサイズの検査のメッセージ	445
	フェーズ 1B: 操作し直して DUPS メッセージを表示する	448
	フェーズ 2: パス名の検査のメッセージ	449
	フェーズ 3: 接続性の検査のメッセージ	454
	フェーズ 4: 参照数の検査のメッセージ	457
	フェーズ 5: シリンダグループ検査のメッセージ	459
	クリーンアップ (後処理) フェーズのメッセージ	461
33	ソフトウェアパッケージで発生する問題の解決 (手順)	463
	ソフトウェアパッケージのシンボリックリンクに関する問題の解決	463
	特定のソフトウェアパッケージのインストールエラー	464
	一般的なソフトウェアパッケージのインストールに関する問題	465
	索引	467

はじめに

本書『Solaris のシステム管理 (上級編)』は、Solaris™ システム管理に関する重要な情報を提供するマニュアルの一部です。このガイドでは、SPARC® プラットフォームおよび x86 プラットフォームにおけるシステム管理について説明します。

このマニュアルでは、SunOS™ 5.9 オペレーティングシステムがインストールされているものとします。さらに、使用するネットワークソフトウェアが設定されているものとします。SunOS 5.9 オペレーティングシステムは Solaris 9 製品の一部で、Solaris 9 には Solaris 共通デスクトップ環境 (CDE) などの多くの機能が含まれています。また、SunOS 5.9 は、AT&T System V リリース 4 オペレーティングシステムに準拠しています。

システム管理者にとって重要な Solaris 9 リリースの新機能については、各章のはじめにある新機能に関する節を参照してください。

注 - Solaris オペレーティングシステムは、SPARC と x86 という 2 種類のハードウェア、つまりプラットフォームで動作します。また、Solaris オペレーティングシステムは、64 ビットと 32 ビットの両方のアドレス空間で動作します。このマニュアルで説明する情報は、章、節、注、箇条書き、図、表、例、またはコード例において特に明記しないかぎり、両方のプラットフォームおよびアドレス空間に該当します。

対象読者

このマニュアルは、Solaris 9 リリースを実行しているシステムの管理者を対象にしています。このマニュアルを読むには、UNIX のシステム管理について 1 ~ 2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

『Solaris のシステム管理』の構成

システム管理マニュアルで説明している内容を次に示します。

マニュアルのタイトル	内容
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントサポート、システムのシャットダウンとブート、ソフトウェア (パッケージとパッチ) の管理
『Solaris のシステム管理 (上級編)』	印刷サービス、端末とモデム、システム資源 (ディスク割り当て、アカウントティング、 <code>crontab</code>)、システムプロセス、Solaris ソフトウェアで発生する問題の解決
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、データのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク、IPv4 と IPv6、DHCP、IP セキュリティ、モバイル IP、IP ネットワークマルチパス
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、LDAP のネーミングとディレクトリサービス
『Solaris のシステム管理 (ネーミングとディレクトリサービス : FNS、NIS+ 編)』	FNS と NIS+ のネーミングとディレクトリサービス
『Solaris のシステム管理 (資源管理とネットワークサービス)』	資源管理、リモートファイルシステム、メール、SLP、PPP
『Solaris のシステム管理 (セキュリティサービス)』	監査、PAM、RBAC、SEAM

Solaris オペレーティング環境の今回のリリースに含まれるオープンソースソフトウェアのライセンス規約、帰属、著作権の記載内容を見るためのデフォルトパスは、`/usr/share/src/freeware-name` または `/usr/sfw/share/src/freeware-name` です。Solaris オペレーティング環境がデフォルト以外の場所にインストールされている場合は、所定のパスを修正して、インストールした場所にあるファイルにアクセスします。

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。URL は、<http://docs.sun.com> です。

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せん。

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。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>system%</code>
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。	<code>system% su</code> <code>password:</code>
<i>AaBbCc123</i>	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、 <code>rm filename</code> と入力します。
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第5章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	<code>sun% grep '^#define \</code> <code>XV_VERSION_STRING'</code>

コード例は次のように表示されます。

■ C シェル

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

■ C シェルのスーパーユーザー

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

■ Bourne シェルおよび Korn シェル

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

■ Bourne シェルおよび Korn シェルのスーパーユーザー

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

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

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

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

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

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。
- このマニュアルでは、「x86」という用語は、Intel 32 ビット系列のマイクロプロセッサチップ、および AMD が提供する互換マイクロプロセッサチップを意味します。
- このマニュアル中の手順を実行したり、例 (コマンド入力、コードなど) を使用する場合には、二重引用符 ("), 左一重引用符 (^), 右一重引用符 (') をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。
- /sbin、/usr/sbin、/usr/bin、/etc ディレクトリにあるコマンドについては、このマニュアルでは絶対パス名で表記していない場合があります。
- このマニュアル中の例は、SunOS 5.9 ソフトウェアの基本部分がインストールされ、バイナリ互換パッケージがインストールされておらず、/usr/ucb が検索パスに設定されていないことを、前提としています。



注意 - /usr/ucb を検索パスに設定する場合は、パスの一番最後に設定してください。ps コマンドや df コマンドなどは、SunOS 5.9 コマンドと /usr/ucb コマンドとで形式やオプションがそれぞれ異なります。

第 1 章

印刷サービスの管理 (トピック)

以下の各章で、印刷サービスの管理について説明します。

第 2 章	ネットワーク上で印刷サービスを管理する場合の概要について説明します。プリンタサーバー、印刷クライアント、LP 印刷サービス、および印刷の新機能について取り上げます。
第 3 章	ネットワーク上で印刷サービスの計画を作成する場合の概要について説明します。ネットワーク上のシステム資源の割り当てやプリンタの定義が含まれます。
第 4 章	システム上でプリンタを設定し、ネットワーク上の他のシステムが利用できるようにする詳しい手順を説明します。
第 5 章	プリンタの削除、印刷ポリシーの設定、印刷要求の管理など、プリンタを管理する詳しい手順を説明します。
第 6 章	文字セット、印刷フィルタ、フォーム、およびフォントを設定し管理する詳しい手順を説明します。
第 7 章	プリンタポート特性の調整や、サポートされていないプリンタの <code>terminfo</code> エントリの追加など、LP 印刷サービスをカスタマイズする詳しい手順を説明します。
第 8 章	LP 印刷サービスの参照情報を説明します。

第 2 章

印刷サービスの管理 (概要)

この章では、Solaris オペレーティングシステム (Solaris OS) のプリンタの管理について説明します。

この章で説明する主な内容は次のとおりです。

- 25 ページの「印刷の新機能」
- 31 ページの「プリンタ関連作業の情報の参照箇所」
- 32 ページの「Solaris オペレーティングシステムでの印刷」
- 35 ページの「LP 印刷サービス」
- 36 ページの「Solaris 印刷クライアントサーバーの処理手順」

印刷の新機能

この節では、Solaris 9 9/04 リリースにおいて拡張されたプリンタサポートについて説明します。

拡張されたプリンタサポートに関する手順については、次の作業関連の章を参照してください。

- 64 ページの「Solaris プリンタマネージャを使用して、新しく接続したプリンタを追加する方法」
- 66 ページの「LP 印刷サービスコマンドを使用して、新しく接続したプリンタを追加する方法」
- 77 ページの「Solaris プリンタマネージャを使用して新しいネットワークプリンタを追加する方法」
- 79 ページの「LP 印刷サービスコマンドを使用して新しいネットワークプリンタを追加する方法」

プリンタサポートの拡張

今回の Solaris リリースでは、幅広いプリンタをサポートするように変更されました。以前の Solaris リリースと比べると、プリンタのサポートは大きく異なります。以前のリリースでは、PostScript™ をネイティブに理解するプリンタを使用しなければ、プレーンな ASCII テキストしか印刷できませんでした。また、サポートされるプリンタタイプ、および、これらのプリンタタイプが PostScript または ASCII テキストに対応するかどうかについての情報も制限されていました。今回のリリースでは、変換プログラムであるラスターイメージプロセッサ (Raster Image Processor, RIP) の追加と、PostScript プリンタ記述 (PostScript Printer Description, PPD) ファイルの使用によって、多様なプリンタでの印刷が可能になりました。

RIP 機能および PPD ファイルを組み込むように、Solaris 印刷サブシステムが変更されました。2つのインタフェーススクリプト `standard_foomatic` と `netstandard_foomatic` が新しく作成されました。これらのインタフェーススクリプトは、Solaris スプーラと Solaris プリンタサーバーのバックエンドプロセスとの間に、汎用の Solaris インタフェースを提供します。

次に、現在サポートされているプリンタタイプの例を示します。

- Lexmark Optra E312
- Epson Stylus Photo 1280
- Canon BJC-55
- QMS magicolor 2+

今回の Solaris リリースにおけるプリンタサポートの拡張には、次の機能が含まれません。

- 26 ページの「ラスターイメージプロセッサ (RIP) のサポート」
- 26 ページの「PostScript プリンタ記述 (PPD) ファイルのサポート」
- 27 ページの「Solaris プリンタマネージャのプリンタ定義画面の変更」
- 29 ページの「`lpadmin` コマンドの新しい `-n` オプション」
- 30 ページの「PPD ファイル使用時の `lpstat` コマンドの出力の変更」

ラスターイメージプロセッサ (RIP) のサポート

RIP を使用すると、PostScript 処理機能を持たないプリンタでも印刷できます。Solaris 印刷ソフトウェアは、プリンタサーバー RIP およびそれをサポートするテクノロジーを提供するようになりました。RIP の動作は画面には表示されません。しかし、適切なプリンタドライバを使用するには、Solaris プリンタマネージャまたは `lpadmin -n` コマンドを使用して、プリンタをそれぞれ構成する必要があります。

PostScript プリンタ記述 (PPD) ファイルのサポート

`lpadmin` コマンド、`lpstat` コマンド、および Solaris プリンタマネージャのプリンタ定義画面は、PPD ファイルの使用をサポートするように変更されました。

次に、この機能に関連する新しいソフトウェアパッケージを示します。

- SUNWa2psr
- SUNWa2psu
- SUNWespgs
- SUNWffiltersr
- SUNWffiltersu
- SUNWfppd
- SUNWgimpprint
- SUNWhpijs
- SUNWimagick
- SUNWpsutils

Solaris プリントマネージャのプリンタ定義画面の変更

Solaris プリントマネージャのプリンタ定義画面は、新しく拡張されたプリンタサポート機能を使用できるように変更されました。今回のリリースでは、Solaris プリントマネージャを使用して次のプリンタ定義を割り当てることができません。

- プリンタタイプ (Printer Type)
- ファイル内容形式 (File Content Type)

Solaris プリントマネージャの次の画面が変更されました。

- 新しいローカルプリンタを設定 (New Attached Printer)
- プリンタのプロパティを変更する (Modify Printer Properties) (接続したプリンタ用の画面)
- 新しいネットワークプリンタを設定 (New Network Printer)
- プリンタのプロパティを変更する (Modify Printer Properties) (ネットワークプリンタ用の画面)

これらの画面では、「プリンタタイプ (Printer Type)」と「ファイル内容形式 (File Content Type)」が次のオプションに変更されました。

- プリンタメーカー (Printer Make)
- プリンタモデル (Printer Model)
- プリンタドライバ (Printer Driver)

次の表に、Solaris プリントマネージャで使用するプリンタ定義とそれらが利用可能かどうか (実行している Solaris ソフトウェアリリースによって異なる) について説明します。

表 2-1 Solaris プリントマネージャで使用するプリンタ定義

プリンタ定義	Solaris 9 9/04 より前のリリースで利用できるか	Solaris 9 9/04 リリースで利用できるか
プリンタ名	できる	できる
プリンタサーバー	できる	できる

表 2-1 Solaris プリンタマネージャで使用するプリンタ定義 (続き)

プリンタ定義	Solaris 9 9/04 より前のリリースで利用できるか	Solaris 9 9/04 リリースで利用できるか
備考欄	できる	できる
プリンタタイプ	できる	できない (デフォルトではできない)
ファイル内容	できる	できない (デフォルトではできない)
プリンタメーカー	できない	できる
プリンタモデル	できない	できる
プリンタドライバ	できない	できる
プリンタ障害通知	できる	できる
オプション	できる	できる
ユーザーアクセスリスト	できる	できる
デフォルトプリンタ	できる	できる
バナーを常に印刷	できる	できる

プリンタ定義「プリンタメーカー」、「プリンタモデル」、および「プリンタドライバ」の詳細については、48 ページの「プリンタの定義の設定」を参照してください。

注 - PPD ファイルの使用は待ち行列ごとの構成オプションであり、必須ではありません。RIP 機能を搭載しているプリンタでは、PPD ファイルは必要ありません。このようなプリンタは、以前の Solaris リリースの場合と同じように構成できます。

以前の Solaris プリンタマネージャのプリンタ定義画面に戻す

パフォーマンスを最大限にするため、Solaris プリンタマネージャは `ppdcache` ファイル内にある PPD ファイルだけを認識します。場合によっては、以前の Solaris プリンタマネージャのプリンタ定義画面オプションに戻すことも可能です。このためには、`/usr/lib/lp/model/ppd/` ディレクトリにある `ppdcache` ファイルの名前を変更します。このファイルの名前を変更するには、まず、スーパーユーザー (あるいは、同等の役割) になります。

`ppdcache` ファイルは、Solaris プリンタマネージャが新しいプリンタ定義画面オプションを表示するために必要なファイルです。このファイルの名前を変更すると、Solaris プリンタマネージャが以前の動作に戻り、以前の画面が表示されます。

注 - 新しいバージョンの Solaris プリンタマネージャを使用して、以前のバージョンの Solaris プリンタマネージャで追加したプリンタを変更することもできます。しかし、ppdcache ファイルの名前を変更した場合、今回の Solaris リリースに含まれるバージョンの Solaris プリンタマネージャで追加したプリンタは変更できなくなります。

lpadmin コマンドの新しい -n オプション

lpadmin コマンドには、新しい -n オプションが追加されました。このオプションを使用すると、PPD ファイルを指定して、新しい印刷待ち行列を作成したり、既存の印刷待ち行列を変更したりできます。

たとえば、foobar という印刷待ち行列を Lexmark プリンタに追加したい場合、次のコマンドを入力します。

```
# lpadmin -p foobar -v device -I postscript -m standard_foomatic \  
-n /usr/lib/lp/model/ppd/Lexmark/Lexmark-Optra_E312-Postscript.ppd.gz
```

lpadmin コマンドに -n オプションを使用するときには、上記の例のように、PPD ファイルへのフルパスとファイル名を指定します。

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

サポートされるプリンタについての情報を見つけるには

サポートされるプリンタについての情報は、/usr/lib/lp/model/ppd ディレクトリにあります。自分のプリンタに必要なファイルが見つからない場合は、独自の PPD ファイルを追加できます。lpadmin -n コマンドを使用して新しい印刷待ち行列を作成する場合、独自の PPD ファイルは任意の場所に格納できます。しかし、Solaris プリンタマネージャを使用して印刷待ち行列を作成する場合、その PPD ファイルのエントリを ppdcache ファイルに含める必要があります。

サポートされるメーカーについての情報は、/usr/lib/lp/model/ppd/manufacturer ディレクトリにあります。このディレクトリで ls コマンドを実行すると、特定のプリンタメーカー用のすべての PPD ファイルの一覧が表示されます。これらのファイルの名前は、サポートされるプリンタモデルを反映しています。

独自の PPD ファイルの指定

lpadmin コマンドに -n オプションを使用して PPD ファイルを指定するときには、その PPD ファイルへのフルパスを指定する必要があります。異なるディレクトリにある PPD ファイルを使用する場合、そのファイルへのフルパスを指定する必要があります。

たとえば、ユーザーのホームディレクトリにある xyz.ppd という PPD ファイルを指定するには、次のコマンドを入力します。

```
# lpadmin -p foo -n /home/user/stuff/xyz.ppd -m \
standard_foomatic -I postscript -v /device
```

PPD ファイル使用時の lpstat コマンドの出力の変更

PPD ファイルを使用して印刷待ち行列を作成した場合、lpstat コマンドの出力には、その待ち行列を使用するように構成されている PPD ファイルが表示されるようになります。PPD ファイルを使用せずに印刷待ち行列を作成した場合、lpstat コマンドの出力は以前と同じです。

次の2つの例に、lpstat コマンドの出力の違いを示します。

次の例では、PPD ファイルを使用せずに印刷待ち行列が作成されています。PPD エントリが「none」であることに注目してください。

```
# lpstat -l -p luna
printer luna is idle. enabled since Wed Apr 14 13:45:24 2004. available.
  Form mounted:
  Content types: any
  Printer types: unknown
  Description:
  Connection: direct
  Interface: /usr/lib/lp/model/standard
  PPD: none
  On fault: write to root once
  After fault: continue
  Users allowed:
    (all)
  Forms allowed:
    (none)
  Banner required
  Character sets:
    (none)
  Default pitch:
  Default page size:
  Default port settings:
```

次の例では、Mitsubishi-CP50_Color_Printer-cp50.ppd.gz という PPD ファイルを使用して印刷待ち行列が構成されていることが、lpstat コマンドの出力からわかります。

```
# lpstat -l -p paper
printer paper is idle. enabled since Tue 30 Mar 2004 01:48:38 PM PST
  available.
  Form mounted:
  Content types: any
  Printer types: unknown
  Description:
  Connection: direct
  Interface: /usr/lib/lp/model/standard_foomatic
  PPD:
  /usr/lib/lp/model/ppd/Mitsubishi/Mitsubishi-CP50_Color_Printer-cp50.ppd.gz
```

```
After fault: continue
Users allowed:
    (all)
Forms allowed:
    (none)
Banner required
Character sets:
    (none)
Default pitch:
Default page size:
Default port settings# lpstat l p <queue>
```

Solaris プリンタマネージャを使用して、PPD ファイルを使用せずに作成した印刷待ち行列を変更することもできます。この場合、Solaris プリンタマネージャは、プリンタメーカー、プリンタモデル、およびプリンタドライバが含まれない、以前のプリンタ定義画面オプションを使用します。

プリンタサポートの拡張の制限

正しいプリンタのメーカー、モデル、およびドライバを選択しているにもかかわらず、印刷ジョブの出力が文字化けを起こすことがあります。この問題は、Solaris x86 システムの /dev/lp パラレルポート経由で印刷しているときに発生します。この問題を修正するには、次のコマンドを実行します。

```
# lpadmin -p printer -H nopush
```

プリンタ関連作業の情報の参照箇所

プリンタ関連作業	参照箇所
Solaris プリンタマネージャと lp コマンドによるプリンタのセットアップ	第 4 章
セットアップ後の lp コマンドによるプリンタの管理	第 5 章
文字セット、フィルタ、フォーム、フォントの管理	第 6 章
LP 印刷サービスのカスタマイズ	第 7 章

Solaris オペレーティングシステムでの印刷

Solaris 印刷ソフトウェアは、ネットワーク上のプリンタへのクライアントアクセスを設定および管理するための環境を提供します。

Solaris 印刷ソフトウェアは、次のコンポーネントから構成されます。

- Solaris プリンタマネージャ。ローカルシステム上やネームサービス内の印刷構成を管理するグラフィカルユーザーインターフェースです。
- LP 印刷サービスコマンド。ローカルシステム上やネームサービス内のプリンタを設定して管理するコマンド行インターフェースです。上記の機能に加え、他の印刷管理ツールにない機能も提供します。

印刷の設定には Solaris プリンタマネージャを使用しますが、Solaris 環境で印刷を完全に制御するためには、LP コマンドの一部を使用する必要があります。詳細は、[第 5 章](#)を参照してください。

Solaris 印刷ソフトウェアの制限事項には次のものがあります。

- 以前の Solaris リリースで `s5` (System V 印刷プロトコル) として定義されたプリンタサーバーはサポートされません。
- 印刷クライアントでの印刷のフィルタリングはできません。

Solaris プリンタマネージャ

Solaris プリンタマネージャは Java ベースのグラフィカルユーザーインターフェースで、ローカルおよびリモートのプリンタ構成を管理できます。このツールは、ネームサービス環境 (NIS、NIS+、フェデレーテッド・ネーミング・サービスを使用した NIS+ (xfn)、およびファイル) 内で使用できます。このツールを使用するには、スーパーユーザーとしてログインする必要があります。

プリンタ構成情報の管理には、Admintool: プリンタ (Printer) よりも Solaris プリンタマネージャの使用をお勧めします。Solaris プリンタマネージャをネームサービスとともに使用すれば、プリンタ情報を一元的に管理できるためです。プリンタ構成情報の格納には、ネームサービスの使用をお勧めします。ネームサービスを使用すると、ネットワーク上のすべてのシステムからプリンタ情報にアクセスできるようになるためです。これにより、印刷管理がより簡単になります。

Solaris プリンタマネージャは、プリンタサーバー、印刷クライアント、ネームサービスデータベースにあるプリンタ情報を認識します。印刷クライアントで Solaris 2.6、Solaris 7、Solaris 8 または Solaris 9 リリースが動作している限り、新しい Solaris プリンタマネージャを使用するために変換作業は必要ありません。

Solaris プリンタマネージャのパッケージは SUNWppm です。

Solaris プリンタマネージャを使用して新しいプリンタを設定する詳しい手順を知りたい場合は、第 4 章に進みます。

ネームサービススイッチでの印刷サポート

ネームサービススイッチファイル `/etc/nsswitch.conf` 内の `printers` データベースは、ネットワーク上のクライアントを出力するために、一元化したプリンタ構成情報を提供します。

ネームサービススイッチファイルに `printers` データベースとそれに対応する情報源を指定すると、印刷クライアントからプリンタ構成情報に自動的にアクセスできるようになるため、この情報を自分のシステムに追加する必要はありません。

ファイル、LDAP、NIS および NIS+ 環境に対応した `/etc/nsswitch.conf` ファイル内のデフォルトの `printers` エントリについて、次の表で説明します。 `nisplus` キーワードは `printers.org_dir` テーブルを表します。キーワード `xfn` は、FNS プリンタコンテキストを示します。

ネームサービスタイプ	デフォルトの <code>printers</code> エントリ
<code>files</code>	<code>printers: user files</code>
<code>ldap</code>	<code>printers: user files ldap</code>
<code>nis</code>	<code>printers: user files nis</code>
<code>nis+</code>	<code>printers: user nisplus files xfn</code>

たとえば、ネームサービスが NIS の場合、印刷クライアントのプリンタ構成情報は次のソースから次に記載する順に検索されます。

- `user` - ユーザーの `$HOME/.printers` ファイルを表します。
- `files` - `/etc/printers.conf` ファイルを表します。
- `nis` - `printers.conf.byname` テーブルを表します。

詳細は、`nsswitch.conf` (4) のマニュアルページおよび『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照してください。

プリンタの管理方法の選択

プリンタ情報をネームサービスに追加すると、ネットワークのすべてのシステムからプリンタにアクセスできるようになります。さらに、プリンタに関するすべての情報が一元化されるため、プリンタ管理が一般に簡単になります。

ネームサービス構成	プリンタ情報を一元化する方法
ネームサービスを使用する	プリンタをLDAP、NIS、NIS+、NIS+ (xfn) データベースのどれかに追加すると、プリンタはネットワーク上のすべてのシステムからアクセスできるようになる
ネームサービスを使用しない	プリンタを追加しても、プリンタ情報はプリンタサーバーの構成ファイルにしか追加されない。したがって、印刷クライアントがそのプリンタを自動的に認識することはできない。 プリンタを必要とする印刷クライアントにはプリンタ情報を追加する必要がある

表 2-2 に、印刷関連の主な作業と、印刷作業に利用できるツールを示します。

表 2-2 Solaris 印刷コンポーネントの機能

コンポーネント	対応するリリース	グラフィカル ユーザーインタ フェース	ネットワークプ リンタの設定	印刷クライアント とプリンタ サーバーの管理	LDAP、NIS、 NIS+、NIS+ (xfn) の使用
Solaris プリン タマネージャ	Solaris 9 およ び互換パー ジョンおよび Solaris Easy Access Server 3.0	あり	できる	できる	できる
Admintool	Solaris 9 およ び互換パー ジョン	あり	できない	できる	できない
lp コマンド	Solaris 9 およ び互換パー ジョン	なし	できる	できる	できる

表 2-1 を使用して、各ネットワーク環境に最適な印刷ツールを決めてから、プリンタの設定情報について第 4 章を参照してください。

印刷構成作業のほとんどは Solaris プリンタマネージャで完了設定できます。ただし、インタフェーススクリプトの作成や独自フィルタの追加など、特別な必要がある場合は、LP 印刷サービスコマンドを使用します。LP コマンドは、Solaris プリンタマネージャの元となるコマンドです。lp コマンドで印刷管理作業をする方法については、第 5 章で説明しています。

LP 印刷サービス

「LP 印刷サービス」とは、ユーザーが作業を続けながらファイルを印刷できるようにするソフトウェアユーティリティの集合です。

LP 印刷サービスの基本情報については、[第 8 章](#)を参照してください。

その他に取り上げられている LP 印刷サービスの事項と参照する章については、次の節で説明します。

ネットワークプリンタの管理

「ネットワークプリンタ」はネットワークに直接接続されているハードウェアデバイスです。ネットワークプリンタは、ネットワークを介して出力デバイスにデータを直接送信します。プリンタやネットワークに接続されたハードウェアには、固有のシステム名と IP アドレスがあります。

一般に、ネットワークプリンタには、プリンタベンダーから提供されているソフトウェアが必要です。プリンタにベンダー支給のソフトウェアがある場合は、プリンタベンダーのソフトウェアを使用します。ネットワークプリンタのベンダーがソフトウェアサポートを提供していない場合には、Sun が提供するソフトウェアを利用できます。このソフトウェアはネットワークプリンタの汎用サポートを提供するもので、必ずしもプリンタが持つすべての機能を利用できるわけではありません。

ネットワークプリンタの設定手順については、[第 4 章](#)を参照してください。

プリンタの管理

プリンタサーバーと印刷クライアントを設定すると、次の管理作業を実行する必要性が頻繁に生じる可能性があります。

- プリンタとリモートプリンタへのアクセスを削除する
- プリンタの状態をチェックする
- 印刷スケジューラを再起動する

プリンタ管理作業を実行する手順については、[第 5 章](#)を参照してください。

プリンタ定義の設定

ネットワーク上でのプリンタの定義は、より効率的な印刷環境をユーザーに提供するための継続的な作業です。たとえば、サイトにあるすべてのプリンタに説明を付ければ、ユーザーはプリンタがどこにあるのかを見つけやすくなります。あるいは、プリンタのクラスを定義することにより、印刷要求を迅速に処理できます。

プリンタ定義の設定については、第3章を参照してください。

文字セット、フィルタ、フォーム、およびフォントの管理

設置要件とネットワーク上のプリンタの種類に応じて、LP 印刷サービスのプリンタ固有の機能を設定して管理する必要があります。たとえば、異なる印字ホイール、フィルタ、フォームをプリンタごとに割り当てることができます。文字セット、印刷フィルタ、フォーム、フォントの設定方法に関する基本情報と手順は、第6章を参照してください。

LP 印刷サービスのカスタマイズ

LP 印刷サービスは、ほとんどのプリンタと印刷ニーズに対応できる十分な柔軟性を持つように設計されていますが、あらゆる状況に対処できるわけではありません。LP 印刷サービスの標準機能では対処できない印刷要求が発生する場合があります。また、LP 印刷サービスによるプリンタの処理方法に当てはまらないプリンタを使用することもあります。

LP 印刷サービスは、次のような方法でカスタマイズできます。

- プリンタポートの特性を調整する
- terminfo データベースを調整する
- プリンタインタフェースプログラムをカスタマイズする
- 印刷フィルタを作成する
- フォームを定義する

LP 印刷サービスのカスタマイズの詳細と手順については、第7章を参照してください。

Solaris 印刷クライアントサーバーの処理手順

この節では、Solaris で印刷がどのように進むかについて概要を説明します。

印刷クライアントの処理手順

次の図に、ユーザーが要求を発行してから印刷されるまでの、印刷要求の流れを示します。

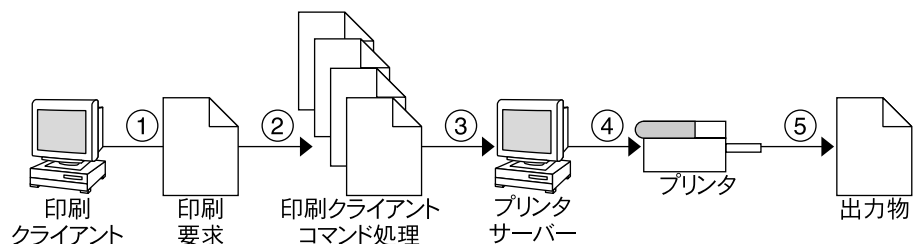


図 2-1 印刷クライアントの処理手順の概要

1. ユーザーは印刷クライアントから印刷要求を出します。
2. 印刷コマンドは印刷構成資源の階層をチェックして、印刷要求をどこに送信するか決定します。
3. 印刷コマンドは、印刷要求を適切なプリンタサーバーに直接送信します。プリンタサーバーは、BSD 印刷プロトコルを受け付ける任意のサーバーであり、SVR4 (LP) プリンタサーバーや BSD lpd ベースのプリンタサーバーなどがあります。
4. プリンタサーバーは印刷要求を適切なプリンタに送信します。
5. 印刷要求が印刷されます。

印刷クライアントの使用

この節では「印刷クライアント」、つまり印刷要求をプリンタサーバーに送信できるシステムと、印刷クライアントが印刷要求を発行するための印刷コマンドを中心に説明します。

図 2-2 に、印刷手順の中で、ユーザーが印刷クライアントから印刷要求を発行する処理を強調して示します。

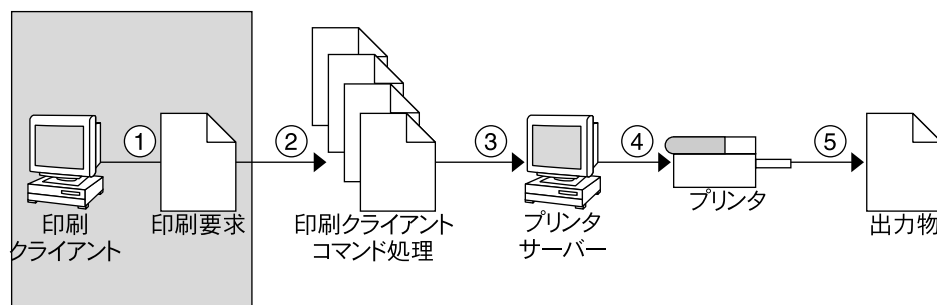


図 2-2 ユーザーが印刷クライアントから印刷要求を発行する

印刷クライアントとは

システムに Solaris 印刷ソフトウェアをインストールして、リモートプリンタにアクセスできるようにすると、そのシステムは印刷クライアントになります。

Solaris 印刷ソフトウェアは、次の資源からプリンタやプリンタ構成情報を見つけます。

- コマンド行インタフェースの `lp -d` コマンド (単独名形式または POSIX 形式)
- ユーザーの `LPDEST` 変数または `PRINTER` 変数
- `/etc/nsswitch.conf` ファイルに `printers` データベースとして設定されたソースの `_default` 変数
- ユーザーの `$HOME/.printers` ファイル
- ローカル `/etc/printers.conf` ファイル
- LDAP または NIS ネームサービスデータベース

印刷クライアントは、その要求をプリンタサーバーの待ち行列に送信します。つまり、クライアントは、ローカルの待ち行列を持ちません。クライアントが印刷要求を一時スプール領域に書き込むのは、プリンタサーバーが利用できない場合か、エラーが発生した場合だけです。サーバーまでの経路がこのように簡素化されているために、印刷クライアントは少ない資源で処理を実行でき、印刷障害の発生する可能性が減り、パフォーマンスが向上します。

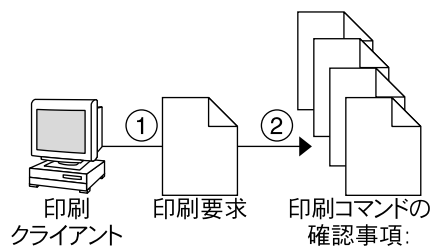
プリンタ構成資源

この節では、印刷ソフトウェアが、プリンタ名とプリンタ構成情報を見つけるのに使用する資源について説明します。

印刷ソフトウェアは、ネットワーク上のすべてのプリンタのプリンタ構成情報を格納するネットワーク (共有) 資源である、ネームサービスを使用できます。ネームサービス (LDAP、NIS、NIS+、または NIS+ (xfn)) は、プリンタ構成情報の管理を簡単にします。プリンタをネームサービスに追加すると、ネットワーク上のすべての印刷クライアントは、そのプリンタにアクセスできます。

印刷ソフトウェアがプリンタを見つける手順

次の図に示すように、印刷ソフトウェアは、プリンタとプリンタ構成情報を見つけるためにより多くのオプションを使用します。



- | | |
|--|--|
| <p>A. 単独形式、POSIX形式、またはコンテキストベース形式のプリンタ名またはプリンタクラス</p> <p>B. デフォルトプリンタを指定するユーザーの PRINTER 環境変数または LPDEST 環境変数</p> <p>C. /etc/nsswitch.conf にあるプリンタデータベース内の _default 変数</p> <p>D. ユーザーの \$HOME/.printers ファイル</p> | <p>E. ローカルの /etc/printers.conf ファイル</p> <p>F. ドメインの 'ou=printers' コンテナにあるプリンタに関する LDAP ディレクトリ情報ツリー</p> <p>G. NIS printers.conf.byname マップ</p> <p>H. NIS+ printers.org_dir テーブル</p> <p>I. NIS+ (xfn) FNS 印刷コンテキスト</p> |
|--|--|

図 2-3 印刷クライアントソフトウェアがプリンタを見つける手順

1. ユーザーは lp コマンドまたは lpr コマンドを使用して、印刷クライアントから印刷要求を出します。ユーザーは、次の3つの形式のいずれかを使用して、宛先のプリンタ名またはプリンタクラスを指定できます。
 - 単独名形式。次の例に示すように、lpコマンドとオプションの後にプリンタ名またはプリンタクラスが続きます。


```
% lp -d neptune filename
```
 - POSIX 形式。次の例に示すように、印刷コマンドとオプションの後に server:printer が続きます。


```
% lpr -P galaxy:neptune filename
```
 - コンテキストベース形式。次の例に示します。


```
% lpr -d thisdept/service/printer/printer-name filename
```
2. 印刷コマンドは、次の手順でプリンタとプリンタ構成情報を見つけます。
 - ユーザーが宛先のプリンタ名またはプリンタクラスを3つの有効な形式のいずれかで指定しているかどうかをチェックします。
 - ユーザーがプリンタ名またはプリンタクラスを有効な形式で指定していない場合、ユーザーの PRINTER 環境変数または LPDEST 環境変数にデフォルトプリンタ名が指定されていないか調べます。
 - どちらの環境変数にもフォルトプリンタが指定されていない場合は、/etc/nsswitch.confファイルにprinters データベースとして設定されたソースを調べます。ネームサービス資源は、次のうちの1つである可能性があります。

- ドメインの ou=printers コンテナ内の LDAP ディレクトリ情報ツリー
- NIS printers.conf.byname マップ
- NIS+ printers.conf_dir マップ
- NIS+ xfn (FNS) 印刷コンテキスト

プリンタサーバーの使用

プリンタサーバーはローカルプリンタが接続されているシステムであり、プリンタがネットワーク上の他のシステムを利用できるようにします。図 2-4 に、印刷手順の中で、プリンタサーバーが印刷要求をプリンタに送信する処理を強調して示します。

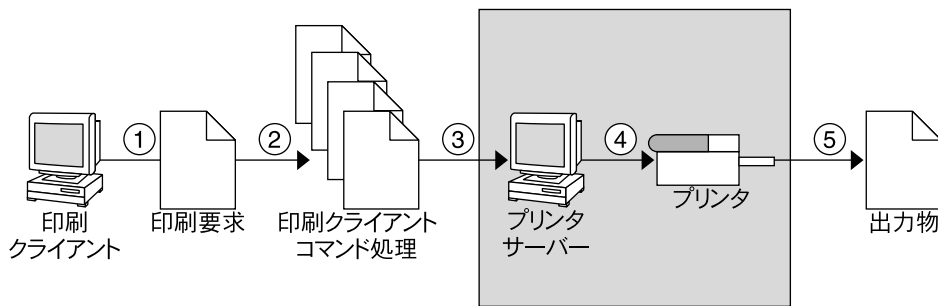


図 2-4 プリンタサーバーが印刷要求をプリンタに送信する

1. 印刷クライアントが印刷要求を発行する。
2. 印刷要求が印刷クライアント上で処理される。
3. 印刷要求がプリンタサーバーに送信される。
4. 印刷要求がプリンタに送信される。
5. プリンタから印刷出力される。

BSD 印刷プロトコル

印刷コマンドは、BSD 印刷プロトコルを使用します。このプロトコルの利点の 1 つは、さまざまなプリンタサーバーと通信できることです。

- lpd ベースのプリンタサーバー
- SunOS 5.9 プリンタサーバーおよび互換性のある SVR4 (LP) プリンタサーバー
- BSD 印刷プロトコルをサポートする、その他のプリンタサーバーやプリンタ

BSD 印刷プロトコルは、さまざまなメーカーの異なるシステム間の互換性を提供するため、業界で広く使用されている標準規格です。Sun は、将来の相互運用性を提供するために、BSD 印刷プロトコルをサポートします。

次に進む手順

Solaris プリンタマネージャを使用して新しいプリンタを設定する詳しい手順を知りたい場合は、[第 4 章](#)に進みます。計画を立てるための情報が必要な場合は、[第 3 章](#)を参照してください。

第 3 章

ネットワーク上のプリンタの計画方法 (概要)

この章では、最も効率よくネットワーク間でプリンタを分散する方法とプリンタの構成を計画する方法について説明します。

この章で説明する主な内容は次のとおりです。

- 44 ページの「ネットワーク上でのプリンタの分散」
- 44 ページの「プリンタサーバーと印刷クライアントを割り当てる」
- 45 ページの「プリンタサーバーの要件と推奨」

プリンタ関連作業の情報の参照箇所

プリンタ関連作業	参照箇所
Solaris プリンタマネージャと lp コマンドによるプリンタのセットアップ	第 4 章
セットアップ後の lp コマンドによるプリンタの管理	第 5 章
文字セット、フィルタ、フォーム、フォントの管理	第 6 章
LP 印刷サービスのカスタマイズ	第 7 章

ネットワーク上でのプリンタの分散

ネットワーク上のプリンタを設定する目的は、ユーザーが1つまたは複数のプリンタにアクセスできるようにすることです。管理者として、各プリンタを1台のシステム専用にするのが効率がよいか、多数のシステムが利用できるようにするのがよいかを判断しなければなりません。ネットワーク環境では、プリンタを複数のプリンタサーバーに分散させます。複数のプリンタサーバーを設定する利点は、あるプリンタサーバーに問題が発生しても、別のプリンタサーバーに印刷要求を振り替えられることです。

集中化した印刷構成を採用した場合も、使い勝手をよくしたり、応答時間を短縮したりするために、プリンタをユーザーのシステムに接続できます。ユーザーのシステムに接続されたプリンタも、ネットワーク上の他のシステムから利用できます。

次の図は、集中化印刷構成を採用した場合もプリンタをユーザーのシステムに接続できる例を示しています。

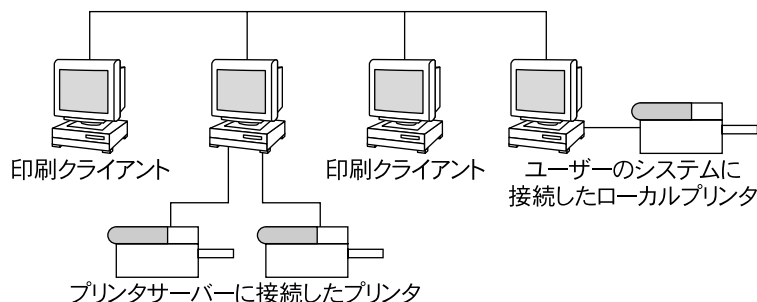


図 3-1 ネットワーク上でプリンタを分散する方法

プリンタサーバーと印刷クライアントを割り当てる

どのシステムにローカルプリンタを接続し、どのシステムでは他のシステム上のプリンタを使用するかを決定する必要があります。ローカルプリンタを接続し、そのプリンタをネットワーク経由で他のシステムでも利用できるようにしているシステムを「プリンタサーバー」と呼びます。プリンタサーバーに印刷要求を送るシステムを「印刷クライアント」と呼びます。

LP 印刷サービスは、Solaris 環境で印刷サービスを管理するソフトウェアです。プリンタをシステムに物理的に接続したら、LP 印刷サービスにプリンタサーバーの特性を定義しなければなりません。プリンタサーバーを設定し終わったら、他のシステムを印刷クライアントとして設定できます。

プリンタサーバーと印刷クライアントは、Solaris リリースのさまざまなバージョンと UNIX オペレーティングシステムのさまざまなバージョンで実行することができます。Solaris 9 リリースとその互換バージョンを実行する印刷クライアントは、SunOS 4.1、BSD UNIX、Linux などの lpd をベースにした印刷サービスを実行するプリンタサーバーに印刷することができます。さらに、lpd をベースにした印刷サービスを実行する印刷クライアントは、Solaris 9 リリースとその互換バージョンを実行するプリンタサーバーに印刷することができます。

以下の図は、Solaris リリースと lpd をベースにした印刷サービスを実行するシステムを備えたネットワークの印刷構成を示しています。Solaris の印刷機能が異機種システム混在環境でどのように動作するのかについては、187 ページの「リモート印刷の処理スケジュール」を参照してください。

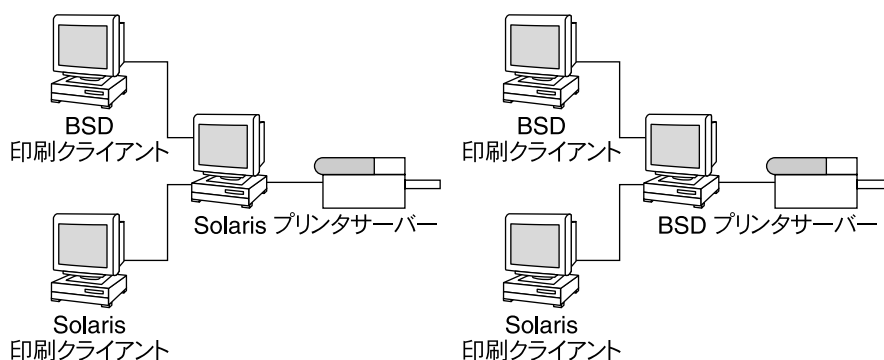


図 3-2 Solaris と lpd をベースにしたシステムの印刷構成

プリンタサーバーの要件と推奨

プリンタは、スタンドアロンシステムまたはネットワーク上の任意のシステムに接続できます。ネットワークに接続されていて、プリンタを備えているシステムは、印刷負荷を管理できる十分な資源があれば、どれでもプリンタサーバーとして使用できます。

スプーリング空間

「スプーリング空間」とは、印刷待ち行列内で要求を格納し、処理するためのディスク空間です。スプーリング空間は、どのシステムをプリンタサーバーとして指定するかを決定する場合の唯一重要な要素です。ユーザーがファイルの印刷要求を出すと、それらのファイルは印刷が完了するまで `/var/spool/lp` ディレクトリに格納されます。`/var` ディレクトリのサイズは、ディスクのサイズとディスクのパーティション分割方法によって異なります。スプーリング空間は、プリンタサーバーの `/var` ディレクトリに割り当てられるか、ファイルサーバーからマウントされてネットワーク上でアクセスされます。

注 - `/var` が別のスライスとして作成されていない場合、`/var` ディレクトリはルート (`/`) ファイルシステムの空間を使用します。これは、不十分なプリンタサーバーの空間になる可能性があります。

ディスク空間

システムをプリンタサーバーの候補として評価するときは、利用できるディスク空間を考慮してください。大きなスプールディレクトリでは、600M バイトのディスク空間を消費することもあります。プリンタサーバーとして指定できるシステム上で、ディスク空間のサイズと分割設定を調べてください。

また、印刷のニーズと印刷クライアントシステムの利用パターンを慎重に調べてください。小さなグループのユーザーが短い電子メールメッセージ、つまり複雑なフォームを必要としない単純な ASCII ファイルだけを印刷する場合は、20 ~ 25M バイトのディスク空間を `/var` ディレクトリに割り当てたプリンタサーバーで十分でしょう。ただし、多数の印刷クライアントユーザーが長い文書、ビットマップ、またはラスタイメージファイルを印刷する場合は、スプーリング空間が頻繁に一杯になります。ユーザーがジョブを印刷待ち行列に入れることができないと、作業の流れは中断してしまいます。より多くのスプーリング空間を要求すると、システム管理者はスプーリング用のディスク空間を追加するか、異なるシステムをグループのプリンタサーバーとして指定しなくてはならない場合があります。

プリンタサーバーの `/var` ディレクトリが小さすぎる場合で、より大きなファイルシステムが利用可能なときは、`/var` ディレクトリ上に大規模ファイルシステムをマウントしてスプーリング空間を追加することができます。ファイルシステムのマウントと `vfstab` ファイルの編集については、『Solaris のシステム管理 (デバイスとファイルシステム)』の「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

メモリー

Solaris 環境を実行するには、少なくとも 64M バイトのメモリーが必要です。プリンタサーバー用にメモリーを追加する必要はありません。ただし、メモリーが多ければ、印刷要求のフィルタリング処理が高速になります。

スワップ空間

プリンタサーバー上のスワップ空間は、LP 印刷サービスを処理するのに十分な量を割り当ててください。スワップ空間を増やす方法については、『Solaris のシステム管理 (デバイスとファイルシステム)』の「追加スワップ空間の構成 (手順)」を参照してください。

ハードディスク

最適のパフォーマンスを発揮させるには、プリンタサーバーにハードディスクとローカル `/var` ディレクトリが必要です。プリンタサーバー用のスプーリング空間をローカルのハードディスク上にマウントしてください。プリンタサーバーに専用のハードディスクとローカルの `/var` ディレクトリがあれば、印刷処理ははるかに高速になります。また、印刷要求の処理に必要な時間をより正確に予測できます。

プリンタ設定の計画

この節では、Solaris OS で印刷するための計画の立て方の概要を説明します。次の情報が含まれています。

- プリンタ名、備考欄、プリンタポートなどのプリンタの定義の設定
- プリンタのメーカー、モデル、およびドライバの選択
- 使用する PPD ファイルの決定
- プリンタタイプとファイル内容の形式の選択

注 - 今回のリリースでは、プリンタタイプとファイル内容形式は Solaris プリンタマネージャで割り当てることができません。Solaris プリンタマネージャを以前の動作に戻す方法については、28 ページの「[以前の Solaris プリンタマネージャのプリンタ定義画面に戻す](#)」を参照してください。

- 障害通知とデフォルトプリンタの宛先の設定
- バナーページを設定するかどうか、あるいはプリンタへのユーザーのアクセスを制限するかどうかの決定
- プリンタクラスと障害回復の設定

プリンタの定義の設定

ネットワーク上でのプリンタの定義は、より効率的な印刷環境をユーザーに提供するための継続的な作業です。たとえば、サイトにあるすべてのプリンタのパラメータを設定すれば、ユーザーはプリンタがどこにあるのかを見つけやすくなります。あるいは、プリンタのクラスを定義することにより、印刷要求を迅速に処理できます。

lpadmin コマンドを使用すると、すべてのプリンタ定義を設定できます。一方、Solaris プリンタマネージャを使用すると、プリンタのインストールまたは変更時にプリンタ定義の一部だけを設定できます。表 3-1 は、プリンタ定義と、その定義を Solaris プリンタマネージャで割り当てることができるかどうかを示しています。

表 3-1 Solaris プリンタマネージャで設定されるプリンタ定義

プリンタ定義	Solaris プリンタマネージャで設定できるか
プリンタ名	設定できる
備考欄	設定できる
プリンタポート	設定できる
プリンタタイプ	設定できない (デフォルトではできない)
ファイル内容形式	設定できない (デフォルトではできない)
プリンタメーカー	設定できる
プリンタモデル	設定できる
プリンタドライバ	設定できる
プリンタ障害通知	設定できる。ただし lpadmin コマンドより機能は少ない
デフォルトプリンタ	設定できる
バナーページの印刷	設定できる。ただし lpadmin コマンドより機能は少ない
プリンタへのユーザーアクセスの制限	設定できる。ただし lpadmin コマンドより機能は少ない
プリンタクラス	設定できない
障害回復	設定できない

プリンタ名

システムにプリンタを追加するときは、その「プリンタ名」を指定します。プリンタ名の必要条件は以下のとおりです。

- 管理ドメイン内のすべてのプリンタ間で一意であること
- 最大 14 文字までの英数字 (ハイフンと下線も含む) であること

- 覚えやすく、プリンタのタイプ、場所、プリンタサーバー名などを識別できること

サイトに合った命名規則を設定してください。たとえば、ネットワーク上で異なるタイプのプリンタを使用する場合は、プリンタ名の一部にプリンタタイプを含めると、ユーザーは適切なプリンタを選択しやすくなります。たとえば、PostScript プリンタは文字 `PS` で識別できます。ただし、サイトのプリンタがすべて PostScript プリンタである場合は、`PS` というイニシャルをプリンタ名の一部として含める必要はありません。

備考欄

`lpadmin -D` コマンドまたは Solaris プリンタマネージャを使用すると、プリンタに説明 (備考欄) を割り当てることができます。プリンタの備考欄には、ユーザーがプリンタを識別できるような情報を含める必要があります。プリンタが設置されている部屋番号、プリンタのタイプやメーカー、印刷に問題がある場合に連絡する担当者名などを含めることができます。

次のコマンドで備考欄を参照することができます。

```
$ lpstat -D -p printer-name
```

プリンタポート

プリンタのインストール時、またはその設定を後から変更するときに、`lpadmin -p printer-name -v device-name` コマンドまたは Solaris プリンタマネージャを使用して、プリンタの接続先となるデバイス、つまり「プリンタポート」を指定できます。

ほとんどのシステムが、2つのシリアルポートと1つのパラレルポートまたは USB ポートを持っています。システムにポートを追加しないかぎり、3台以上のシリアルプリンタおよび2台以上のパラレルプリンタまたは3台以上の USB プリンタを直接接続することはできません。

Solaris プリンタマネージャを使用すると、以下のプリンタポートタイプを選択することができます。これらのオプションには、`lpadmin` コマンドと同じ柔軟性があります。

プリンタポートタイプ	対応するデバイス名のオプション
シリアル	<code>/dev/term/a</code>
シリアル	<code>/dev/term/b</code>
パラレル	<code>/dev/printers/0</code> → <code>/dev/ecpp0</code>
USB	<code>/dev/printers/[1-9]</code>
プリンタサーバーが認識するポート名を指定	Other

LP 印刷サービスは、標準プリンタインタフェースプログラムからの設定を使用してプリンタポートを初期設定します。プリンタインタフェースプログラムの詳細は、126 ページの「印刷フィルタの管理」を参照してください。デフォルト設定で機能しないパラレルプリンタやシリアルプリンタがある場合は、145 ページの「プリンタポート特性の調整」のポート設定をカスタマイズする方法を参照してください。

x86 のみ – x86 システムで複数のポートを使用している場合、デフォルトでは最初のポートだけが有効です。2 番目以降のポートはデフォルトでは無効です。複数のポートを使用するためには、追加の `asy` (シリアル) ポートや `lp` (パラレル) ポートごとに、デバイスドライバのポート構成ファイルを手作業で編集しなければなりません。x86 ポート構成ファイルのパスは、次のとおりです。

```
/platform/i86pc/kernel/drv/asy.conf
```

```
/platform/i86pc/kernel/drv/lp.conf
```

プリンタメーカー

「プリンタメーカー」とは、プリンタの製造業者名です。プリンタメーカーは、プリンタ本体、梱包材、および同梱のマニュアルなどに印刷されています。

次に、今回のリリースで利用できるプリンタメーカーの例を示します。

- Lexmark
- Epson
- Canon
- QMS
- Xerox

プリンタモデル

プリンタの製造業者はいくつかのプリンタのタイプおよびモデルを製造しています。「プリンタモデル」はプリンタを正確に定義します。プリンタモデルは通常、プリンタの前面または上面に刻印されています。また、この情報は梱包材や同梱のマニュアルなどにも示されています。

次に、今回のリリースで利用できるプリンタモデルの例を示します。

- Lexmark Optra E312
- Lexmark Z32
- Lexmark 1000

プリンタドライバ

「プリンタドライバ」で、プリンタで印刷するとき使用するドライバを選択します。プリンタドライバは、指定したプリンタメーカーとプリンタモデル用の PPD ファイルを選択したときに決定されます。

プリンタタイプ

注 - 今回のリリースでは、Solaris プリンタマネージャでプリンタタイプを割り当てることができなくなりました。その代わりに、Solaris プリンタマネージャで新しいプリンタを追加したり既存のプリンタを変更したりする際には、プリンタのメーカー、モデル、およびドライバを選択する必要があります。以前 Solaris プリンタマネージャで割り当てたプリンタ定義を使用する方法については、28 ページの「以前の Solaris プリンタマネージャのプリンタ定義画面に戻す」を参照してください。

「プリンタタイプ」とは、プリンタの種類を表す一般名です。プリンタタイプは、プリンタのさまざまな制御シーケンスが入っている terminfo データベースエントリを識別します。通常、プリンタタイプはメーカーのモデル名からとります。たとえば、DECwriter のプリンタタイプ名は decwriter です。ただし、共通プリンタタイプ ps はこの規則に従いません。ps は、多くの PostScript プリンタモデルのプリンタタイプとして使用されます。

プリンタタイプを指定するには、lpadmin -T コマンドを使用します。

terminfo データベース内のプリンタ名

各プリンタタイプに関する情報は、terminfo データベース (/usr/share/lib/terminfo) に格納されています。この情報には、プリンタの機能と初期制御データが含まれます。インストールするプリンタは、terminfo データベース内のエントリに対応していなければなりません。

```
$ pwd
/usr/share/lib/terminfo
$ ls
1 3 5 7 9 B H P a c e g i k m o q s u w y
2 4 6 8 A G M S b d f h j l n p r t v x z
$
```

各サブディレクトリには、端末またはプリンタに関してコンパイル済みのデータベースエントリが入っています。各エントリは、プリンタまたは端末のタイプの頭文字別に編成されています。たとえば、Epson プリンタがある場合は、/usr/share/lib/terminfo/e ディレクトリ内を探すと、Epson プリンタの特定のモデルが見つかります。

```
$ cd /usr/share/lib/terminfo/e
$ ls
emots          ep2500+high   ep48          ergo4000      exidy2500
```

```

env230          ep2500+low    epson2500      esprit
envision230    ep40          epson2500-80  ethernet
ep2500+basic   ep4000       epson2500-hi  ex3000
ep2500+color   ep4080       epson2500-hi80 exidy
$

```

上記のように、Epson プリンタのエントリがあります。

NEC プリンタがある場合は、`/usr/share/lib/terminfo/n` ディレクトリ内を探すと、使用中の NEC プリンタモデルが見つかります。

```

$ cd /usr/share/lib/terminfo/n
$ ls
ncr7900      ncr7900iv    netronics     network      nuc
ncr7900-na   ncr7901     netty        netx         nuclterm
ncr7900i     nec         netty-Tabs   newhp
ncr7900i-na  net        netty-vi     newhpkeyboard
$

```

上記のように、このディレクトリには、NEC のエントリが含まれています。

プリンタタイプの選択

ローカル PostScript プリンタの場合は、プリンタタイプとして PostScript (PS) または Reverse PostScript (PSR) を使用します。使用するプリンタが PostScript をサポートしていれば、プリンタタイプが `terminfo` データベースに含まれていても、PS または PSR を選択してください。

PostScript プリンタでページの印刷面を上にして印刷すると、文書は逆方向に印刷されます。1 ページ目はスタックの 1 番下になり、最終ページは 1 番上になります。プリンタのタイプを PSR として指定すると、LP 印刷サービスはプリンタに送る前にページの順序を逆転させます。つまり、最終ページが最初に印刷され、各ページは正順にスタックされます。ただし、LP 印刷サービスがページ順を確実に変更できるのは、Adobe® Technical Note #5001 の『*PostScript Language Document Structuring Conventions Specification*』(Adobe Developer Relations の Web サイトで入手可能) に準拠する PostScript ファイルの場合だけです。

プリンタで複数の種類のプリンタをエミュレートできる場合は、`lpadmin -T` コマンドを使用して複数のタイプを割り当てることができます。複数のプリンタタイプを指定すると、LP 印刷サービスは各印刷要求に適したタイプを使用します。

該当する `terminfo` ディレクトリ内でプリンタタイプが見つからないことがあります。プリンタのタイプは、そのプリンタのメーカー名に対応しているとは限りません。たとえば、PostScript プリンタのタイプによっては、メーカーや製品名に固有のエントリの代わりに、PS または PSR エントリ (`/usr/share/lib/terminfo/P` ディレクトリに入っています) を使用できます。

例外的なタイプのプリンタを使用する場合は、さまざまなエントリを試してみなければ、プリンタのモデルに使用できる特定の `terminfo` エントリを判断できないことがあります。できれば、プリンタに使用できるエントリを `terminfo` データベース内

で見つけてください。既存のエントリを使用するほうが、新たにエントリを作成するよりはるかに簡単です。独自のエントリを作成しなければならない場合は、147 ページの「サポートされていないプリンタの terminfo エントリを追加する」を参照してください。役立つヒントが掲載されています。

ファイル内容形式の選択

Solaris 9/04 より前のリリースでは、Solaris プリンタマネージャは次の一覧にあるファイル内容形式を提供しており、ローカルプリンタをインストールまたは変更するときには、この一覧から選択できました。しかし、現在、ファイル内容形式は Solaris プリンタマネージャで割り当てることができなくなりました。今回のリリースでは、Solaris プリンタマネージャでプリンタを追加するときには、PPD ファイルを使用して、プリンタのメーカー、モデル、およびドライバを選択します。このとき、推奨されるファイル内容形式は PostScript です。このファイル内容形式はツールによって設定されます。

場合によっては、Solaris プリンタマネージャを以前の動作に戻すことも可能です。詳細は、28 ページの「以前の Solaris プリンタマネージャのプリンタ定義画面に戻す」を参照してください。

注 - 新しいプリンタを追加するとき、あるいは、既存のプリンタを変更するとき、ファイル内容形式を指定するには、`lpadmin -I` コマンドを使用する必要があります。

印刷フィルタはファイルの内容を、目的のプリンタが受け付けることができる形式に変換します。「ファイル内容形式」は、フィルタを通さずに直接印刷できるファイル内容の形式を LP 印刷サービスに通知します。フィルタなしに印刷するには、必要なフォントをプリンタ上でも利用できなければなりません。フィルタは、その他のファイル形式用に設定して使用します。

ほとんどのプリンタは、以下のファイルタイプを直接印刷することができます。

- プリンタタイプと同じタイプ。たとえば、PostScript プリンタ用であれば PS
- simple タイプ。たとえば、ASCII テキストファイル

ユーザーがファイルの印刷要求を出すときは、`lp -T content-type` コマンドを使用してそのファイルの内容形式を指定することができます。要求を出すときにファイルの内容形式を指定しないと、LP サーバーは要求の最初のファイルを見て内容形式を判定します。ファイルが `^D%` または `%!` で始まっている場合、その要求には PostScript™ データが含まれると見なされます。それ以外の場合、ファイルは simple (ASCII) テキストと見なされます。LP 印刷サービスはファイル内容形式を使用して、ファイル内容をプリンタで処理できる形式に変換するためのフィルタを決めます。

選択結果は LP 印刷サービスが使用する名前に変換されます。次の表は、Solaris プリンタマネージャで選択できるファイル内容形式を示しています。

表 3-2 Solaris プリンタマネージャによるファイル内容形式の選択

ファイル内容形式	LP 印刷サービス名	説明
PostScript	postscript	PostScript ファイルはフィルタを通す必要がない
ASCII	simple	ASCII ファイルはフィルタを通す必要がない
PostScript と ASCII	simple, postscript	PostScript ファイルも ASCII ファイルもフィルタを通す必要がない
なし	" "	プリンタのタイプに一致するもの以外は、すべてのファイルがフィルタを通す必要がある
任意	任意	フィルタは使用されない。プリンタがファイル内容形式を直接処理できなければ、そのファイルは印刷されない

プリンタの機能に最も適合するファイル内容形式を選択してください。PostScript は、Solaris プリンタマネージャのデフォルトの選択で、通常はほとんどそのまま使用できます (PostScript ファイルには、フィルタ処理が不要なことを示します)。

通常使用するプリンタ

この節では、Solaris ソフトウェアで最も一般的に使用されるプリンタのプリンタタイプとファイル内容形式について説明します。掲載されていませんが、ここで説明するプリンタの多くは、simple 内容形式のファイルも直接印刷できます。

PostScript プリンタがある場合は、プリンタタイプ PS または PSR と内容形式 postscript を使用してください。PSR はページの順序を逆転させ、各ページを逆順で印刷してバナーページを最後に印刷します。

表 3-3 は、PostScript 以外の他のプリンタと、各プリンタの構成に使用するプリンタタイプを示しています。これらのプリンタでは、ファイル内容形式は simple です。

注 - Sun では表 3-3 のプリンタをサポートしていませんが、フィルタ処理を行うか、プリンタがファイル内容形式を直接印刷できれば、サポートしていないプリンタを使用できます。以下の製品に不明な点がある場合は、製造元に問い合わせてください。

表 3-3 Sun がフィルタを提供していない PostScript 以外のプリンタ

プリンタ	プリンタタイプ
Daisy	daisy

表 3-3 Sun がフィルタを提供していない PostScript 以外のプリンタ (続き)

プリンタ	プリンタタイプ
Datagraphix	datagraphix
DEC LA100	la100
DEC LN03	ln03
DECwriter	decwriter
Diablo	diablo
	diablo-m8
Epson 2500 系列	epson2500
	epson2500-80
	epson2500-hi
	epson2500-hi80
Hewlett-Packard HPCL printer	hplaser
IBM Proprinter	ibmproprinter

terminfo データベースにないプリンタを設定したい場合は、150 ページの「サポートされていないプリンタの terminfo エントリを追加する方法」を参照してください。

第 4 章

プリンタの設定手順 (手順)

この章では、Solaris プリンタマネージャを使用して、プリンタを設定し、ネットワーク上のシステムからアクセスできるようにする方法を説明します。

Solaris プリンタマネージャを使用したプリンタの設定手順については、57 ページの「印刷の設定 (作業マップ)」を参照してください。

プリンタの概要については、第 2 章を参照してください。

印刷の設定 (作業マップ)

作業	説明	参照先
新しく接続したプリンタの追加 (Solaris プリンタマネージャを使用)	プリンタをシステムに物理的に接続したあとで、Solaris プリンタマネージャを使用してプリンタを印刷できるようにする	64 ページの「Solaris プリンタマネージャを使用して、新しく接続したプリンタを追加する方法」
新しく接続したプリンタの追加 (LP 印刷サービスコマンドを使用)	プリンタをシステムに物理的に接続したあとで、LP 印刷サービスコマンドを使用してプリンタを印刷できるようにする	66 ページの「LP 印刷サービスコマンドを使用して、新しく接続したプリンタを追加する方法」
プリンタへのアクセスの追加	Solaris プリンタマネージャを使用して、印刷クライアントでプリンタへのアクセスを追加する	70 ページの「Solaris プリンタマネージャを使用してプリンタアクセスを追加する方法」

作業	説明	参照先
(省略可能) .printers ファイルの設定	\$HOME/.printers ファイルを使用して、ユーザーが独自のカスタムプリンタの別名を設定できるようにする	71 ページの「.printers ファイルを設定する方法」
新しいネットワークプリンタの追加 (ベンダー提供ツールを使用)	プリンタをネットワークに物理的に接続したあとで、プリンタのベンダーが提供するツールを使用してネットワークプリンタを構成する プリンタをネットワークに物理的に接続したあとで、LP 印刷サービスコマンドを使用してネットワークプリンタを設定する	77 ページの「プリンタベンダー提供のツールを使用して新しいネットワークプリンタを追加する方法」 79 ページの「LP 印刷サービスコマンドを使用して新しいネットワークプリンタを追加する方法」
新しいネットワークプリンタの追加 (Solaris プリンタマネージャを使用)	Solaris プリンタマネージャを使用して、プリンタをネットワークから利用できるようにする	77 ページの「Solaris プリンタマネージャを使用して新しいネットワークプリンタを追加する方法」
新しいネットワークプリンタの追加 (LP 印刷サービスコマンドを使用)	LP 印刷サービスコマンドを使用して、プリンタをネットワークから利用できるようにする	79 ページの「LP 印刷サービスコマンドを使用して新しいネットワークプリンタを追加する方法」
(省略可能) バナーページの抑制	バナーページが印刷されないように制御できる	97 ページの「バナーページを抑制する方法」
(省略可能) 障害警告の設定	Solaris プリンタマネージャよりも具体的な障害警告を設定できる	100 ページの「プリンタの障害警告を設定する方法」
(省略可能) 障害回復の設定	プリンタの障害後の回復方法を設定できる	102 ページの「プリンタの障害回復を設定する方法」
(省略可能) プリンタへのアクセスの制限	Solaris プリンタマネージャは許可リストを設定できるが、プリンタにアクセスするユーザーを制限したい場合は、拒否リストを設定する	104 ページの「プリンタへのユーザーアクセスを制限する方法」

Solaris プリンタマネージャによる印刷の設定

Solaris プリンタマネージャは Java ベースのグラフィカルユーザーインターフェースで、ローカルおよびリモートのプリンタ構成を管理できます。このツールは、ネームサービス環境 (NIS、NIS+、フェデレーテッド・ネーミング・サービスを使用した NIS+ (xfn)、およびファイル) 内で使用できます。このツールを使用するには、スーパーユーザーとしてログインする必要があります。

Solaris プリントマネージャを使用すると、プリンタサーバーを設定したり（「新しいローカルプリンタを追加 (Add New Attached Printer)」または「ネットワークプリンタを追加 (Add a Network Printer)」）、印刷クライアントを設定したりできます（「プリンタへのアクセスを追加 (Add Access to Printer)」）。「ローカルプリンタ」または「接続したプリンタ」とは、プリンタサーバーに物理的にケーブル接続したプリンタのことです。「ネットワークプリンタ」とは、ネットワークに物理的に接続したプリンタのことです。Solaris プリントマネージャを使用すると、ローカルプリンタまたはネットワークプリンタを追加できます。プリンタへのアクセスを追加する（または、リモートアクセスを追加する）とは、印刷クライアント（サーバー以外のすべてのマシン）がプリンタにアクセスできるようにする手順のことです。

Solaris プリントマネージャによるプリンタの設定

次の表はプリンタの属性を示しています。Solaris プリントマネージャを使用してプリンタを設定するときに必要となる情報を判断するのに役立ててください。

プリンタ定義	説明	例	デフォルト設定	必須か省略可能か
プリンタ名	プリンタの名前	laser1	なし	接続したプリンタやネットワークプリンタをインストールし、プリンタへのアクセスを追加するのに必須
プリンタサーバー	プリンタサーバーの名前	venus	なし	接続したプリンタやネットワークプリンタをインストールし、プリンタへのアクセスを追加するのに必須 — このフィールドはツールによって入力される
備考欄	ユーザー定義の文字列	laser printer near breakroom	なし	省略可能
プリンタポート	プリンタが接続されているデバイス	/dev/term/a	/dev/term/a	接続したプリンタをインストールするのに必須
プリンタタイプ	プリンタのタイプ	unknown	PostScript	Solaris 99/04 より前のリリースの場合、接続したプリンタまたはネットワークプリンタをインストールするのに必須

プリンタ定義	説明	例	デフォルト設定	必須か省略可能か
ファイル内容形式	印刷する内容	any	PostScript	Solaris 99/04 より前のリリースの場合、接続したプリンタまたはネットワークプリンタをインストールするのに必須
プリンタメーカー	プリンタのメーカー	Lexmark	なし	接続したプリンタやネットワークプリンタをインストールするのに必須
プリンタモデル	プリンタのモデル	Lexmark Optra E312	なし	接続したプリンタやネットワークプリンタをインストールするのに必須
プリンタドライバ	PPD ファイルを指定することによって使用されるドライバ	Foomatic/PostScript	なし	接続したプリンタやネットワークプリンタをインストールするのに必須
宛先	ネットワークプリンタの宛先名	例については、75 ページの「宛先 (またはネットワークプリンタアクセス) 名の選択」を参照	なし	ネットワークプリンタをインストールするのに必須
プロトコル	プリンタとの通信に使用するプロトコル	TCP	BSD	ネットワークプリンタをインストールするのに必須
プリンタ障害通知	ユーザーに障害を通知する方法	Mail to superuser	Write to superuser	省略可能
デフォルトプリンタ	デフォルトプリンタを識別する	なし	なし	省略可能
バナーを常に印刷	バナーが印刷される	なし	バナーが印刷される	省略可能
ユーザーアクセスリスト	プリンタサーバーで印刷できるユーザーのリスト	rimmer, lister	すべてのユーザーが印刷できる	省略可能

Solaris プリンタマネージャと特殊文字

Solaris プリンタマネージャは、入力画面のさまざまなテキストフィールドへのユーザー入力をチェックします。チェックの方法は2種類あります。一般的に不当な入力と特定のフィールドで不当な入力がないかどうかをチェックします。

Solaris プリンタマネージャは、ヘルプ画面以外の入力で、次の文字を受け付けません。

- シェルのメタ文字。“\\${^&*(){}|!;?:<>”など。ただし、ネットワークプリンタ画面の宛先フィールドではコロン(:)を使用できる
- 複数バイト文字
- #、スペース、タブ。ただし、備考欄フィールドではタブを使用できる

Solaris プリンタマネージャの起動

Solaris プリンタマネージャを起動するには、CDE ワークスペースメニューから「プリンタ管理 (Printer Administrator)」を選択するか、あるいはコマンド行からプリンタマネージャを起動します。詳細については、次の節を参照してください。

▼ Solaris プリンタマネージャを起動する方法

1. 次の前提条件を満たしていることを確認します。Solaris プリンタマネージャを使用するには、次の条件を満たす必要があります。
 - ビットマップディスプレイモニター (Sun ワークステーションに付属の標準ディスプレイモニターなど) があること。Solaris プリンタマネージャは、ビットマップ画面のコンソールを持つシステムでだけ使用できます。
 - CDE 環境のような X Window System を実行しているか、xhost 環境が動作するシステムでリモート表示機能を使用していること。
 - 接続したプリンタやネットワークプリンタをインストールする場合は、プリンタサーバーにスーパーユーザーとしてログインしていること。プリンタへのアクセスを追加する場合は、印刷クライアントにスーパーユーザーとしてログインしていること。
 - LDAP、NIS、NIS+、または NIS+ (xfn) データベースを管理するのに必要なアクセス特権を持っていること。
 - ネームサービスが LDAP の場合は、次の事項が必要になります。
 - プリンタの管理者の識別名 (DN) とディレクトリのパスワード
 - 管理ドメインにサービスを提供する LDAP サーバーの IP アドレスまたは名前。この情報は、通常は自動的に表示されて選択されます。
LDAP を使用したプリンタ情報の管理については、33 ページの「ネームサービススイッチでの印刷サポート」を参照してください。
 - ネームサービスが NIS の場合は、NIS マスターの root パスワードが必要です。

- ネームサービスが NIS+ の場合は、次の手順を実行しなければならない場合があります。

- NIS+ マスターにスーパーユーザーとしてログインします。
- 次に示すようにプリンタテーブルを所有するグループを確認します。

```
# niscat -o printers.org_dir.domain_name.com
.
.
.
Group          : "admin.domain_name.com"
```

- 必要なら、`printers.org_dir.<domain>` ファイルの更新を許可された NIS+ `admin` グループに、Solaris プリンタマネージャを実行するシステムを追加します。

```
# nisgrpadm -a admin.domain_name.com host_name
```

- Solaris プリンタマネージャを実行するシステムにスーパーユーザーとしてログインします。NIS+ の構成によっては、`/usr/bin/keylogin` コマンドの実行が必要な場合があります。詳細については、`keylogin` (1) を参照してください。

- ネームサービスが NIS+ (xfn) の場合は、次の手順を実行しなければならない場合があります。

- NIS+ マスターにスーパーユーザーとしてログインします。
- フェデレーテッド・ネーミングテーブルを所有するグループを確認します。

```
# niscat -o fns.ctx_dir.domain_name.com
.
.
.
Group          : "admin.domain_name.com"
```

- 必要なら、`fns.ctx_dir.<domain>` ファイルの更新を許可された NIS+ `admin` グループに、Solaris プリンタマネージャを実行するシステムを追加します。

```
# nisgrpadm -a admin.domain_name.com host_name
```

- Solaris プリンタマネージャを実行するシステムにスーパーユーザーとしてログインします。NIS+ の構成によっては、`/usr/bin/keylogin` コマンドの実行が必要な場合があります。詳細については、`keylogin` (1) を参照してください。

- SUNWppm パッケージがインストールされていること。

```
# pkginfo | grep SUNWppm
system      SUNWppm      Solaris Print Manager
```

2. 次のいずれかの方法で、Solaris プリンタマネージャを起動します。

- CDE のワークスペースメニューの「ツール (Tools)」オプションから「プリンタ管理 (Printer Administrator)」を選択します。

- CDE フロントパネルから「アプリケーション (Applications)」メニューを選択し、アプリケーション (Application Manager) の「システム管理 (System_Admin)」ウィンドウで「プリンタ管理 (Printer Administrator)」アイコンをクリックします。
- コマンド行で次のコマンドを入力します。

```
# /usr/sadm/admin/bin/printmgr &
```

Solaris プリンタマネージャのメインウィンドウ上に「ネームサービスを選択 (Select Naming Service)」ウィンドウが重なって表示されます。

リモートシステムから Solaris プリンタマネージャを使用したい場合は、次のように行います。

ローカルシステムで `xhost` コマンドを使用し、リモートシステム表示用のアクセスを与えるには、次のように入力します。

```
# xhost +remote-system
```

次に、リモートシステムにログインして、`DISPLAY` 環境変数を設定し、Solaris プリンタマネージャを起動します。

```
# DISPLAY=local-system:display_number
# export DISPLAY
# /usr/sadm/admin/bin/printmgr &
```

注 - CDE メニューやコマンド行から Solaris プリンタマネージャを起動できない場合は、次の確認をしてください。

- a. ローカルシステムやリモートシステムの Xserver プロセスに接続する権限がスーパーユーザー (root) にない可能性があります。その場合は、次のように入力します。

```
$ xhost +hostname
$ su
(Enter the root password)
# /usr/sadm/admin/bin/printmgr &
```

Solaris プリンタマネージャを再起動する前に、ローカルシステムまたはリモートシステムの名前で `hostname` を置き換えます。

- b. ローカルシステムまたはリモートシステムに `SUNWppm` パッケージがインストールされていることを確認します。

```
# pkginfo | grep SUNWppm
```

3. ネットワークで使用されているネームサービスを「ネームサービスを選択 (Select Naming Service)」ウィンドウから選択します。選択肢には `NIS+ (xfn)`、`NIS+`、`NIS`、`files` があります。
4. ドメイン名が正しいことを確認します。
ネームサービスが正常に読み込まれると、Solaris プリンタマネージャのメインウィンドウが表示されます。

プリンタサーバーの設定

接続したプリンタまたはネットワークプリンタをシステムに追加すると、そのプリンタにローカルシステムからアクセスできるようになります。プリンタをインストールするシステムが「プリンタサーバー」になります。

次の節では、Solaris プリンタマネージャで、今回のリリースで拡張されたプリンタサポートを利用して、新しく接続したプリンタを追加する方法について説明します。Solaris プリンタマネージャを使用して、PPD ファイルを指定せずに、新しく接続したプリンタを追加する方法については、28 ページの「以前の Solaris プリンタマネージャのプリンタ定義画面に戻す」を参照してください。

今回のリリースで追加または変更された Solaris プリンタマネージャの機能の概要については、25 ページの「印刷の新機能」を参照してください。

▼ Solaris プリンタマネージャを使用して、新しく接続したプリンタを追加する方法

今回のリリースでは、Solaris プリンタマネージャで割り当てることができるプリンタ定義が変更されました。プリンタ定義「プリンタタイプ (Printer Type)」と「ファイル内容形式 (File Content Type)」は利用できません。その代わりに、Solaris プリンタマネージャを使用して、新しく接続したプリンタを追加したり、既存のプリンタを変更したりする場合には、プリンタ定義「プリンタメーカー (Printer Make)」、「プリンタモデル (Printer Model)」、および「プリンタドライバ (Printer Driver)」を指定する必要があります。

Solaris プリンタマネージャの追加または変更された機能の概要については、25 ページの「印刷の新機能」を参照してください。

1. プリンタサーバーであるシステムを選択します。
2. プリンタをプリンタサーバーに接続し、プリンタの電源を入れます。
ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。
3. プリンタを接続したプリンタサーバー上で **Solaris** プリンタマネージャを起動します。
手順については、61 ページの「Solaris プリンタマネージャを起動する方法」を参照してください。
4. 「プリンタ (Printer)」メニューから「新しいローカルプリンタを設定 (New Attached Printer)」を選択します。
「新しいローカルプリンタを設定 (New Attached Printer)」ウィンドウが表示されます。

5. 必須フィールドに適切な情報を入力します。

フィールドに入力するために情報が必要な場合は、「ヘルプ (Help)」ボタンをクリックします。

たとえば、PPD ファイルを指定して、新しく接続したプリンタを追加するには、次のフィールドを入力します。

プリンタ名:
備考欄:
プリンタポート:
プリンタメーカー:
プリンタモデル:
プリンタドライバ:
プリンタ障害通知:
オプション:
デフォルトプリンタ:
バナーを常に印刷:
ユーザーアクセスリスト:

「プリンタサーバー (Printer Server)」フィールドは Solaris プリンタマネージャによって入力されます。

PPD ファイルを指定せずに、新しく接続したプリンタを追加するには、次のフィールドを入力します。

プリンタ名:
備考欄:
プリンタポート:
プリンタタイプ:
ファイル内容形式:
プリンタ障害通知:
オプション:
デフォルトプリンタ:
バナーを常に印刷:
ユーザーアクセスリスト:

「プリンタサーバー (Printer Server)」フィールドは Solaris プリンタマネージャによって入力されます。

PPD ファイルを指定せずに、新しく接続したプリンタを追加する方法については、28 ページの「[以前の Solaris プリンタマネージャのプリンタ定義画面に戻す](#)」を参照してください。

6. 「了解 (OK)」をクリックします。

7. プリンタがインストールされていることを確認します。確認するには、**Solaris** プリンタマネージャのメインウィンドウに新しいプリンタエントリがあるか調べます。

8. プリンタが要求どおり印刷できることを確認します。

```
$ lp -d printer-name filename
```

9. **Solaris** プリンタマネージャを終了します。

「プリンタマネージャ (Print Manager)」メニューから「終了 (Exit)」を選択します。

Solaris プリンタマネージャの新しい機能が動作していることを確認する

今回のリリースでは、Solaris プリンタマネージャで割り当てることができるプリンタ定義が変更されました。「プリンタタイプ (Printer Type)」と「ファイル内容形式 (File Content Type)」オプションは、「プリンタメーカー (Printer Make)」、「プリンタモデル (Printer Model)」、および「プリンタドライバ (Printer Driver)」画面オプションに変更されました。

Solaris プリンタマネージャを使用して印刷待ち行列を作成または変更するときには、ネームサービスとして「files」オプションを選択します。メニューから「プリンタ (Printer)」、「新しいローカルプリンタを設定 (New Attached Printer)」オプションを選択します。ドロップダウンメニューに「プリンタメーカー (Printer Make)」と「プリンタモデル (Printer Model)」が表示される場合、新しい機能は動作しています。

RIP 機能が動作していることを確認する

RIP を使用すると、PostScript 処理機能を持たないプリンタでも印刷できます。

RIP 機能が動作していることを確認するには、新しい印刷待ち行列を作成する必要があるため、Solaris プリンタマネージャの「新しいローカルプリンタを設定 (New Attached Printer)」画面または「新しいネットワークプリンタを設定 (New Network Printer)」画面から、プリンタのメーカーとモデルの対を 1 つ選択します。そして、新しいプリンタで印刷してみます。印刷ジョブの出力が正しい場合、RIP 機能は動作しています。印刷ジョブの出力が正しくない場合、RIP 機能は動作していません。

▼ LP 印刷サービスコマンドを使用して、新しく接続したプリンタを追加する方法

この手順では、今回のリリースで拡張されたプリンタサポートを利用して、新しく接続したプリンタを追加する方法について説明します。lpadmin コマンドの -n オプションを使用すると、新しく接続したプリンタを追加したり、既存のプリンタを変更するときに、PPD ファイルを指定できます。この手順のあとに、2 つの例を示します。1 番目の例では、PPD ファイルを指定して、新しく接続したプリンタを追加する方法を示します。2 番目の例では、PPD ファイルを指定せずに、新しく接続したプリンタを追加する方法を示します。

1. プリンタをシステムに接続して、プリンタの電源を入れます。
ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。
2. PPD ファイルを指定して接続したプリンタを構成するのに必要な情報を収集します。
 - プリンタ名とポートデバイス

- ファイル内容形式
- PPD ファイル

プリンタで使用する PPD ファイルを決定するには、まず、プリンタのメーカー、モデル、およびドライバを定義します。

LP 印刷サービスコマンドを使用してプリンタを追加または変更する際の、PPD ファイルの使用の概要については、25 ページの「印刷の新機能」を参照してください。

3. このプリンタで使用する、プリンタ名、ポートデバイス、ファイル内容形式、および PPD ファイルを定義します。

PPD ファイルを指定する場合、ファイル内容形式は通常「PostScript」です。

- a. このプリンタで使用する、プリンタ名、ポートデバイスを指定します。

```
# lpadmin -p printer-name -v /dev/printers/0
使用するデバイスは /dev/printers/0 です。
```

- b. このプリンタで使用するインタフェーススクリプトを指定します。

```
# lpadmin -p printer-name -m standard_foomatic
```

- c. このプリンタで使用するファイル内容形式と PPD ファイルを指定します。

```
# lpadmin -p printer-name -I content-type -n
/usr/lib/lp/model/ppd/manufacturer/ppdfile
ここで指定する PPD ファイルは /usr/lib/lp/model/ppd/manufacturer
ディレクトリにあります。
```

- d. プリンタの説明 (備考欄) を指定します。

```
# lpadmin -p printer-name -D "printer-description"
詳細は、lpadmin(1M) のマニュアルページを参照してください。
```

4. (省略可能) フィルタをプリンタサーバーに追加します。

```
# cd /etc/lp/fd
# for filter in *.fd;do
> name=`basename $filter .fd`
> lpfilter -f $name -F $filter
> done
```

5. プリンタが印刷要求を受け入れて、印刷できるようにします。

```
# accept printer-name
# enable printer-name
```

6. プリンタが正しく構成されていることを確認します。

```
# lpstat -p printer-name
lpstat コマンドの出力には、指定した PPD ファイルが表示されます。
```

例—LP 印刷サービスコマンドで、PPD ファイルを指定して、新しく接続したプリンタを追加する

この例では、PPD ファイルを指定して、新しく接続したプリンタを追加する方法を示します。lpadmin コマンドの -n オプションを使用すると、PPD ファイルを指定して、新しい印刷待ち行列を追加できます。この例では次の情報を使用しています。実際に指定する情報とは異なります。

- プリンタ名: paper
- ポートデバイス: /dev/printers/0
- ファイル内容形式: postscript
- PPD ファイル: /usr/lib/lp/model/ppd/Lexmark/Lexmark-Optra_E312-Postscript.ppd.gz

```
# lpadmin -p paper -v /dev/printers/0 1
# lpadmin -p paper -I postscript 2
# lpadmin -p paper -m standard_footmatic 3
# lpadmin -p paper -n /usr/lib/lp/model/ppd/Lexmark/ \
Lexmark-Optra_E312-Postscript.ppd.gzmake4
# lpadmin -p paper -D "Color printer on third floor, rm 3003" 5
# cd /etc/lp/fd
# for filter in *.fd;do
  > name=`basename $filter .fd`
  > lpfilter -f $name -F $filter
  > done 6
# accept paper
  destination "paper" now accepting requests
# enable paper 7
printer "paper" now enabled
# lpstat -p paper 8
printer paper is idle. enabled since Feb 28 11:21 2004. available.
```

1. このプリンタで使用するプリンタ名とポートデバイスを指定する。
2. プリンタが直接印刷できるファイル内容形式を設定する。
3. PPD ファイルを利用するためのインタフェーススクリプトを指定する。
4. PPD ファイルを指定する。
5. プリンタの説明を追加する。
6. 印刷フィルタをプリンタサーバーに追加する。
7. プリンタが印刷要求を受け入れて、印刷できるようにする。
8. プリンタが印刷できる状態にあることを確認する。

例—LP 印刷サービスコマンドで、PPD ファイルを指定せずに、新しく接続したプリンタを追加する

この例では、PPD ファイルを指定せずに、新しいローカルプリンタを追加する方法を示します。この例のコマンドは、プリンタが接続されているプリンタサーバーで実行しなければなりません。この例では次の情報を使用しています。実際に指定する情報とは異なります。

- プリンタ名: luna
- ポートデバイス: /dev/term/b
- プリンタタイプ: PS
- ファイル内容形式: postscript

```
# chown lp /dev/term/b
# chmod 600 /dev/term/b 1
# lpadmin -p luna -v /dev/term/b 2
# lpadmin -p luna -T PS 3
# lpadmin -p luna -I postscript 4
# lpadmin -p luna -D "Room 1954 ps" 5
# cd /etc/lp/fd
# for filter in *.fd;do
  > name='basename $filter .fd'
  > lpfilter -f $name -F $filter
  > done 56
# accept luna
  destination "luna" now accepting requests
# enable luna 7
printer "luna" now enabled
# lpstat -p luna 8
printer luna is idle. enabled since Feb 28 11:21 2003. available.
```

1. lp に所有権とポートデバイスへの単独アクセスを設定する。
2. このプリンタで使用するプリンタ名とポートデバイスを指定する。
3. プリンタタイプを設定する。
4. プリンタが直接印刷できるファイル内容形式を指定する。
5. プリンタの説明を追加する。
6. 印刷フィルタをプリンタサーバーに追加する。
7. プリンタが印刷要求を受け入れて、印刷できるようにする。
8. プリンタが印刷できる状態にあることを確認する。

次に進む手順

次の表を見て、次に進む手順を決めてください。

作業	参照箇所
印刷クライアントに新しくインストールしたプリンタへのアクセスを追加する (プリンタ情報をネームサービスデータベースに追加していない場合)	70 ページの「Solaris プリンタマネージャを使用してプリンタアクセスを追加する方法」
.printers ファイルを設定する	71 ページの「.printers ファイルを設定する方法」

印刷クライアントの設定

印刷クライアントは、プリンタ用のサーバーではないが、プリンタにアクセスできるシステムのことです。印刷クライアントは、プリンタサーバーのサービスを使用して、印刷ジョブのプール、スケジュール、およびフィルタリングを実行します。1つのシステムがあるプリンタのプリンタサーバーになり、同時に他のプリンタの印刷クライアントになることも可能です。

プリンタへのアクセスは、ドメイン全体で有効にすることも、マシン単位で有効にすることもできます。これは、プリンタ情報をネームサービスデータベースに追加するかどうかによって異なります。

次の項では、新しい Solaris プリンタマネージャを使用して、印刷クライアントでプリンタアクセスを追加する方法を説明します。この手順のあとに、lp コマンドを使用してプリンタアクセスを追加する例を示します。

▼ Solaris プリンタマネージャを使用してプリンタアクセスを追加する方法

1. リモートプリンタへのアクセスを追加したいシステム上で **Solaris** プリンタマネージャを起動します。
手順については、61 ページの「Solaris プリンタマネージャを起動する方法」を参照してください。
2. 「プリンタ (Printer)」メニューから「プリンタへのアクセスを追加 (Add Access to Printer)」の順に選択します。
「プリンタへのアクセスを追加 (Add Access to Printer)」ウィンドウが表示されません。
3. ウィンドウに情報を入力します。
フィールドに入力するために情報が必要な場合は、「ヘルプ (Help)」ボタンをクリックします。
4. 「了解 (OK)」をクリックします。
5. プリンタへのアクセスが追加されていることを確認します。確認するには、**Solaris** プリンタマネージャのメインウィンドウに新しいプリンタエントリがあるか調べます。
6. プリンタが要求どおり印刷できることを確認します。

```
$ lp -d printer-name filename
```
7. **Solaris** プリンタマネージャを終了します。

「プリンタマネージャ (Print Manager)」メニューから「終了 (Exit)」を選択します。

例—LP 印刷サービスコマンドを使用してプリンタアクセスを追加する

リモートプリンタで印刷する場合、リモートプリンタにアクセスを追加しなければなりません。次の例では、プリンタサーバー saturn のプリンタ luna にアクセスを構成する方法を示します。システム saturn は、プリンタ luna の印刷クライアントになります。

```
# lpadmin -p luna -s saturn 1
# lpadmin -p luna -D "Room 1954 ps" 2
# lpadmin -d luna 3
# lpstat -p luna 4
```

1. プリンタとプリンタサーバーを指定する。
2. プリンタの説明を追加する。
3. プリンタをシステムのデフォルトのプリンタ宛先として設定する。
4. プリンタが用意できていることを確認する。

.printers ファイルの設定

ユーザーがプリンタ情報をカスタマイズする必要がない場合、ユーザーのホームディレクトリに .printers ファイルを設定する必要はありません。しかし、.printers ファイルは、ユーザーが独自のプリンタ別名を設定できる場所です。別名 default を使用すれば、デフォルトのプリンタを設定できます。また、特殊別名 all を設定すれば、印刷要求を取り消したりプリンタの情報をチェックしたりするときの対象となるプリンタのリストを定義できます。

LP 印刷サービスが .printers ファイルを使用するかどうかは、ネームサービススイッチ (/etc/nsswitch.conf) によって制御されます。デフォルト構成では、印刷サービスは、ユーザーのホームディレクトリでプリンタ構成情報を探してから他のネームサービスを調べます。つまり、ユーザーのプリンタ構成ファイルを設定することによって、ネームサービスの共有情報ではなく、好みのプリンタ情報を使用できます。

.printers ファイルの詳細については、printers(4) のマニュアルページを参照してください。ネームサービススイッチの詳細については、nsswitch.conf(4) のマニュアルページを参照してください。

▼ .printers ファイルを設定する方法

1. スーパーユーザーとしてシステムにログインします。

2. 任意のエディタで、**.printers** ファイルをユーザーのホームディレクトリに作成します。
3. (省略可能) 別名 **default** を設定して、指定のプリンタをデフォルトにします。次の例に示すようなエントリを使用します。

```
_default printer-name
```

4. (省略可能) 別名 **_all** を設定して、印刷要求を取り消したりプリンタの状態をチェックしたりするときの対象となるプリンタを定義します。次の例に示すようなエントリを使用します。

```
_all printer1,printer2,printer3
```

5. そのファイルを **.printers** として保存します。

ネットワークプリンタの追加

「ネットワークプリンタ」とは、ネットワークに直接接続されているハードウェアデバイスです。これは、ネットワークプリンタがプリンタサーバーにケーブルで実際に接続されていなくても、プリンタサーバーからアクセスできることを意味します。ネットワークプリンタは専用のシステム名と IP アドレスを持っています。ネットワークプリンタがプリンタサーバーに接続されていない場合でも、そのプリンタ用のプリンタサーバーを設定しておく必要があります。プリンタサーバーは、ネットワークプリンタの待ち行列化機能と印刷管理機能を提供します。

ネットワークプリンタを追加するときには、LP 印刷サービスコマンドを使用しても、Solaris プリントマネージャを使用してもかまいません。今回のリリースで追加または変更された印刷機能については、25 ページの「印刷の新機能」を参照してください。

ネットワークプリンタは、ベンダー提供の印刷プログラムを必要とする特別なプロトコルを 1 つ以上使用することがあります。ベンダーから提供される印刷プログラムの設定手順は、それぞれ異なることがあります。プリンタにベンダー提供サポートが付いていない場合、ほとんどのデバイスについて Solaris のネットワークプリンタサポートを使用できます。可能な場合は、プリンタのベンダーが提供するソフトウェアを使用します。

ベンダーは、SVR4 プリンタインタフェーススクリプトを提供して標準プリンタインタフェーススクリプトを置き換えている場合があります。その場合、SVR4 インタフェーススクリプトはベンダー提供の印刷プログラムを呼び出して、ジョブをプリンタに送ります。このスクリプトが提供されない場合は、標準インタフェーススクリプトを変更してベンダー提供の印刷プログラムを呼び出す必要があります。この作業は、標準インタフェーススクリプトのプリンタごとのコピーをベンダー提供の印刷プログラムを呼び出すように編集することで実行できます。

ネットワークプリンタ構成で使用する用語を説明します。

- プリンタサーバー – プリンタのジョブをスプールおよびスケジュールするマシン。プリンタサーバーはプリンタが構成されるマシンです。
- プリンタホストデバイス – プリンタホストデバイスは、ベンダーが提供するソフトウェアとハードウェアです。ネットワークに対応していないプリンタにネットワークプリンタの機能をサポートします。プリンタホストデバイスとそのデバイスに接続された1つまたは複数のプリンタの組み合わせは「ネットワークプリンタ」になります。
- プリンタノード – プリンタノードは、物理的なプリンタまたはプリンタホストデバイスのいずれかのことです。ネットワークサポートが物理的なプリンタにあるときは、物理的なプリンタです。ネットワークインタフェースを提供するために外部ボックスを使用しているときは、プリンタホストデバイスです。プリンタノード名は、IPアドレスが与えられているマシン名です。この名前はシステム管理者が選択するもので、デフォルトやベンダーの要件はありません。ノードと同様に、プリンタノード名も一意でなければなりません。
- プリンタ名 – 何らかのプリンタのコマンドを使用する時に、コマンド行に入力する名前。プリンタ名は、システム管理者がプリンタ構成時に選択します。物理的に1台のプリンタに対して、複数のプリンタ名または待ち行列名をつけることができ、それぞれがそのプリンタへのアクセスを提供できます。
- 宛先またはネットワークプリンタのアクセス名 – プリンタのサブシステムがプリンタへのアクセス用に使用するプリンタノードポートの内部名。アクセス名は、プリンタノード名か、プリンタベンダーポート指定付きのプリンタノード名です。プリンタベンダーポート指定は、プリンタベンダーのマニュアルで明示的に定義されています。これはプリンタに固有です。プリンタがプリンタホストデバイスでありプリンタでもある場合、ポート指定は、プリンタホストデバイスのマニュアルに説明されています。書式は次のいずれかです。

printer-node-name

または

printer-node-name:port-designation

- プロトコル – プリンタとのケーブル経由の通信に使用するプロトコル。プリンタベンダーから提供されているマニュアルには、選択するプロトコルについての説明があります。ネットワークプリンタサポートは、BSD プリンタプロトコルと raw TCP の両方を提供します。実装によって、両方を使用するように設定できます。
- タイムアウト (再試行間隔) – タイムアウトはプリンタへの接続試行の間の待機秒数をあらわすシード (seed) 数です。このシード数は、接続の試行間に待機する最小の秒数であり、接続が失敗するごとに増えます。プリンタへの接続が繰り返して失敗すると、ユーザーの介入を要求するメッセージがユーザーに戻されます。接続が成功するか、ジョブの所有者がジョブを取り消すまで、再接続の試行は続けられます。

ネットワークプリンタ用のプリンタベンダー提供のソフトウェア

一般に、ネットワークプリンタには、プリンタベンダーから提供されているソフトウェアが必要です。プリンタにベンダー提供のソフトウェアがある場合、可能ならば、プリンタベンダーのソフトウェアを使用します。プリンタベンダー提供のソフトウェアは、そのプリンタの特性をサポートするように設計されていて、プリンタの能力を最大限に活用します。プリンタのマニュアルをよく読んで、プリンタを LP 印刷システムにインストールおよび構成してください。

Sun のネットワークプリンタのサポート

ネットワークプリンタベンダーがソフトウェアサポートを提供していない場合、Sun が提供するソフトウェアを利用できます。このソフトウェアは、ネットワークプリンタの汎用サポートを提供するもののため、必ずしもプリンタで利用できるすべての機能を使用できません。

ネットワークプリンタを追加するための一般的な説明については、第 4 章を参照してください。次は、Sun 提供のソフトウェアを使用したプリンタの管理を説明します。

ネットワークプリンタサポートの呼び出し

ネットワークプリンタ用のソフトウェアサポートは、次のどちらかのインタフェーススクリプト経由で呼び出されます。

- PPD ファイルを指定する場合 — `netstandard_foomatic`
- PPD ファイルを指定しない場合 — `netstandard`

ネットワークインタフェーススクリプトでネットワークプリンタを構成すると、ネットワークプリンタサポートモジュールが呼び出されます。

次に、PPD ファイルを指定して、プリンタをネットワークサポート用に構成するコマンドを示します。

```
# lpadmin -p printer-name -m netstandard_foomatic
```

次に、PPD ファイルを指定せずに、プリンタをネットワークサポート用に構成するコマンドを示します。

```
# lpadmin -p printer-name -m netstandard
```

プロトコルの選択

印刷サブシステムは、BSD 印刷プロトコルと raw TCP を使用してプリンタと通信します。プリンタのマニュアルには、使用するプロトコルについての情報が提供されています。一般に、プリンタに使用するのは TCP プロトコルです。

プロトコルを選択するには、次のどちらかのコマンドを入力します。

```
# lpadmin -p printer-name -o protocol=bsd
```

```
# lpadmin -p printer-name -o protocol=tcp
```

選択したプロトコルが BSD 印刷プロトコルの場合、さらにコントロールファイルをプリンタに送信する順番を選択できます。一部のプリンタは、コントロールファイルの後にデータファイルという順番を仮定しますが、その逆を仮定するプリンタもあります。詳しくは、プリンタベンダーのマニュアルを参照してください。デフォルトでは、コントロールファイルを先に送信します。

順番を選択するには、次のどちらかのコマンドを入力します。

```
# lpadmin -p printer-name -o bsdctrl=first
```

```
# lpadmin -p printer-name -o bsdctrl=last
```

プリンタノード名の選択

システム管理者はプリンタノード名を選択します。ネットワーク上のノードと同様に、この名前は一意でなければなりません。プリンタノード名は、プリンタの IP アドレスと関連付けられます。

宛先 (またはネットワークプリンタアクセス) 名の選択

印刷サブシステムはプリンタのアクセス情報を必要とします。このサブシステムは、プリンタへのネットワーク接続を行うときに宛先名を使用します。この名前は、システム管理者が `lpadmin` コマンドで印刷サブシステムに提供します。これは、プリンタ構成データベースの一部になります。プリンタアクセス名はプリンタノード名であり、ポート名で修飾される場合もあります。ポート指定はプリンタベンダー間で異なります。ポート指定については、プリンタのマニュアルを参照してください。次に、プリンタアクセス名の書式を示します。

```
printer_node-name [:port-designation]
```

例 1 — ポート指定 (番号) 付き宛先名 (またはネットワークプリンタアクセス名)

TCP の共通ポート指定は 9100 です。プリンタノード名が `pn1` で、プリンタベンダーがそのポートを 9100 と定義している場合、プリンタアクセス名は次のようになります。 `pn1:9100`。この場合にプリンタを構成するには、次のコマンドを使用します。

```
# lpadmin -p printer_name -o dest=pn1:9100
```

例 2 — ポート指定 (名前) 付き宛先名 (またはネットワークプリンタアクセス名)

BSD プロトコルを使用するとき、ポート指定は番号でなく、プリンタベンダーが定義した名前です (例: `xxx_parallel_1`). プリンタノード名が `cardboard` の場合、プリンタアクセス名は `cardboard:xxx_parallel_1` になります。
`cardboard:xxx_parallel_1`. この場合にプリンタを構成するには、次のコマンドを使用します。

```
# lpadmin -p printer-name -o dest=cardboard:xxx_parallel_1
```

例 3 — ポート指定なしの宛先名 (またはネットワークプリンタアクセス名)

ポート指定がなく、プリンタノード名が `newspaper` の場合、プリンタアクセス名はプリンタノード名 `newspaper` になります。 `newspaper`. この場合にプリンタを構成するには、次のコマンドを使用します。

```
# lpadmin -p printer_name -o dest=newspaper
```

タイムアウト値の設定

タイムアウトオプションは、プリンタに接続しようとする試行間で待機する時間 (秒数) を個別に選択するためのものです。ウォームアップ時間が長いプリンタの場合には、タイムアウト値を大きくします。デフォルトは 10 秒です。

タイムアウト値は、印刷プロセスが成功するか失敗するかには影響を与えません。これは、ソフトウェアが初期タイムアウトカウントとして使用するシード値です。失敗が続くと、このカウントは増えます。プリンタへの接続の試行が連続して失敗すると、メッセージがスプーラに送信されます。これによって、ユーザーの介入が必要であることをユーザーに警告します。プリンタの電源が入っていなかったり、用紙がなくなっていたりするときにも、このメッセージが生成される可能性があります。たとえば、プリンタがウォームアップしているときに、このようなメッセージが頻繁に生成されるようであれば、タイムアウト値を増やすことで間違ったメッセージを減らすことができます。

システム管理者は最適なタイムアウト値を探してください。タイムアウト値を設定するには次のコマンドを入力してください。

```
# lpadmin -p printer-name -o timeout=n
```

ネットワークプリンタアクセスの管理

各ネットワークプリンタは、そのプリンタへのアクセスを提供するサーバーを 1 つだけ持っています。この制限によって、サーバーはそのプリンタへのアクセスを管理して、ジョブの一貫性を保つことができます。

ネットワークプリンタのデフォルトデバイスは `/dev/null` です。プリンタに待ち行列が1つしかない場合はこのデバイスで十分です。複数の待ち行列が必要であれば、そのデバイスをファイルに設定します。この設定によって、印刷システムはプリンタへのアクセスを待ち行列間で制限できます。次のコマンドは、デバイスファイルを作成して、ネットワークプリンタデバイスとして構成しています。

```
# touch /path/filename
# chmod 600 /path/filename
# lpadmin -p printer-name -v /path/filename
```

次の例では、`devtreedown` というデバイスファイルを作成しています。

```
# touch /var/tmp/devtreedown
# chmod 600 /var/tmp/devtreedown
# lpadmin -p treedown -v /var/tmp/devtreedown
```

▼ プリンタベンダー提供のツールを使用して新しいネットワークプリンタを追加する方法

1. プリンタをネットワークに接続して、プリンタの電源を入れます。
ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。IPアドレスを取得して、プリンタノード名を選択します。これは、ネットワークにノードを追加することと同じです。
2. プリンタのマニュアルに従って、ネットワークプリンタを追加してください。
プリンタのマニュアルを使用して、ネットワークプリンタを構成してください。手順は、ベンダーやプリンタに固有です。
3. 新しいプリンタへのアクセスをクライアントに追加します。
これでプリンタは追加されました。プリンタへのアクセスをクライアントに作成します。詳細は、70 ページの「[Solaris プリンタマネージャを使用してプリンタアクセスを追加する方法](#)」を参照してください。
4. オプションの作業を完了します。
ネットワークプリンタを設定するときは、オプションの作業がいくつかあります。残りの作業については、57 ページの「[印刷の設定 \(作業マップ\)](#)」を参照してください。

▼ Solaris プリンタマネージャを使用して新しいネットワークプリンタを追加する方法

今回のリリースでは、Solaris プリンタマネージャで割り当てることができるプリンタ定義が変更されました。Solaris プリンタマネージャの追加または変更された機能の概要については、25 ページの「[印刷の新機能](#)」を参照してください。

1. プリンタサーバーであるシステムを選択します。
2. プリンタをネットワークに接続します。プリンタの電源を入れます。
ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。
3. プリンタサーバーで **Solaris** プリンタマネージャを起動します。
手順については、61 ページの「Solaris プリンタマネージャを起動する方法」を参照してください。
4. 「プリンタ (Printer)」メニューから「新しいネットワークプリンタを設定 (New Network Printer)」を選択します。
「新しいネットワークプリンタを設定 (New Network Printer)」ウィンドウが表示されます。
5. 必須フィールドに適切な情報を入力します。
フィールドに入力するために情報が必要な場合は、「ヘルプ (Help)」ボタンをクリックします。
たとえば、PPD ファイルを指定して、新しいネットワークプリンタを追加するには、次のフィールドを入力します。

プリンタ名:
備考欄:
プリンタポート:
プリンタメーカー:
プリンタモデル:
プリンタドライバ:
プリンタ障害通知:
宛先:
プロトコル:
オプション:
デフォルトプリンタ:
バナーを常に印刷:
ユーザーアクセスリスト:

「プリンタサーバー (Printer Server)」フィールドは Solaris プリンタマネージャによって入力されます。

PPD ファイルを指定せずに、新しいネットワークプリンタを追加するには、次のフィールドを入力します。

プリンタ名:
備考欄:
プリンタポート:
プリンタタイプ:
ファイル内容形式:
プリンタ障害通知:
宛先:
プロトコル:
オプション:
デフォルトプリンタ:
バナーを常に印刷:
ユーザーアクセスリスト:

「プリンタサーバー (Printer Server)」フィールドは Solaris プリンタマネージャによって入力されます。

Solaris プリンタマネージャを使用して、PPD ファイルを指定せずに、新しいネットワークプリンタを追加する方法については、28 ページの「以前の Solaris プリンタマネージャのプリンタ定義画面に戻す」を参照してください。

6. 「了解 (OK)」をクリックします。
7. プリンタがインストールされていることを確認します。確認するには、**Solaris** プリンタマネージャのメインウィンドウに新しいプリンタエントリがあるか調べます。
8. プリンタが要求どおり印刷できることを確認します。

```
$ lp -d printer-name filename
```
9. **Solaris** プリンタマネージャを終了します。
「プリンタマネージャ (Print Manager)」メニューから「終了 (Exit)」を選択します。

▼ LP 印刷サービスコマンドを使用して新しいネットワークプリンタを追加する方法

この手順では、今回のリリースで拡張されたプリンタサポートを利用して、新しいネットワークプリンタを追加する方法について説明します。この手順のあとに、2 つの例を示します。1 番目の例では、lp コマンドで、PPD ファイルを指定して、新しいネットワークプリンタを追加する方法を示します。2 番目の例では、lp コマンドで、PPD ファイルを指定せずに、新しいネットワークプリンタを追加する方法を示します。

以下は、ネットワークプリンタサポートソフトウェアを使用してネットワークプリンタを設定するのに必要な手順です。このソフトウェアは、プリンタにベンダー提供のソフトウェアが付属していない場合に使用します。

新しいネットワークプリンタを追加したり既存のネットワークプリンタを変更したりする際の、PPD ファイルの使用の概要については、25 ページの「印刷の新機能」を参照してください。

1. プリンタをネットワークに接続して、プリンタの電源を入れます。
ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。IP アドレスを取得して、プリンタノード名を選択します。この手順は、ネットワークにノードを追加することと同じです。
2. PPD ファイルを指定して新しいネットワークプリンタを構成するのに必要な情報を収集します。
 - プリンタ名とポートデバイス

- ネットワークプリンタアクセス名
- プロトコル
- タイムアウト
- ファイル内容形式
- PPD ファイル

PPD ファイルを決定するには、まず、プリンタのメーカー、モデル、およびドライバを定義します。

詳細については、72 ページの「ネットワークプリンタの追加」を参照してください。

3. このプリンタで使用する、プリンタ名、ポートデバイス、ファイル内容形式、および PPD ファイルを定義します。PPD ファイルを指定する場合、ファイル内容形式は通常「PostScript」です。

- a. このプリンタで使用するプリンタ名とポートデバイスを指定します。

```
# lpadmin -p printer-name -v /dev/null
```

使用するデバイスは /dev/null です。

- b. このプリンタで使用するインタフェーススクリプトを指定します。

```
# lpadmin -p printer-name -m netstandard_foomatic
```

ネットワークプリンタ用のサポートソフトウェアと一緒に提供されるインタフェーススクリプトは /usr/lib/lp/model/netstandard_foomatic です。

- c. このプリンタで使用するファイル内容形式と PPD ファイルを指定します。

```
# lpadmin -p printer-name -I content-type -n
/usr/lib/lp/model/ppd/manufacturer/ppdfile
```

ここで指定する PPD ファイルは /usr/lib/lp/model/ppd/manufacturer/ ディレクトリにあります。

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

- d. プリンタの説明(備考欄)を指定します。

```
# lpadmin -p printer-name -D "printer-description"
```

4. プリンタ宛先、プロトコル、およびタイムアウト値を設定します。

```
# lpadmin -p printer-name -o dest=access-name:port -o protocol=protocol
-o timeout=value
```

-p printer-name

ネットワークプリンタ名を指定する

<code>-o dest=access-name:port</code>	ネットワークプリンタアクセス名と、プリンタのマニュアルに定義されていれば指定されたプリンタベンダーポートに、プリンタ宛先を設定する
<code>-o protocol=protocol</code>	プリンタとケーブル経由で通信するために使用するプロトコルを設定する。BSD と raw TCP の両方をサポートしている
<code>-o timeout=value</code>	プリンタへの接続の試行間で待機する秒数を表す再試行タイムアウト値を設定する

5. (省略可能) フィルタをプリンタサーバーに追加します。

```
# cd /etc/lp/fd
# for filter in *.fd;do
  > name=`basename $filter .fd`
  > lpfilter -f $name -F $filter
> done
```

6. プリンタが印刷要求を受け入れて、その要求を印刷できるようにします。

```
# accept printer-name
# enable printer-name
```

7. プリンタが正しく構成されていることを確認します。

```
# lpstat -p printer-name
```

8. 新しいプリンタへのアクセスをクライアントに追加します。

これでプリンタは追加されました。プリンタへのアクセスをクライアントに作成します。詳細は、70 ページの「印刷クライアントの設定」を参照してください。ネームサービスを使用して印刷クライアントを定義する方法については、33 ページの「ネームサービススイッチでの印刷サポート」を参照してください。

9. オプションの作業を完了します。

プリンタを設定するときは、オプションの作業がいくつかあります。残りの作業については、57 ページの「印刷の設定 (作業マップ)」を参照してください。ネームサービスを使用して印刷クライアントを定義する方法については、33 ページの「ネームサービススイッチでの印刷サポート」を参照してください。

例—LP 印刷サービスコマンドで、PPD ファイルを指定して、新しいネットワークプリンタを追加する

この例では、PPD ファイルを指定して、新しいネットワークプリンタを追加する方法を示します。lpadmin コマンドの -n オプションを使用すると、PPD ファイルを指定して、新しい印刷待ち行列を追加できます。この例のコマンドは、プリンタサーバーで実行する必要があります。この例では次の情報を使用しています。実際の状況に合わせて指定する情報を変更してください。

- プリンタ名: luna1
- デバイス: /dev/null
- インタフェース: /usr/lib/lp/model/netstandard_foomatic
- ネットワークプリンタアクセス名: nimquat:9100
- プロトコル: tcp
- タイムアウト: 5
- ファイル内容形式: postscript
- PPD ファイル: /usr/lib/lp/model/ppd/*manufacturer/ppdfile*

```
# lpadmin -p luna1 -v /dev/null 1
# lpadmin -p luna1 -m netstandard_foomatic 2
# lpadmin -p luna1 -o dest=nimquat:9100 -o protocol=tcp
-o timeout=5 3
# lpadmin -p luna1 -I postscript 4
# lpadmin -p luna1 -n /usr/lib/lp/model/ppd/manufacturer/ppdfile 5
# lpadmin -p luna1 -D "Second floor color printer"6
# cd /etc/lp/fd
# for filter in *.fd;do
    > name=`basename $filter .fd`
    > lpfilter -f $name -F $filter
    > done 7
# accept luna1
    destination "luna1" now accepting requests
# enable luna1 8
printer "luna1" now enabled
# lpstat -p luna1 9
printer luna1 is idle. enabled since Mon Apr 19 15:31:15 2004. available.
```

1. プリンタ名を定義する。デバイスを /dev/null に設定する。
2. ネットワークプリンタ用のインタフェーススクリプトを定義する。
3. 宛先、プロトコル、およびタイムアウトを設定する。
4. プリンタが直接印刷できるファイル内容形式を指定する。
5. 使用する PPD ファイルを指定する。
6. プリンタの説明を追加する。
7. 印刷フィルタをプリンタサーバーに追加する。
8. プリンタが印刷要求を受け入れて、印刷できるようにする。
9. プリンタが印刷できる状態にあることを確認する。

例—LP 印刷サービスコマンドで、PPD ファイルを指定せずに、新しいネットワークプリンタを追加する

この例では、PPD ファイルを指定せずに、新しいネットワークプリンタを追加する方法を示します。これらのコマンドは、プリンタサーバー上で実行する必要があります。この例では次の情報を使用しています。実際の状況に合わせて指定する情報を変更してください。

- プリンタ名: luna1
- デバイス: /dev/null

- インタフェース: /usr/lib/lp/model/netstandard
- ネットワークプリンタアクセス名: nimquat:9100
- プロトコル: tcp
- タイムアウト: 5
- ファイル内容形式: postscript
- プリンタタイプ: PS

```

# lpadmin -p lun1 -v /dev/null 1
# lpadmin -p lun1 -m netstandard 2
# lpadmin -p lun1 -o dest=nimquat:9100 -o protocol=tcp
-o timeout=5 3
# lpadmin -p lun1 -I postscript -T PS 4# lpadmin -p lun1 -D "Room 1954 ps" 5
# cd /etc/lp/fd
# for filter in *.fd;do
    > name=`basename $filter .fd`
    > lpfilter -f $name -F $filter
    > done 6
# accept lun1
    destination "lun1" now accepting requests
# enable lun1 7
printer "lun1" now enabled
# lpstat -p lun1 8
printer lun1 is idle. enabled since Feb 28 11:21 2003. available.

```

1. プリンタ名を定義する。デバイスを /dev/null に設定する。
2. ネットワークプリンタ用のインタフェーススクリプトを定義する。
3. 宛先、プロトコル、およびタイムアウトを設定する。
4. プリンタが直接印刷できるファイル内容形式と、プリンタタイプを指定する。
5. プリンタの説明を追加する。
6. 印刷フィルタをプリンタサーバーに追加する。
7. プリンタが印刷要求を受け入れて、印刷できるようにする。
8. プリンタが印刷できる状態にあることを確認する。

プリンタ構成情報を変換する

この節では、SunOS 5.5.1 リリースまたはその互換バージョンを稼動しているシステムのプリンタ構成情報を変換する方法を説明します。この節では、この情報を印刷クライアントにコピーして印刷クライアントが既存のプリンタにアクセスできるようにする方法も説明します。

注 - 既存のプリンタの台数が少ない場合は、プリンタ構成情報を変換して印刷クライアントに配布するのではなく、Solaris プリンタマネージャか Admintool を使用してプリンタへのアクセスを追加してください。プリンタへのアクセスの追加については、57 ページの「印刷の設定 (作業マップ)」を参照してください。

プリンタ構成情報を変換する (作業マップ)

次の作業マップに、プリンタ構成情報を変換する作業の概要を示します。

作業	説明	参照先
既存のプリンタ構成情報を変換する	<p>SunOS 5.5.1 リリースを実行しているシステムのプリンタ構成情報を変換する - SunOS 5.5.1 リリースを使用しているサイトでは、 /etc/lp/printers ディレクトリのプリンタ構成情報を /etc/printers.conf 構成ファイルに変換する。通常、この作業は1回行うだけでよい</p> <p>lpd ベースのプリンタサーバーを実行するシステムのプリンタ構成情報を変換する - lpd ベースのソフトウェアを使用しているサイトでは、 /etc/printcap ファイルのプリンタ構成情報を /etc/printers.conf 構成ファイルに変換する。通常、この作業は1回行うだけでよい</p>	<p>85 ページの「Solaris 2.5.1 システムのプリンタ情報を変換する方法」</p> <p>85 ページの「lpd ベースのプリンタサーバーを実行するシステムのプリンタ情報を変換する方法」</p>

既存のプリンタ構成情報の変換

既存のプリンタ構成情報は、Solaris 9 またはその互換バージョンをインストールあるいはアップグレードするときに自動的に変換されます。この節では、SunOS 5.5.1 リリースあるいは lpd ベースのプリンタサーバーを実行するシステムのプリンタ構成情報を /etc/printers.conf プリンタ構成ファイルに変換する方法を説明します。変換作業を自動化するために、2つの印刷管理コマンドのいずれかを使用します。

- `conv_lp` コマンドは、SunOS 5.9 システムの /etc/lp/printers ディレクトリの情報を、そのシステムの /etc/printers.conf ファイルのエントリに変換します。手順については、85 ページの「Solaris 2.5.1 システムのプリンタ情報を変換する方法」を参照してください。
- `conv_lpd` コマンドは、lpd ベースのプリンタサーバーを実行するシステムの /etc/printcap 構成ファイルの情報を、/etc/printers.conf ファイルのエントリに変換します。手順については、85 ページの「lpd ベースのプリンタサーバーを実行するシステムのプリンタ情報を変換する方法」を参照してください。

ネームサービスを使用していない場合、サイトにある既存のプリンタを含む、`/etc/printers.conf` のマスターファイルを作成します。次に、このマスターファイルをすべての印刷クライアントにコピーするか、(ネームサービスを使用している場合は) NIS または NIS+ にロードします。新しい印刷クライアントは、最初から、サイトにある既存のプリンタにアクセスできるようにしておくことを推奨します。



注意 - NIS または NIS+ ネームサービスを使用してプリンタ情報を構成している場合、印刷クライアント上の `/etc/printers.conf` ファイルは使用しないでください。印刷クライアントは、最初に `/etc/printers.conf` ファイルを使用してプリンタを検出します。そのときに、`/etc/printers.conf` ファイル内の情報と、NIS マップまたは NIS+ マップ内のプリンタ情報に矛盾がある場合は、予期せぬ結果が生じることがあります。この問題を回避するために、印刷クライアントが NIS ネームサービスまたは NIS+ を使用してプリンタ情報を構成するときは、印刷クライアント上の `/etc/printers.conf` ファイルを削除してください。

▼ Solaris 2.5.1 システムのプリンタ情報を変換する方法

1. Solaris 9 を実行しているシステムに、スーパーユーザーとしてログインします。
2. そのシステムの `/etc/lp/printers` ディレクトリ内のプリンタ構成情報を `/etc/printers.conf` ファイルに変換します。

```
# /usr/lib/print/conv_lp
```

▼ lpd ベースのプリンタサーバーを実行するシステムのプリンタ情報を変換する方法

1. lpd ベースのプリンタサーバーを実行するシステムの `/etc/printcap` ファイルを、Solaris 9 リリースを実行するシステムにコピーします。
2. `/etc/printcap` ファイルのコピー先である Solaris 9 を実行しているシステムに、スーパーユーザーとしてログインします。
3. `/etc/printcap` ファイル内のプリンタ構成情報を `/etc/printers.conf` ファイルに変換します。

```
# /usr/lib/print/conv_lpd
```


第 5 章

プリンタの管理 (手順)

この章では、LP コマンドでプリンタを管理する手順について説明します。

この章で説明する手順は次のとおりです。

- 88 ページの「プリンタとリモートプリンタへのアクセスを削除する方法」
- 91 ページの「プリンタの状態をチェックする方法」
- 92 ページの「印刷スケジューラを停止する方法」
- 93 ページの「印刷スケジューラを再起動する方法」
- 93 ページの「プリンタの説明 (備考欄) を追加する方法」
- 95 ページの「システムのデフォルトプリンタを設定する方法」
- 96 ページの「バナーページを選択可能にする方法」
- 97 ページの「バナーページを抑制する方法」
- 99 ページの「プリンタのクラスを定義する方法」
- 100 ページの「プリンタの障害警告を設定する方法」
- 102 ページの「プリンタの障害回復を設定する方法」
- 104 ページの「プリンタへのユーザーアクセスを制限する方法」
- 106 ページの「印刷要求の状態をチェックする方法」
- 110 ページの「プリンタへの印刷要求を受け付けるまたは拒否する方法」
- 108 ページの「プリンタを使用可能または使用不可にする方法」
- 111 ページの「印刷要求を取り消す方法」
- 112 ページの「特定のユーザーからの印刷要求を取り消す方法」
- 113 ページの「印刷要求を別のプリンタに移動する方法」
- 115 ページの「印刷要求の優先順位を変更する方法」

印刷と LP 印刷サービスの概要については、第 2 章を参照してください。

プリンタと印刷スケジューラの管理

この節では、プリンタと印刷スケジューラを管理するために日常的に行う作業について説明します。

プリンタとプリンタアクセスの削除

プリンタの交換が必要な場合や、プリンタを別の場所に移動したい場合は、プリンタサーバーから物理的に削除する前に、LP 印刷サービスからプリンタ情報を削除しなければなりません。また、プリンタ上の現在の印刷要求がすべて印刷されるか、別のプリンタに移動して印刷されるかを確認する必要があります。

プリンタ情報をプリンタサーバーから削除するだけでなく、印刷クライアントまたはネットワークネームサービスからも削除する必要があります。プリンタサーバーからローカルプリンタを削除する場合は、印刷クライアントまたはネットワークネームサービスからリモートプリンタエントリを削除する必要があります。プリンタを別のプリンタサーバーに移動する場合は、印刷クライアントまたはネットワークネームサービスから古いリモートプリンタエントリを削除する必要があります。そうしてから、リモートプリンタの移動先へのアクセスを追加します。

ローカルとリモートのプリンタの削除方法については、88 ページの「[プリンタとリモートプリンタへのアクセスを削除する方法](#)」を参照してください。Solaris プリンタマネージャを使用して、ローカルプリンタまたはリモートプリンタを削除できます。ただし、Solaris プリンタマネージャでは、待ち行列に入っている印刷要求を別のプリンタに移動できません。

▼ プリンタとリモートプリンタへのアクセスを削除する方法

プリンタをサービスから削除するとき、プリンタアクセスを印刷クライアントから削除するには、次の手順を使用します。そうしてから、プリンタ情報をプリンタサーバーから削除します。

1. 削除したいプリンタへアクセスできる印刷クライアントに、スーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷クライアントからプリンタに関する情報を削除します。

```
print-client# lpadmin -x printer-name
```


-x 指定したプリンタを削除する
プリンタ名 削除したいプリンタ名

指定したプリンタに関する情報が、印刷クライアントの `/etc/lp/printers` ディレクトリから削除されます。

- 印刷クライアントが同じプリンタサーバー上の別のプリンタを使用しない場合は、そのプリンタサーバーに関する情報を印刷クライアントから削除します。

```
print-client# lpsystem -r print-server
```

-r 指定したプリンタサーバーを削除する
print-server 削除したいプリンタサーバー名

プリンタサーバーが、印刷クライアントの `/etc/lp/Systems` ファイルから削除されます。

- プリンタへアクセスできる各印刷クライアント上で、[手順 2](#) から [手順 3](#) までを繰り返します。
- プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
- プリンタ上で印刷要求を停止します。

```
print-server# reject printer-name
```

`reject printer-name` 指定したプリンタの印刷要求を拒否する

この手順を実行すると、プリンタの削除処理中は、そのプリンタの待ち行列に新しい要求が入らなくなります。詳細については、[110 ページ](#)の「[プリンタへの印刷要求を受け付けるまたは拒否する方法](#)」を参照してください。

- プリンタを停止します。

```
print-server# disable printer-name
```

この手順を実行すると、印刷要求の印刷は停止されます。印刷を停止する方法については、[108 ページ](#)の「[プリンタを使用可能または使用不可にする方法](#)」を参照してください。

- 待ち行列に残っている印刷要求がある場合は、別のプリンタに移動します。
印刷要求を別のプリンタに移動する方法については、[113 ページ](#)の「[印刷要求を別のプリンタに移動する方法](#)」を参照してください。
- プリンタをプリンタサーバーから削除します。

```
print-server# lpadmin -x printer-name
```

プリンタの構成情報が、プリンタサーバーの /etc/lp/printers ディレクトリから削除されます。

10. 削除したばかりのプリンタを使用していた印刷クライアントが、まだプリンタサーバー上で別のプリンタを使用中でなければ、その印刷クライアントに関する情報を削除します。

```
print-server# lpsystem -r print-client1[,print-client2 ...]
```

-r	指定したプリンタサーバーを削除する
print-client	プリンタサーバーから削除したい印刷クライアント名。このコマンドで複数の印刷クライアントを指定できる。印刷クライアント名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む

指定した印刷クライアントが、プリンタサーバーの /etc/lp/Systems ファイルから削除されます。

11. プリンタ情報が削除されていることを確認します。

- a. 印刷クライアント上でプリンタ情報が削除されていることを確認します。

```
print-client$ lpstat -p printer-name -l
```

上記のコマンドの出力で、プリンタが存在しないことを示すエラーが表示されます。

- b. プリンタサーバー上でプリンタ情報が削除されていることを確認します。

```
print-server$ lpstat -p printer-name -l
```

上記のコマンドの出力で、プリンタが存在しないことを示すエラーが表示されます。

例 — プリンタとリモートプリンタへのアクセスを削除する

次の例は、印刷クライアント terra とプリンタサーバー jupiter からプリンタ luna を削除する方法を示しています。この例では、プリンタサーバーから印刷クライアント terra を削除する方法も示しています。

```
terra# lpadmin -x luna
Removed "luna".
terra# lpstat -p luna -l
jupiter# lpadmin -x luna
jupiter# lpsystem -r terra
Removed "terra".
jupiter# lpstat -p luna -l
```

プリンタの状態のチェック

多くの日常的なプリンタ管理作業には、LP 印刷サービスや特定のプリンタの状態に関する情報が必要です。たとえば、どのプリンタが使用できるかを判断し、そのプリンタの特性を検査しなければならない場合があります。lpstat コマンドを使用すると、LP 印刷サービスや特定のプリンタに関する状態情報を調べることができます。

▼ プリンタの状態をチェックする方法

1. ネットワーク上の任意のシステムにログインします。

2. プリンタの状態をチェックする

ここでは、最も一般的に使用するオプションだけを掲載してあります。その他のオプションについては、lpstat(1) を参照してください。

```
$ lpstat [-d] [-p printer-name [-D] [-l]] [-t]
```

-d	システムのデフォルトプリンタが表示される
-p printer-name	プリンタが使用可能かアイドル状態か、いつ使用可能または使用不可になったか、および印刷要求を受け付けているかどうかが表示される このコマンドで複数のプリンタ名を指定できる。プリンタ名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む。printer-name を指定しなければ、すべてのプリンタの状態が表示される
-D	指定した printer-name の説明 (備考欄) が表示される
-l	指定した printer-name の特性が表示される
-t	すべてのプリンタの状態、使用可能かどうか、印刷要求を受け付けているかどうかなど、LP 印刷サービスに関する状態情報が表示される

例 — プリンタの状態をチェックする

次の例は、システムのデフォルトプリンタを表示する方法を示しています。

```
$ lpstat -d
system default destination: luna
```

次の例は、プリンタ luna の状態を表示する方法を示しています。

```
$ lpstat -p luna
printer luna is idle. enabled since Jul 12 11:17 2001. available.
```

次の例は、プリンタ asteroid とプリンタ luna の説明 (備考欄) を表示する方法を示しています。

```
$ lpstat -p "asteroid luna" -D
printer asteroid faulted. enabled since Jul 12 11:35 2001. available.
unable to print: paper misfeed jam

Description: Printer by break room
printer luna is idle. enabled since Jul 12 11:36 2001. available.
Description: Printer by server room.
```

次の例は、プリンタ luna の特性を表示する方法を示しています。

```
$ lpstat -p luna -l
printer luna is idle. enabled since Thu Jul 12 15:02:32 ...
  Form mounted:
  Content types: postscript
  Printer types: PS
  Description:
  Connection: direct
  Interface: /usr/lib/lp/model/standard
  After fault: continue
  Users allowed:
    (all)
  Forms allowed:
    (none)
  Banner not required
  Character sets:

  Default pitch:
  Default page size: 80 wide 66 long
  Default port settings:
```

印刷スケジューラの再起動

印刷スケジューラ lpsched は、プリンタサーバー上の印刷要求を処理します。ただし、印刷スケジューラがシステム上で動作を停止したために、印刷要求の受け付けや印刷が停止されることがあります。

次の節では、印刷スケジューラの再起動の方法について説明します。印刷スケジューラが動作を停止するときに印刷要求が印刷中だった場合は、印刷スケジューラを再起動すると、その印刷要求全体が印刷されます。

▼ 印刷スケジューラを停止する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷スケジューラが動作しているかどうかをチェックします。

```
# lpstat -r
```

印刷スケジューラが動作していない場合は、「scheduler is not running」というメッセージが表示されます。

3. 印刷スケジューラが動作している場合は停止します。

```
# /etc/init.d/lp stop
```

▼ 印刷スケジューラを再起動する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷スケジューラが動作しているかどうかをチェックします。

```
# lpstat -r
```

印刷スケジューラが動作していない場合は、「scheduler is not running」というメッセージが表示されます。

3. 印刷スケジューラが動作していない場合は起動します。

```
# /etc/init.d/lp start
```

その他のプリンタ定義の設定とリセット

この節では、プリンタ定義の設定またはリセットの手順を説明します。次のプリンタ定義の一部は、Solaris プリンタマネージャを使用して設定できます。次の手順では、迅速にプリンタ定義を設定またはリセットするための、lp コマンドの使用方法を示しています。

▼ プリンタの説明 (備考欄) を追加する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタの説明 (備考欄) を追加します。

```
# lpadmin -p printer-name -D "comment"
```

-p *printer-name*

説明 (備考欄) を追加するプリンタ名

`-D "comment"` 設置場所や管理担当者など、プリンタの特性を指定する。
シェルが解釈する文字 (*、?、\、!、^ など) は、一重引用
符で囲む

詳細については、`lpadmin(1M)` のマニュアルページを参照してください。
プリンタの説明 (備考欄) はプリンタサーバーの `/etc/lp/printers/printer-
name/comment` ファイルに追加されます。

3. Description 情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタの説明 (備考欄) を追加する

次の例は、プリンタ `luna` の説明 (備考欄) を追加する方法を示しています。

```
# lpadmin -p luna -D "Nathans office"
```

デフォルトプリンタの指定

印刷コマンドを使用するときにプリンタ名を入力しなくてもすむように、ユーザーのデフォルトプリンタを指定できます。あるプリンタをデフォルトとして指定する前に、そのプリンタをシステム上の印刷サービスに認識させなければなりません。次のいずれかを設定すれば、ユーザーのデフォルトプリンタを設定できます。

- `LPDEST` 環境変数
- `PRINTER` 環境変数
- ユーザーの `.PRINTERS` ファイルの `_default` 変数
- システムのデフォルトプリンタ (`lpadmin -d` コマンドまたは Solaris プリンタマネージャを使用)

アプリケーションがプリンタを指定する場合は、システムのデフォルトプリンタを設定したかどうかに関係なく、その出力先が印刷サービスに使用されます。アプリケーションにプリンタの出力先がない場合や、印刷コマンドの使用時にプリンタ名が指定されていない場合は、印刷コマンドはデフォルトプリンタを特定の順序で検索します。表 5-1 は、システムのデフォルトプリンタの検索順序を示しています。

表 5-1 デフォルトプリンタの検索順序

検索順序	<code>/usr/bin/lp</code> コマンドを使用	lpd ベースの互換コマンド (<code>lpr</code> 、 <code>lpq</code> および <code>lprm</code>) を使用
1	<code>LPDEST</code> 変数	<code>PRINTER</code> 変数
2	<code>PRINTER</code> 変数	<code>LPDEST</code> 変数

表 5-1 デフォルトプリンタの検索順序 (続き)

検索順序	/usr/bin/lp コマンドを使用	lpd ベースの互換コマンド (lpr、lpq および lprm) を使用
3	システムのデフォルトプリンタ	システムのデフォルトプリンタ

▼ システムのデフォルトプリンタを設定する方法

1. デフォルトプリンタを設定したいシステムに、スーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. システムのデフォルトプリンタを設定します。

```
# lpadmin -d [printer-name]
```

`-d printer-name` システムのデフォルトプリンタとして割り当てるプリンタ名。 `printer-name` を指定しなければ、システムはデフォルトプリンタなしで設定される

デフォルトプリンタ名がシステムの `/etc/lp/default` ファイルに入力されません。

3. システムのデフォルトプリンタをチェックします。

```
$ lpstat -d
```

例 — システムのデフォルトプリンタを設定する

次の例は、プリンタ `luna` をシステムのデフォルトプリンタとして設定する方法を示しています。これは、`LPDEST` または `PRINTER` 環境変数が設定されていない場合、`luna` がシステムのデフォルトプリンタとして使用されることを意味します。

```
# lpadmin -d luna
# lpstat -d
system default destination: luna
```

バナーページの印刷

バナーページには、印刷要求を出したユーザー、印刷要求 ID、要求の印刷時期が出力されます。また、バナーページには、ユーザーがプリントアウトを識別しやすいように変更可能なタイトルを付けることもできます。

バナーページは、印刷ジョブの所有者を簡単に識別できるようにします。これは、多数のユーザーが同じプリンタにジョブを依頼するときに特に便利です。ただし、バナーページを印刷すると用紙の消費量が増えますが、1 台のプリンタを使用するユーザー

ザーが少ない場合は必要ないことがあります。また場合によっては、バナーページを印刷しない方がよいこともあります。たとえば、プリンタに支払い小切手などの特殊な用紙やフォームが装着されている場合は、バナーページを印刷すると問題が起きることがあります。

デフォルトでは、印刷サービスはバナーページを強制的に印刷します。ただしユーザーは、印刷要求を出すときにバナーページの印刷を抑制するかどうかを選択できます。これは、`lpadmin` コマンドまたは Solaris プリンタマネージャから設定することができます。ユーザーが選択できるようにする場合、ユーザーがバナーページの印刷を抑制するには、`-o banner` オプションを使用する必要があります。

また、バナーページが不要な場合、バナーページの印刷を抑制してまったく印刷されないようにすることもできます。バナーページの印刷は、`lpadmin` コマンドを使用することによって抑制できます。

表 5-2 バナーページの印刷

使用するコマンド	バナーページの印刷	変更
<code>lpadmin -p printer -o banner</code> または <code>lpadmin -p printer -o banner=always</code>	常に行われる	一般ユーザーが <code>lp -o nobanner</code> コマンドを使用すると、要求は印刷されるが <code>nobanner</code> 引数は無視される root または別の権限のユーザーの場合、 <code>nobanner</code> 引数が使用される
<code>lpadmin -p printer -o nobanner</code> <code>lpadmin -p printer -o banner=optional</code>	デフォルトで有効。ただし、 <code>lp -o nobanner</code> コマンドを使えば要求単位で無効にできる	該当しない
<code>lpadmin -p printer -o banner=never</code>	無効	不可

詳細は、97 ページの「バナーページを抑制する方法」を参照してください。

▼ バナーページを選択可能にする方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. バナーページを選択可能にします。

```
# lpadmin -p printer-name -o banner=optional
```



```
-p printer-name          バナーページ印刷を選択可能にするプリンタ名
-o banner=optional      ユーザーが印刷要求を出すときにバナーページなしを指定できるようにする
```

すべての印刷要求でバナーページを強制印刷したい場合は、`-o banner=always` オプションを指定します。

バナーページの設定は、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルに指定します。

3. 次のコマンドの出力には、「**Banner not required**」という行が入っています。

```
§ lpstat -p printer-name -l
```

例 — バナーページを選択可能にする

次の例は、プリンタ `luna` のバナーページを選択可能にする方法を示しています。

```
# lpadmin -p luna -o banner=optional
```

▼ バナーページを抑制する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. バナー印刷を抑制します。

```
lpadmin -p printer-name -o banner=never
```

```
-p printer-name          バナーページ印刷を選択可能にするプリンタ名
-o banner=never          どのような状況でもバナーページ印刷を無効にする
```

バナーページの設定は、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルに指定します。

3. 次のコマンドの出力に **Banner not printed** という行が含まれていることを確認します。

```
§ lpstat -p printer-name -l
```

4. プリンタに印刷要求を送ってバナーページが印刷されないことを確認します。

例 — バナーページ印刷を抑制する

次の例は、プリンタ luna のバナーページを印刷しないようにする方法を示しています。

```
# lpadmin -p luna -o banner=never
```

プリンタクラスの設定

印刷サービスを使用すると、複数のローカルプリンタを1つのクラスにグループ化できます。この作業は、lpadmin -c コマンドを使用しなければ実行できません。

プリンタクラスを設定すると、ユーザーは印刷要求の出力先として(個々のプリンタではなく)そのクラスを指定できます。そのクラスで空いている最初のプリンタが印刷に使用されます。その結果、プリンタはできる限りビジーに保たれるので、応答時間が短縮されます。

印刷サービスに認識されるデフォルトのプリンタクラスはなく、定義したプリンタクラスのみが存在することになります。プリンタクラスを定義するには、次の3つの方法があります。

- プリンタタイプ別。たとえば、PostScript プリンタ
- 場所別。たとえば、5階に設置されたプリンタ
- 作業グループまたは部署別。たとえば、経理部

また、1つのクラスには特定の順序で使用される複数のプリンタを含めることができます。LP印刷サービスでは、常に各プリンタがクラスに追加された順番に従って利用できるプリンタをチェックします。したがって、最初に高速プリンタにアクセスしたい場合は、高速プリンタを低速プリンタよりも先にクラスに追加します。その結果、高速プリンタで最大限の印刷要求が処理されることとなります。低速プリンタは、高速プリンタが使用されているときのバックアッププリンタとして確保されます。

注 - 印刷要求の負荷は、ローカルプリンタのクラス内のプリンタ間でのみ調整されません。

クラス名も、プリンタ名と同様に固有の名前でなければなりません。クラス名は14文字以内の英数字で、下線を使用できます。

プリンタクラスは定義しなくてもかまいません。プリンタクラスを使用するとネットワーク上のユーザーに利点があると判断した場合にのみ、クラスを追加してください。

▼ プリンタのクラスを定義する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタのクラスを定義します。

```
# lpadmin -p printer-name -c printer-class
```

<code>-p printer-name</code>	プリンタのクラスに追加するプリンタ名
<code>-c printer-class</code>	プリンタのクラス名

指定したプリンタが、プリンタサーバーの `/etc/lp/classes/printer-class` ファイル内でそのクラスのリストの最後に追加されます。プリンタクラスが存在しない場合は、作成されます。

3. プリンタがプリンタクラスの中にあることを確認します。

```
$ lpstat -c printer-class
```

例 — プリンタのクラスを定義する

次の例は、プリンタ `luna` をプリンタクラス `roughdrafts` に追加する方法を示しています。

```
# lpadmin -p luna -c roughdrafts
```

障害の通知の設定

事前に選択しておく、印刷サービスはプリンタ障害を検出したときに通知できます。プリンタの障害通知を受け取る方法として、以下のいずれかの方法を、`lpadmin -A` コマンドまたは Solaris プリンタマネージャを使用して選択することができます。

- `root` がログインしている端末にメッセージを書き込む
- `root` に電子メールを送る
- 通知しない

ただし、`lpadmin -A` コマンドを使用すると、選択したプログラムで指定されるメッセージを受信するようにすることもできます。また、`lpadmin -A` コマンドで、すでに知っているエラーに関する通知を選択的に抑制することもできます。

障害通知を配信するプログラムを指定しなければ、障害警告の内容は事前に定義済みのメッセージになります。このメッセージは、プリンタが印刷を停止しており、解決が必要であることを示します。

次の表は、`lpadmin -A` コマンドでプリンタに設定できる警告値を示しています。これらの警告値は、印字ホイール、フロントカートリッジ、フォームについても設定できます。

表 5-3 印刷障害の警告値

-A alert の値	説明
'mail [user-name]'	警告メッセージをプリンタサーバー上の root か lp、またはユーザー名として指定した <i>user-name</i> に電子メールで送信する
'write [user-name]'	警告メッセージをプリンタサーバー上の root か lp のコンソールウィンドウ、またはユーザー名として指定した <i>user-name</i> のコンソールウィンドウに送信する。指定したユーザーが警告メッセージを受け取るには、プリンタサーバーにログインしていなければならない
'command'	警告ごとに <i>command</i> ファイルを実行する。環境変数とカレントディレクトリは保存され、ファイルの実行時に復元される
quiet	障害が解決されるまで警告を停止する。この値は、ユーザー (root または指定したユーザー) が繰り返し警告を受け取るときに使用する
none	警告を送信しない。プリンタの障害警告を指定しない場合は、これがデフォルト値である

▼ プリンタの障害警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタに障害警告を設定します。

```
# lpadmin -p printer-name -A alert [-W minutes]
```

-p <i>printer-name</i>	プリンタ障害の警告を指定するプリンタ名
-A <i>alert</i>	プリンタ障害が起きたときに出される警告の種類を指定する。 <i>alert</i> の有効値については表 5-3を参照してください。有効な値は mail、write、quiet など。
-W <i>minutes</i>	障害警告が出される間隔(分単位)を指定する。このオプションを指定しなければ、警告は一度だけ送信される

障害警告の設定は、プリンタサーバーの `/etc/lp/printers/printer-name/alert.sh` ファイルに入力されます。

3. 次のコマンドの出力から、「**On fault**」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタの障害警告を設定する

次の例は、障害警告をユーザー joe に電子メールで送信し、その後は5分ごとに送信するプリンタ mars の設定方法を示しています。

```
# lpadmin -p mars -A 'mail joe' -W 5
```

次の例は、障害警告をコンソールウィンドウに送信し、その後は10分ごとに送信するプリンタ venus の設定方法を示しています。

```
# lpadmin -p venus -A write -W 10
```

次の例は、プリンタ mercury の障害警告を停止する方法を示しています。

```
# lpadmin -p mercury -A none
```

次の例は、プリンタ venus の障害が解決するまで、障害警告を停止する方法を示しています。

```
# lpadmin -p venus -A quiet
```

プリンタの障害回復の設定

障害通知を送信しないことを選択した場合には、問題を解決するために印刷障害を検出することができます。LP印刷サービスは、障害のあるプリンタを継続して使用しません。プリンタ障害の警告に加えて、印刷要求が必要とするときに、印字ホイール、フォントカートリッジ、およびフォームを取り付けるようにシステム管理者に知らせる警告も設定できます。

lpadmin -F コマンドを使用すると、プリンタ専用の障害回復オプションを定義できます。これは、Solaris プリンタマネージャではできません。

プリンタ障害は、用紙切れやトナーカートリッジの交換が必要であるなど、きわめて単純な場合があります。より重大な問題としては、完全なプリンタ障害や電源障害などがあります。プリンタ障害を解決すると、障害が発生したときに有効だった印刷要求は、次のいずれかの方法で印刷を開始します。

- 印刷を最初から開始する
- 印刷を停止したページの先頭から印刷を再開する
- プリンタを使用可能にした後に、印刷を停止したページの先頭から印刷を再開する

印刷が停止したページの先頭から印刷を継続するには、LP印刷サービスは別の印刷フィルタを必要とします。この印刷フィルタは、デフォルトの印刷フィルタによって設定される制御シーケンスを記録します。プリンタは、これらの制御シーケンスを使用してページ境界を追跡します。指定した印刷フィルタで回復処理を実行できなければ、印刷サービスから通知されます。フィルタの作成方法については、165 ページの「新しい印刷フィルタを作成する方法」を参照してください。

プリンタ障害を解決した直後に印刷を再開したい場合は、enable コマンドを使用してプリンタを使用可能にします。

次の表は、lpadmin -F コマンドでプリンタに設定できる障害回復値を示しています。

表 5-4 プリンタ障害回復の値

-F recover-options の値	説明
beginning	障害回復後に、ファイルの先頭から印刷を再開する
continue	障害回復後に、印刷が停止されたページの先頭から印刷を開始する。この回復オプションには印刷フィルタが必要
wait	障害回復後に、プリンタを使用可能にするまで印刷が停止される。enable コマンドでプリンタを使用可能にすると、印刷は停止されたページの先頭から始まる。この回復オプションには印刷フィルタが必要

▼ プリンタの障害回復を設定する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタの障害回復を設定します。

```
# lpadmin -p printer-name -F recovery-options
```

-p printer-name	障害からの回復方法を指定するプリンタ名
-F recovery-options	beginning、continue、wait の3つの有効な回復オプション。 recovery-options の有効値については、表 5-4を参照してください。

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

障害回復の設定がプリンタサーバーの /etc/lp/printers/printer-name/configuration ファイルに入力されます。

3. 次のコマンドの出力から、「**After fault**」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタの障害回復を設定する

次の例は、印刷が停止したページの先頭から再開させるプリンタ luna の設定方法を示しています。

```
# lpadmin -p luna -F continue
```

プリンタへのユーザーアクセスを制限する

利用できるプリンタの一部またはすべてにアクセスできるユーザーを制限する必要がある場合があります。たとえば、一部のユーザーが高品質プリンタ上で印刷できないようにして経費を抑えることができます。プリンタへのユーザーアクセスを制限するには、プリンタサーバー上で `lpadmin -u` コマンドを使用して「許可」リストと「拒否」リストを作成できます。Solaris プリンタマネージャを使用すると、許可リストのみを作成できます。どちらのリストも作成しなければ、プリンタはそこにアクセスできる全ユーザーが利用できます。

許可リストには、指定したプリンタへのアクセスを許可されるユーザー名が入っています。拒否リストには、指定したプリンタへのアクセスを拒否されるユーザー名が入っています。

許可リストと拒否リストには、次の規則が適用されます。

許可リストと拒否リストの規則	ユーザーアクセスの制限
許可リストも拒否リストも作成しない、または両方のリストが空	そのプリンタには全ユーザーがアクセスできる
許可リストで <code>a11</code> を指定する	そのプリンタには全ユーザーがアクセスできる
拒否リストで <code>a11</code> を指定する	サーバー上の <code>root</code> と <code>lp</code> 以外の全ユーザーのアクセスが拒否される
許可リストにエントリを作成する	拒否リストは無視される。リストに指定されているユーザーだけがプリンタにアクセスできる
拒否リストを作成し、許可リストは作成しないか許可リストを空にする	拒否リストで指定されたユーザーはプリンタにアクセスできない

実際にプリンタへのアクセスを制御しているのはプリンタサーバーなので、許可リストと拒否リストを作成できるのはプリンタサーバー上でだけです。許可リストと拒否リストを作成した場合、プリンタサーバーは、プリンタへのユーザーアクセスを排他的に制御します。

表 5-5 は、プリンタへのユーザーアクセスを制限するために許可リストまたは拒否リストに追加できる値を示しています。

表 5-5 許可リストと拒否リストの値

<i>user-list</i> の値	説明
<i>user</i>	任意のシステム上の特定ユーザー
<i>all</i>	すべてのシステム上の全ユーザー
<i>none</i>	すべてのシステム上の全ユーザーが該当しない
<i>system!user</i>	特定システム上の特定ユーザー
<i>!user</i>	ローカルシステム上の特定ユーザー
<i>all!user</i>	任意のシステム上の特定ユーザー
<i>all!all</i>	すべてのシステム上の全ユーザー
<i>system!all</i>	特定システム上の全ユーザー
<i>!all</i>	ローカルシステム上の全ユーザー

▼ プリンタへのユーザーアクセスを制限する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタへのユーザーアクセスを許可または拒否します。

```
# lpadmin -p printer-name -u allow:user-list [ deny:user-list]
```

<i>-p printer-name</i>	許可または拒否ユーザーアクセスリストを適用するプリンタ名
<i>-u allow:user-list</i>	許可ユーザーアクセスリストに追加するユーザー名。このコマンドで複数のユーザーを指定できる。空白またはコンマを使用して名前を区切る。空白を使用する場合は、名前のリストを引用符で囲む。 <i>user-list</i> の有効な値については、表 5-5 を参照
<i>-u deny:user-list</i>	拒否ユーザーアクセスリストに追加するユーザー名。このコマンドで複数のユーザーを指定できる。空白またはコンマを使用して名前を区切る。空白を使用する場合は、名前のリストを引用符で囲む。 <i>user-list</i> の有効な値については、表 5-5 を参照

指定したユーザーが、プリンタサーバーの次のファイル内でプリンタの許可または拒否リストに追加されます。

```
/etc/lp/printers/printer-name/users.allow  
/etc/lp/printers/printer-name/users.deny
```

注 - 許可リストの *user-list* に *none* を指定した場合、プリンタサーバー用に次のファイルは作成されません。

```
/etc/lp/printers/printer-name/alert.sh  
/etc/lp/printers/printer-name/alert.var  
/etc/lp/printers/printer-name/users.allow  
/etc/lp/printers/printer-name/users.deny
```

3. 次のコマンドの出力から、「**Users allowed**」または「**Users denied**」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタへのユーザーアクセスを制限する

次の例は、*nathan* と *george* にだけプリンタ *luna* へのアクセスを許可する方法を示しています。

```
# lpadmin -p luna -u allow:nathan,george
```

次の例は、*nathan* と *george* にだけプリンタ *asteroid* へのアクセスを拒否する方法を示しています。

```
# lpadmin -p asteroid -u deny:"nathan george"
```

印刷要求の管理

ユーザーが印刷クライアントから印刷要求を出すと、その印刷要求はプリンタサーバー上の待ち行列に追加されてからプリンタに送信されます。印刷要求が待ち行列に入っている間は、クライアントシステムからの要求で取り消したり、その状態情報を取得したりできます。LP コマンドで印刷要求の移動、停止、再開または優先順位の変更などを実行するには、プリンタサーバーにログインする必要があります。これらの動作によって、印刷サービスを絶えずスムーズに機能させることができます。

次の表は、`lp -H` コマンドを使用して印刷要求の優先順位を変更する値を示しています。

表 5-6 印刷要求の優先順位を変更する値

<code>-h change-priority</code> の値	説明
hold	ユーザーが取り消すか、LP 印刷サービスに要求の印刷再開を指示するまで、印刷要求を停止する
resume	停止されていた印刷要求を待ち行列に戻す。印刷要求は、優先順位と待ち行列内での位置に従って印刷される。すでに印刷中の印刷ジョブを停止すると、resume は印刷要求が次に印刷される要求になるように待ち行列の先頭に配置する
immediate	印刷要求を待ち行列の先頭に配置する。要求がすでに印刷中であれば、次の要求をすぐに印刷できるように、印刷中の要求を停止する

▼ 印刷要求の状態をチェックする方法

1. ネットワーク上の任意のシステムにログインします。
2. プリンタと印刷要求の状態をチェックします。
 ここには、最も一般的に使用するオプションだけを掲載してあります。その他の有効なオプションについては、lpstat (1) のマニュアルページを参照してください。

```
$ lpstat -o [list] | -u [user-list]
```

<code>-o list</code>	<p>特定のプリンタの印刷要求の状態が表示される。list には、1 つ以上のプリンタ名、プリンタクラス名、または印刷要求 ID を指定できる。</p> <p>list には、複数のプリンタ名、クラス名、ID を指定できる。値を区切るには空白またはコンマを使用する。空白を使用する場合は、値のリストを引用符で囲む。</p> <p>list を指定しなければ、すべてのプリンタへの印刷要求の状態が表示される</p>
<code>-u user-list</code>	<p>特定のユーザーの印刷要求の状態が表示される。user-list では複数のユーザー名を指定できる。</p> <p>このコマンドで複数のユーザーを指定できる。ユーザー名を区切るには空白またはコンマを使用する。空白を使用する場合は、名前のリストを引用符で囲む。</p> <p>user-list を指定しなければ、すべてのユーザーの印刷要求の状態が表示される</p>

lpstat コマンドを使用して印刷要求の状態をチェックすると、印刷要求ごとに 1 行ずつ表示されます。各行には、左から右に、要求 ID、ユーザー、出力サイズ

(バイト)、要求日時、および要求に関する情報(「being filtered」など)が表示されます。

例 — 印刷要求の状態をチェックする

次の例は、ユーザー fred にはプリンタ luna への印刷要求が 1 つあることを示しています。

```
$ lpstat
luna-1    fred      1261     Jul 12 17:34
```

次の例は、ユーザー paul には待ち行列に入っている印刷要求がないことを示しています。

```
$ lpstat -u paul
```

次の例は、プリンタ moon の待ち行列に入っている印刷要求が 2 つあることを示しています。

```
$ lpstat -o moon
moon-78   root      1024     Jul 14 09:07
moon-79   root      1024     Jul 14 09:08
```

プリンタの印刷の有効化と無効化

enable コマンドと disable コマンドは、プリンタが待ち行列に入っている印刷要求を印刷するか保留するかを制御します。プリンタを使用不可にすると、プリンタは待ち行列内の要求の印刷を保留します。ただし、要求はそのまま待ち行列に追加されたままです。要求が待ち行列に追加されないようにするには、プリンタを設定して印刷要求を拒否させなければなりません。印刷要求の拒否については、109 ページの「印刷要求の受け付けまたは拒否」を参照してください。

Solaris プリンタマネージャを使用してプリンタを追加すると、プリンタは有効になり印刷要求を受け付けます。Solaris プリンタマネージャは、それ以上のプリンタ管理は提供しません。

プリンタが使用不可になっている場合は、使用可能にしなければなりません。この状態は、プリンタ障害が起きると発生することがあります。プリンタを使用可能にすると、印刷サービスがそれ以後に印刷待ち行列の要求を拒否しても、待ち行列が空になるまで、印刷待ち行列からの要求が印刷されます。

次の図は、プリンタが使用不可になったときに印刷要求の処理が中断される様子を示しています。

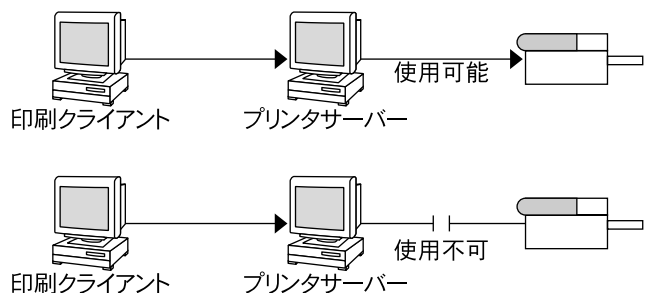


図 5-1 プリンタが使用可能または使用不可になる場合

▼ プリンタを使用可能または使用不可にする方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷要求の印刷を停止します。

```
# disable [-c | -w] [-r "reason "] printer-name
```

<code>disable</code>	現在のジョブを取り消してから、プリンタを使用不可にする。現在のジョブは保存され、プリンタが使用可能になると、印刷が再開される。
<code>-c</code>	現在のジョブを取り消してから、プリンタを使用不可にする。現在のジョブが後から印刷されることはない
<code>-w</code>	現在のジョブが終了するまで待ってからプリンタを使用不可にする
<code>-r "reason"</code>	プリンタが使用不可になった理由をユーザーに示す。理由は格納され、ユーザーがプリンタの状態をチェックする (<code>lpstat -p</code> コマンド) と表示される
プリンタ名	印刷要求の印刷を停止するプリンタ名

注 - プリンタのクラスを使用可能または使用不可にすることはできません。個々のプリンタのみを使用可能または使用不可にすることができます。

3. 印刷要求の印刷を開始します。

```
# enable printer-name
```

4. プリンタが使用可能になっていることを確認します。

```
$ lpstat -p printer-name
```

例 — プリンタを使用可能または使用不可にする

次の例は、プリンタ luna 上の現在のジョブを停止し、後から印刷できるように保存して、プリンタが印刷要求の印刷を停止した理由を表示する方法を示しています。

```
# disable -r "changing the form" luna
```

次の例は、プリンタ luna の印刷要求の印刷を開始する方法を示しています。

```
# enable luna
printer "luna" enabled
```

印刷要求の受け付けまたは拒否

accept コマンドと reject コマンドを使用すると、印刷要求が格納される印刷待ち行列のオンとオフを切り替えることができます。

reject コマンドを使用すると、プリンタの印刷待ち行列がオフになります。新しい印刷要求はプリンタサーバーの待ち行列に入れません。その待ち行列に入っているすべての印刷要求は、そのまま印刷されます。すでに待ち行列に入っている要求の印刷を停止したい場合は、そのプリンタを使用不可にしなければなりません。次の表は、accept、reject、enable、disable の各コマンドの機能を比較したものです。

表 5-7 accept、reject、enable、disable の各コマンドの機能

コマンド	機能
accept	印刷待ち行列に送信された印刷要求を受け付ける
enable	印刷待ち行列にある要求を印刷する
reject	印刷待ち行列に送信された印刷要求を拒否する
disable	現在印刷待ち行列にある印刷要求を停止する

印刷要求が拒否されると、印刷サービスはその要求を出したユーザーにメッセージを送り、指定されたプリンタには印刷要求が受け付けられていないことを通知します。

また、要求を受け付けない理由をコマンド行から指定できます。その理由は、ユーザーがプリンタの待ち行列をチェックしようとするユーザーのシステムに表示されます。次の図は、印刷待ち行列が拒否されたときに印刷要求が中断される様子を示しています。

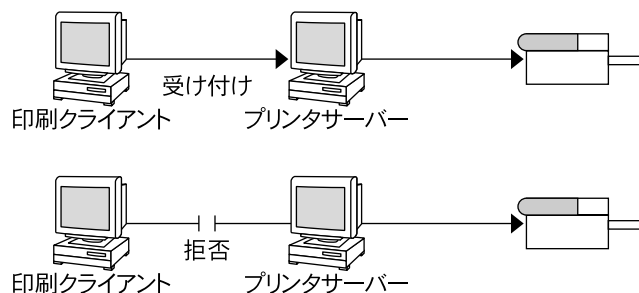


図 5-2 印刷待ち行列が要求を受け付けるか拒否する場合

▼ プリンタへの印刷要求を受け付けるまたは拒否する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタの印刷要求を停止します。

```
# reject [-r "reason"] printer-name
```

`-r "reason"` プリンタが印刷要求を拒否している理由をユーザーに示す。
理由は格納され、ユーザーがプリンタの状態をチェックする
(`lpstat -p`コマンド) と表示される

プリンタ名 印刷要求を停止するプリンタ名

待ち行列に入れられた要求は、プリンタが使用可能になっている限りそのまま印刷されます。印刷を停止するようにプリンタを使用不可にする手順については、[108 ページの「プリンタを使用可能または使用不可にする方法」](#)を参照してください。

3. プリンタの印刷要求を開始します。

```
# accept printer-name
```

4. プリンタの状態が、印刷要求を受け付けているか、拒否しているかを確認します。

```
$ lpstat -p printer-name
```

例 — プリンタへの印刷要求を受け付けるまたは拒否する

次の例は、プリンタ luna の印刷要求の受け入れを停止する方法を示しています。

```
# reject -r "luna is down for repairs" luna
destination "luna" will no longer accept requests
```

次の例は、プリンタ luna が印刷要求を受け入れるように設定する方法を示しています。

```
# accept luna
destination "luna" now accepting requests
```

印刷要求の取り消し

cancel コマンドを使用すると、印刷待ち行列からの印刷要求を取り消したり、印刷中のジョブを取り消すことができます。cancel コマンドには、次の3つの使用方法があります。

- 要求識別番号 (要求 ID) を使用して要求を取り消す。
- すべてまたは指定したプリンタ上で特定のユーザーからの要求を取り消す。
- 現在印刷中のジョブを取り消す。

cancel コマンドを使用すると、要求が取り消され、待ち行列内の次の要求が印刷されることを示すメッセージが表示されます。次の場合にのみ、印刷要求を取り消すことができます。

- 要求を出したユーザーが、要求を出したシステムにログインしている状態で取り消しを実行した場合
- プリンタサーバーの /etc/printers.conf ファイルで user-equivalence オプションが構成されていて、要求を出したユーザーが任意のクライアントシステムから取り消しを実行した場合
- プリンタサーバーにスーパーユーザーまたは lp としてログインしているか、同等の役割になっている場合

特定の要求を取り消すには、その要求 ID を知る必要があります。リクエスト ID は必ず、プリンタ名、ハイフン、印刷要求番号から構成されています。たとえば、luna-185 となります。

印刷要求を依頼すると、その要求 ID が表示されます。印刷要求 ID を忘れた場合は、-o printer オプションを指定して lpstat コマンドを使用すると ID を調べることができます。

▼ 印刷要求を取り消す方法

1. 他のユーザーの印刷要求を取り消すには、スーパーユーザーまたは **lp** になるか、同等の役割になります。
2. 取り消す印刷要求の要求 ID を決めます。

```
# lpstat
```

詳細は、106 ページの「印刷要求の状態をチェックする方法」を参照してください。

3. 印刷要求を取り消します。

```
$ cancel request-id | printer-name
```

<i>request-id</i>	取り消す印刷要求の要求 ID。このコマンドで複数の要求 ID を指定できる。空白またはコンマを使用する。空白を使用する場合は、要求 ID のリストを引用符で囲む
プリンタ名	現在印刷中の印刷要求を取り消したいプリンタを指定する。 このコマンドで複数のプリンタ名を指定できる。プリンタ名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む。

4. 印刷要求が取り消されていることを確認します。

```
$ lpstat -o printer-name
```

例 — 印刷要求を取り消す

次の例は、luna-3 と luna-4 の印刷要求を取り消す方法を示しています。

```
$ cancel luna-3 luna-4  
request "luna-3" cancelled  
request "luna-4" cancelled
```

次の例は、プリンタ luna 上で現在印刷中の印刷要求を取り消す方法を示しています。

```
# cancel luna  
request "luna-9" cancelled
```

▼ 特定のユーザーからの印刷要求を取り消す方法

1. (省略可能) 他のユーザーの印刷要求を取り消すには、スーパーユーザーまたは **lp** になるか、同等の役割になります。
2. 特定のユーザーから印刷要求を取り消します。

```
$ cancel -u user-list [printer-name]
```


<code>-u user-list</code>	<p>特定のユーザーの印刷要求を取り消す。</p> <p><code>user-list</code> では複数のユーザー名を指定できる。ユーザー名を区切るには空白またはコンマを使用する。空白を使用する場合は、名前リストを引用符で囲む。</p>
プリンタ名	<p>指定したユーザーの印刷要求を取り消したいプリンタを指定する</p> <p><code>printer-name</code> では複数のプリンタ名を指定できる。プリンタ名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む。</p> <p><code>printer-name</code> を指定しないと、ユーザーの印刷要求はすべてのプリンタで取り消される</p>

例 — 特定のユーザーからの印刷要求を取り消す

次の例は、プリンタ `luna` 上でユーザー `george` から依頼されたすべての印刷要求を取り消す方法を示しています。

```
# cancel -u george luna
request "luna-23" cancelled
```

次の例は、ユーザー `george` から依頼されたすべての印刷要求をすべてのプリンタ上で取り消す方法を示しています。

```
# cancel -u george
request "asteroid-3" cancelled
request "luna-8" cancelled
```

印刷要求の移動

プリンタの使用方法を変更する計画がある場合や、プリンタの使用を中止する場合は、LP 印刷サービスを設定して新たな印刷要求を拒否しなければなりません。さらに、現在印刷待ち行列に入っている要求があれば、移動するか取り消すかする必要があります。 `lpmove` コマンドを使用すると、個々の印刷要求またはすべての印刷要求を別のローカルプリンタに移動できます。

要求 ID は印刷要求を移動しても変更されないので、ユーザーは引き続き各自の要求を調べることができます。新しく指定したプリンタでは満たせない要件 (ファイル内容形式やフォームなど) を持つ印刷要求は移動できません。この種の印刷要求は取り消さなければなりません。

▼ 印刷要求を別のプリンタに移動する方法

あるプリンタから別のプリンタにすべての要求を移動する場合は、要求 ID がわからなくてもかまいません。ただし移動する前に、影響を受ける印刷要求の数を調べておくといでしょう。

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。

2. (省略可能) 元のプリンタ上で印刷要求の要求 ID をチェックします。

```
# lpstat -o printer-name1
```

3. (省略可能) 宛先プリンタが印刷要求を受け付けているかどうかをチェックします。

```
# lpstat -p printer-name2
```

```
-p printer-name2          印刷要求の移動先となるプリンタ名
```

4. 元のプリンタから宛先プリンタにすべての印刷要求を移動します。

```
# lpmove printer-name1 printer-name2
```

```
printer-name1          すべての印刷要求の移動元となるプリンタ名
```

```
printer-name2          すべての印刷要求の移動先となるプリンタ名
```

詳細は、`lpmove (1M)` のマニュアルページを参照してください。

一部の要求を宛先プリンタ上で印刷できない場合は、元のプリンタの待ち行列内に残ります。要求 ID を使用すると、`lpmove` コマンドで特定の印刷要求を別のプリンタに移動することもできます。

5. 元のプリンタ上で印刷要求の受け付けを開始します。

すべての印刷要求を別のプリンタに移動すると、`lpmove` コマンドはそのプリンタへの印刷要求の受け付けを自動的に停止します。そのプリンタへの新しい印刷要求の受け付けを開始したい場合は、この手順が必要です。

```
# accept printer-name1
```

6. 移動元のプリンタの待ち行列に残っている印刷要求をチェックします。

```
$ lpstat -o printer-name1
```

次のコマンドを使用して、すべての指定した印刷要求が宛先のプリンタ待ち行列に移動していることを確認します。

```
$ lpstat -o printer-name2
```

例 — 印刷要求を別のプリンタに移動する

次の例は、プリンタ `luna` からプリンタ `terra` に印刷要求を移動し、移動元プリンタ `luna` に印刷要求の受付を再開させる方法を示しています。

```
# lpmove luna terra
# accept luna
```

印刷要求の優先順位の変更

印刷要求を出し終わったら、その優先順位をプリンタサーバーの待ち行列内で次のように変更できます。

- 印刷が終了していない場合は、その印刷要求を停止できます。要求を保留状態にして停止すると、現在印刷されている場合は、再開するまで印刷されません。他の印刷要求は、停止中の要求よりも先に印刷されます。
- 任意の印刷要求を待ち行列の先頭に移動できます。その場合、先頭の印刷要求は次に印刷されます。ジョブの印刷を即座に開始したい場合は、現在印刷中のジョブを停止して中断できます。
- 引き続き印刷したいジョブの優先順位を変更して、待ち行列内で優先順位が低い要求の前と、優先順位が同じか高い要求の後の間に移動できます。

▼ 印刷要求の優先順位を変更する方法

1. 印刷要求を保持するプリンタサーバーに、スーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 優先順位を変更したい印刷要求の要求 ID を決めます。

```
# lpstat
```

詳細は、106 ページの「印刷要求の状態をチェックする方法」を参照してください。

3. 印刷要求の優先順位を変更します。

```
# lp -i request-id -H change-priority
```

`-i request-id`

変更したい印刷要求の要求 ID。

このコマンドで複数の要求 ID を指定できる。空白またはコンマを使用する。空白を使用する場合は、要求 ID のリストを引用符で囲む

`-H change-priority`

印刷要求の優先順位を変更する方法として、`hold`、`resume`、`immediate` のいずれかを指定する。

`change-priority` の有効値については、表 5-6 を参照してください。

また、`lp` の `-q` コマンドを使用すると、指定した印刷要求の優先順位を変更できます。優先順位は、最上位の 0 から最下位の 39 までの間で変更できます。

例 — 印刷要求の優先順位を変更する

次の例は、要求 ID が `asteroid-79` の印刷要求を優先順位 1 に変更する方法を示しています。

```
# lp -i asteroid-79 -q 1
```

第 6 章

文字セット、フィルタ、フォーム、 フォントの管理 (手順)

この章では、文字セット、印刷フィルタ、フォーム、およびフォントについての基本事項と、それらを設定して管理する手順について説明します。

この章で説明する手順は次のとおりです。

- 120 ページの「印字ホイールとフォントカートリッジを定義する方法」
- 121 ページの「印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法」
- 123 ページの「印字ホイールまたはフォントカートリッジの装着を促す警告を設定する方法」
- 124 ページの「選択可能文字セットの別名を設定する方法」
- 128 ページの「印刷フィルタを追加する方法」
- 128 ページの「印刷フィルタを削除する方法」
- 129 ページの「印刷フィルタに関する情報を表示する方法」
- 132 ページの「フォームを追加する方法」
- 133 ページの「フォームを削除する方法」
- 134 ページの「フォームを取り外し、装着する方法」
- 135 ページの「フォームの装着に関する警告を設定する方法」
- 137 ページの「フォームに関する情報を表示する方法」
- 137 ページの「フォームの現在の状態を表示する方法」
- 138 ページの「フォームへのユーザーアクセスを制限する方法」
- 139 ページの「フォームへのプリンタアクセスを制限する方法」
- 142 ページの「ダウンロードされた PostScript フォントをインストールする方法」
- 143 ページの「ホスト常駐 PostScript フォントをインストールする方法」

印刷については、第 2 章を参照してください。

文字セットの管理

プリンタでテキストを各種フォント書体で印刷する方法は、それぞれ異なります。たとえば、PostScript プリンタは、テキストをグラフィックスとして処理します。これらのプリンタは、複数のフォントを使用してテキストを生成し、ページ上の任意の位置、サイズ、または方向にテキストを配置できます。その他の形式のプリンタは、印字ホイール、フォントカートリッジ、プログラムされた選択可能な文字セットのいずれかを使用するため、フォントの種類と大きさには制限があります。通常、1つのプリンタ形式には1つの印刷方法が適用されます。

必要に応じてプリンタにフォントを装着する必要があるという点で、LP 印刷サービスでは印字ホイールとフォントカートリッジを同様に扱うことができます。ホイールまたはカートリッジを物理的に装着する必要がある文字セットを、「プリンタに装着する文字セット」といいます。物理的に装着する必要がなく、プリンタにあらかじめプログラムされていて、印刷要求によって選択可能な文字セットを、「プリンタに組み込みの文字セット」といいます。

PostScript 以外のプリンタを設定する場合は、ユーザーが利用可能な、印字ホイールまたは選択可能な文字セットを LP 印刷サービスに指定する必要があります。ユーザーが印刷要求を出すときに、`lp -s` コマンドを使用すると、ジョブの印刷に使用する印字ホイールまたは選択可能な文字セットを指定できます。ユーザーは、すでに定義してある名前でもフォントを参照するだけなので、実際に使用される文字セットの種類を知る必要はありません。たとえば、印刷ホイールを `gothic` と定義したとします。この `gothic` 印刷ホイールを要求するには、次のコマンドを使用します。

```
% lp -s gothic filename
```

選択可能な文字セット

プリンタによってサポートされる選択可能文字セットは、そのプリンタの `terminfo` エントリに表示されています。たとえば、`ln03` プリンタのエントリは、`/usr/share/lib/terminfo/1/ln03` です。 `tput` コマンドを使用して、`terminfo` データベースの任意のプリンタタイプの選択可能文字セットの名前を選択できます。 `tput` コマンドの構文は次のとおりです。

```
tput -T printer-type csn
```

`csn` オプションは文字セット番号 (character set number) の省略形です。番号は、プリンタが初期化された後に常に設定されるデフォルトの文字セット番号である 0 で始まります。その他の文字セット名を表示するには、`-0` の代わりに `-1`、`-2`、`-3` などを使用してコマンドを繰り返してください。選択可能文字セットごとに、`terminfo` 名 (たとえば `usascii`、`english`、`finnish` など) が返されます。

通常、`terminfo` 文字セット名は、プリンタのマニュアルで使用されている文字セット名となるべく一致させてください。同じ文字セット名を使用しないメーカーもあるため、`terminfo` 名はプリンタタイプごとに異なる場合があります。

LP 印刷サービスを使用して選択可能文字セットを登録する必要はありません。ただし、より意味のある名前または別名を与えることができます。

注 - プリンタで使用できる選択可能文字セットを指定しない場合、LP 印刷サービスは、プリンタが任意の文字セット名 (cs0、cs1、cs2 など) またはプリンタが認識する terminfo 名を受け付けることができると仮定します。

lpstat -p -l コマンドを使用して、プリンタサーバーに接続されているプリンタごとに、定義されている選択可能文字セット名を表示できます。

注 - PostScript のフォントは、terminfo データベースのエントリではなく PostScript フィルタによって制御されるため、lpstat -p -l コマンドを使用しても PostScript プリンタ用の文字セットは表示されません。PostScript フォントの管理方法については、140 ページの「フォントの管理」を参照してください。

プリンタに装着する文字セット

別の文字セットを使用するもう一つの方法は、物理的にプリンタに装着できる取り外し可能な印字ホイールまたはフォントカートリッジを使用することです。

プリンタに装着する文字セットを管理するには、LP 印刷サービスに、使用したい印字ホイール名と、プリンタが異なる印字ホイールを必要とするときの警告方法を指定します。次に、ユーザーが lp -s コマンドを使用して特定の文字セットを要求すると、スケジューラは印字ホイールを装着するよう警告を送信し、印刷要求が印刷待ち行列に入れられます。正しい印字ホイールを装着して、印字ホイールを装着したことを LP 印刷サービスに指示すると、ジョブが印刷されます。詳細は、121 ページの「印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法」を参照してください。

1 台のプリンタに対して複数の印字ホイールやカートリッジを指定しなければ、LP 印刷サービスは、プリンタが 1 つの固定印字ホイールまたはカートリッジしか持っておらず、ユーザーはプリンタを使用する際に特殊な印字ホイールやカートリッジを指定できないと見なします。

選択可能文字セットとは違って、印字ホイールまたはカートリッジ用に選択する名前は、terminfo データベースのエントリとは関係がありません。印字ホイール名またはカートリッジ名は、ユーザーが LP 印刷サービスと通信を行うためにだけ使用されます。

ただし、印字ホイールまたはカートリッジ用に選択する名前は、ユーザーがわかりやすいものにしてください。その名前がフォントの書体を表すようにしてください。さらに、その名前は、同じ種類の印字ホイールやカートリッジ、または選択可能文字

セットを持つプリンタの場合には、同じ名前にします。それによって、ユーザーは、どのプリンタ、印字ホイール、カートリッジ、選択可能文字セットを使用するかに関係なく、フォントの書体(文字セット)を指定できます。

システム管理者とプリンタユーザーは、印字ホイールまたはカートリッジに同じ名前を使用してください。 そうしないと、ユーザーが指定する文字セットと管理者が装着するものが異なる可能性があります。

印字ホイールの確認

印字ホイールを確認する手順は、フォームを確認する手順と似ています。一部のプリンタは(通常、文字ベースの印字を行うプリンタ)、特定のフォントや文字セットを提供する印字ホイールや印字カートリッジのような、取り外し可能な印字ヘッドを持っています。ユーザーは名前の付いた文字セットを要求できます。その文字セットがない場合、LP 印刷サービスは要求元または管理者に通知します。印刷ジョブは、印字ホイールが変更されるまで、印刷待ち行列に格納されます。

印字ホイールまたはカートリッジの装着の警告

LP 印刷サービスから出す警告を指定するのと同じ方法で、印字ホイールまたはカートリッジを装着する際に出す警告を指定します。警告の概要については、99 ページの「障害の通知の設定」を参照してください。

▼ 印字ホイールとフォントカートリッジを定義する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタで使用できる印字ホイールまたはフォントカートリッジを定義します。

```
print-server# lpadmin -p printer-name -S hard-charset1[,hard-charset2...]
```

<code>-p printer-name</code>	プリンタで使用できる印字ホイールまたはフォントカートリッジを定義する
------------------------------	------------------------------------

`-s hard-charset`

印字ホイールまたはフォントカートリッジを定義するプリンタ名。

このコマンドで複数のプリンタに装着する文字セット名を指定できる。文字セット名を区切るには空白またはコンマを使用する。空白を使用する場合は、文字セット名のリストを引用符で囲む。

ユーザーにとって意味のある名前を定義して、ユーザーに通知する

印字ホイールまたはフォントカートリッジの定義が、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルに追加されます。

3. プリンタサーバーの印刷クライアントにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
4. 印刷クライアントに対して同じ印字ホイールまたはフォントカートリッジを定義します。

```
print-client# lpadmin -p printer-name -s hard-charset1[,hard-charset2...]
```

このコマンドの変数は、手順 2 と同じです。

印字ホイールまたはフォントカートリッジの定義が、印刷クライアントの `/etc/lp/printers/printer-name/configuration` ファイルに追加されます。

5. 印字ホイールまたはフォントカートリッジを使用する必要がある印刷クライアントごとに、手順 3 と手順 4 を繰り返します。
6. プリンタサーバーと印刷クライアント上で、次のコマンド出力の「**Character sets**」見出しの後にある情報を確認します。

```
$ lpstat -p printer-name -l
```

例 — 印字ホイールを定義する

次の例は、印刷クライアント `asteroid` のプリンタ `luna` 上で印字ホイール `pica` を定義する方法を示しています。

```
asteroid# lpadmin -p luna -s pica
```

▼ 印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタ内の印字ホイールまたはフォントカートリッジを取り外します。

```
# lpadmin -p printer-name -M -S none
```

-p *printer-name* 印字ホイールまたはフォントカートリッジを取り外すプリンタ名

-M -S none 現在の印字ホイールまたはフォントカートリッジを取り外すように指定する

詳細については、lpadmin(1M) のマニュアルページを参照してください。
現在の印字ホイールまたはフォントカートリッジが、プリンタサーバーの
/etc/lp/printers/*printer-name*/configuration ファイルから削除されま
す。

3. 印字ホイールまたはフォントカートリッジをプリンタから削除します。
4. プリンタに新しい印字ホイールまたはフォントカートリッジを入れます。
5. 新しい印刷ホイールまたはフォントカートリッジを装着します。

```
# lpadmin -p printer-name -M -S hard-charset
```

-p *printer-name* 印字ホイールまたはフォントカートリッジを装着するプリンタ名

-M -S *hard-charset* 装着したい印字ホイールまたはフォントカートリッジのプリンタに装着する文字セット名

印字ホイールまたはフォントカートリッジが、プリンタサーバーの
/etc/lp/printers/*printer-name*/configuration ファイルに追加されます。
装着された印字ホイールまたはフォントカートリッジは、取り外されるか、新しい
ものが装着されるまで使用可能です。

6. 次のコマンドの出力の中で、「**Print wheels**」または「**Character set**」の見だしの下にある情報をチェックします。印刷ホイール名または文字セット名と注意「**mounted**」が表示されます。

```
$ lpstat -p printer-name -l
```

例 — 印字ホイールを取り外すまたは装着する

次の例は、プリンタ luna の現在の印刷ホイールを取り外し、pica 印刷ホイールを装着する方法を示しています。

```
# lpadmin -p luna -M -S none  
# lpadmin -p luna -M -S pica
```

▼ 印字ホイールまたはフォントカートリッジの装着を促す警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷ホイールまたはフォントカートリッジの装着を促す警告を設定します。

```
# lpadmin -S hard-charset -A alert [-Q requests] [-W minutes]
```

-S <i>hard-charset</i>	警告を設定したい印字ホイールまたはフォントカートリッジのプリンタに装着する文字セット名
-A <i>alert</i>	印字ホイールまたはフォントカートリッジが要求されたときに 出される警告の種類を指定する。 <i>alert</i> の有効値については、 表 5-3 を参照してください。有効な値は mail、write、 quiet など。 mail または write を指定すると、あらかじめ定義された警告 メッセージが表示される。この警告メッセージは、指定した 印字ホイールまたはフォントカートリッジの装着を促すもの で、それを使用するように設定されている 1 つ以上のプリンタ 名が含まれる
-Q <i>requests</i>	警告が出される前に、印字ホイールまたはフォントカート リッジが、待ち行列に入っていないなければならない印刷要求の 数を指定する。このオプションを指定しなければ、待ち行列に 印刷要求が 1 つ入っただけで警告が出される
-W <i>minutes</i>	警告が出される間隔 (分単位) を指定する。このオプションを 指定しなければ、警告は一度だけ送られる

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

警告は、プリンタサーバーの `/etc/lp/pwheels/charset-name/alert.sh` ファイルに追加されます。

3. 次のコマンドの出力をチェックして、印字ホイールまたはフォントカートリッジの装着を促す警告が追加されているかどうかを確認します。

```
# lpadmin -S hard-charset -A list
```

あるいは、警告を出すために印刷要求に低い番号を設定した場合、最小限の要求を満たすために十分な印刷要求を出し、印字ホイールまたはフォントカートリッジの装着を促す警告を受け取ることを確認します。

例 — 印字ホイールまたはフォントカートリッジの装着を促す警告を設定する

次の例は、印刷待ち行列に elite 印字ホイールに対する 10 の印刷要求があるとき、elite に関して 5 分間隔で電子メールで警告が送られるように設定する方法を示しています。

```
# lpadmin -S elite -A mail -Q 10 -W 5
```

次の例は、印刷待ち行列に finnish フォントカートリッジに対する 5 つの印刷要求があるときに、finnish に関して 1 分間隔で電子メールで警告が送られるように設定する方法を示しています。

```
# lpadmin -S finnish -A mail -Q 5 -W 1
```

次の例は、印刷待ち行列に elite 印字ホイールに対する 5 つの印刷要求があるときに、elite に関して 10 分間隔でコンソールウィンドウに警告が送られるように設定する方法を示しています。

```
# lpadmin -S elite -A write -Q 5 -W 10
```

次の例は、elite 印字ホイールに警告が送られないように設定する方法を示します。

```
# lpadmin -S elite -A none
```

▼ 選択可能文字セットの別名を設定する方法

選択可能文字セットの terminfo 名が正しい場合は、この手順を実行する必要はありません。terminfo データベースの使用については、[147 ページの「サポートされていないプリンタの terminfo エントリを追加する」](#)を参照してください。

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 指定したプリンタタイプの選択可能文字セット名を表示します。

```
# tput -T printer-type csn
```

<code>-T printer-type</code>	terminfo データベースに入っているプリンタタイプ。 terminfo データベースのエントリについては、 51 ページの「プリンタタイプ」 を参照
<code>n</code>	指定したプリンタタイプの選択可能文字セットを表す番号 (0、1、2、3、4、5 など)。プロンプト記号に続いて選択可能 文字セット名が表示される。たとえば、 <code>cs1</code> と指定すると、 <code>english#</code> と表示される

詳細は、tput (1) のマニュアルページを参照してください。

3. 選択可能文字セットの別名を設定します。

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

<code>-p printer-name</code>	選択可能文字セットの別名を設定するプリンタ名
<code>-S select-charset</code>	別名を設定する選択可能文字セット名。この名前は、手順2で検索できる
<code>alias</code>	指定した選択可能文字セットの別名。選択可能文字セット名の他に、この別名を使用できる。 このコマンドで複数の別名を設定できる。別名を区切るには空白またはコンマを使用する。空白を使用する場合は、別名のリストを引用符で囲む

別名は、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルに追加されます。

4. プリンタサーバーの印刷クライアントであるシステムにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。

5. 選択可能文字セットの別名を設定します。

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

このコマンドの変数は、手順3と同じです。

別名は、印刷クライアントの `/etc/lp/printers/printer-name/configuration` ファイルに追加されます。

6. 別名を使用する必要がある印刷クライアントごとに、手順4と手順5を繰り返します。

7. プリンタサーバーと印刷クライアント上で、次のコマンドの出力の中に選択可能文字セットの別名のリストがあることを確認します。

```
$ lpstat -p printer-name -l
```

または、選択可能文字セットに別名を使用する印刷要求を出して、出力をチェックします。

例 — 選択可能文字セットの別名を設定する

次の例は、選択可能文字セット名を表示し、`ln03` プリンタタイプのプリンタ `luna` 上の `usascii` 選択可能文字セットの別名として `text` を指定する方法を示しています。

```
# tput -T ln03 cs0
usascii# tput -T ln03 cs1
english# tput -T ln03 csn2
finnish# tput -T ln03 csn3
japanese# tput -T ln03 cs4
```

```
norwegian#  
# lpadmin -p luna -S usascii=text
```

印刷フィルタの管理

「印刷フィルタ」とは、ファイルの内容形式を出力先プリンタが受け付けられる内容形式に変換するプログラムのことです。LP 印刷サービスはフィルタを使用して、次の機能を提供します。

- ファイルを特定タイプのプリンタで正しく印刷できるように、1つのデータ形式から別のデータ形式に変換する。
- 両面印刷、横方向印刷、ドラフト印刷、または高品質印刷などの特別な印刷モードを処理する。
- プリンタの障害を検出して LP 印刷サービスに通知する。その結果、印刷サービスはユーザーとシステム管理者に警告を出すことができる。

すべての印刷フィルタが上記のすべての機能を実行できるわけではありません。各機能はプリンタに固有なので、別々に実装できます。

LP 印刷サービスは、表 6-1 に示す PostScript フィルタを提供します。これらのフィルタプログラムは、`/usr/lib/lp/postscript` ディレクトリに入っています。通常、PostScript 印刷を行う場合は、プリンタサーバーの設定時にフィルタプログラムをインストールする以外に何も行う必要はありません。Solaris プリンタマネージャが提供されるフィルタを自動的に使用可能にします。ただし、他のプリンタを管理する場合は、それらのプリンタの印刷フィルタを管理する必要がある場合があります。

印刷フィルタの作成

新しい印刷フィルタを作成するには、印刷フィルタプログラムを作成し、印刷フィルタの定義を作成する必要があります。フィルタには、入力形式、出力形式、およびそのフィルタ内でコマンド行引数を処理する言語を提供する複雑なオプションが入っています。説明と手順については、156 ページの「新しい印刷フィルタの作成」を参照してください。

印刷フィルタの追加、変更、削除、および復元

印刷フィルタは、プリンタサーバーだけで追加、変更、または削除できます。

`lpfilter` コマンドを使用すると、利用可能なフィルタを管理できます。フィルタに関するシステム情報は、`/etc/lp/filter.table` ファイルに格納されます。`lpfilter` コマンドは、テーブルに書き出すフィルタに関する情報を、フィルタ記述

子ファイルから取得します。提供されているフィルタ記述子ファイル (PostScript のみ) は、`/etc/lp/fd` ディレクトリに入っています。実際のフィルタプログラムは、`/usr/lib/lp` ディレクトリに入っています。

LP 印刷サービスでは、定義できる印刷フィルタの数に制限はありません。使用しないフィルタを削除して LP 印刷サービスによる処理を減らすことができます。(LP は、特定の印刷要求に使用するフィルタを見つけるために、すべてのフィルタを検査します)。確信が持てない場合は、フィルタを削除しないでください。

フィルタを追加、変更、または削除すると、LP 印刷サービスによって提供されている元のフィルタの一部を上書きしたり、削除したりしてしまう可能性があります。必要に応じて元のフィルタを復元し、追加したフィルタを削除できます。

SunOS ソフトウェアには、PostScript フィルタのデフォルトセットが組み込まれています。デフォルトセットは、Solaris プリンタマネージャによってプリンタサーバーに自動的に追加されます。lpd ベースの印刷サービスで使用されていた TranScript フィルタは、SunOS にも相当するものがある場合とない場合があります。表 6-1 は、デフォルトの PostScript フィルタと、該当する TranScript フィルタが存在する場合はそのフィルタ名を示しています。

表 6-1 デフォルトの PostScript フィルタ

フィルタ (Filter)	動作	相当する TranScript
download	ダウンロードフォント	
dpost	ditroff から PostScript へ	psdit
postdaisy	daisy から PostScript へ	
postdmd	dmd から PostScript へ	
postio	PostScript プリンタへのシリアルインタフェース	pscomm
postior	プリンタとの通信	
postmd	マトリックス型グレースケールから PostScript へ	
postplot	plot から PostScript へ	psplot
postprint	simple から PostScript へ	enscript
postreverse	ページの反転または選択	psrev
posttek	TEK4014 から PostScript へ	ps4014

SunOS ソフトウェアには、次のフィルタは組み込まれていません。

- TeX
- oscat (NeWSprint™ opost)
- Enscript

Enscript の代わりに `postreverse`、`postprint`、`postio`、`dpost` の各フィルタが組み込まれています。

Solaris プリンタマネージャは、プリンタサーバーにデフォルトの PostScript フィルタを追加します。これらのフィルタでは処理できない印刷を行う必要がある場合は、カスタム印刷フィルタの作成方法については、165 ページの「[新しい印刷フィルタを作成する方法](#)」を参照してください。

▼ 印刷フィルタを追加する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. 印刷フィルタの定義に基づく印刷フィルタを追加します。

```
# lpfilter -f filter-name -F filter-def
```

<code>-f filter-name</code>	印刷フィルタ用に選択する名前
<code>-F filter-def</code>	印刷フィルタの定義名

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

印刷フィルタは、プリンタサーバーの `/etc/lp/filter.table` ファイルに追加されます。

3. 次のコマンドの出力の中の印刷フィルタについての情報をチェックして、印刷フィルタが追加されているか確認します。

```
# lpfilter -f filter-name -l
```

例 — 印刷フィルタを追加する

次の例は、`daisytroff.fd` 印刷フィルタ定義を持つ `daisytroff` 印刷フィルタを追加する方法を示しています。

```
# lpfilter -f daisytroff -F /etc/lp/fd/daisytroff.fd
```

▼ 印刷フィルタを削除する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. 印刷フィルタを削除します。

```
# lpfilter -f filter-name -x
```


<code>-f filter-name</code>	削除する印刷フィルタ名
<code>-x</code>	指定したフィルタを削除する

印刷フィルタは、プリンタサーバーの `/etc/lp/filter.table` ファイルから削除されます。

3. フィルタが削除されたことを確認します。

```
# lpfilter -f filter-name -l
指定した名前のフィルタがないというエラーメッセージが表示されます。
```

例 — 印刷フィルタを削除する

次の例は、`daisytroff` 印刷フィルタを削除する方法を示しています。

```
# lpfilter -f daisytroff -x
```

▼ 印刷フィルタに関する情報を表示する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. 印刷フィルタに関する情報を要求します。

```
# lpfilter -f filter-name -l
```

<code>-f filter-name</code>	情報を表示したい印刷フィルタ。利用できるすべての印刷フィルタに関する情報を表示するには、 <code>filter-name</code> に <code>all</code> を指定する
<code>-l</code>	指定したフィルタに関する情報を表示する

指定した1つ以上の印刷フィルタに関する情報が表示されます。

例 — 印刷フィルタに関する情報を表示する

次の例は、`postdaisy` 印刷フィルタに関する情報を要求する方法と、それに応答して表示される情報を示しています。

```
# lpfilter -f postdaisy -l
Input types: daisy
Output types: postscript
Printer types: any
Printers: any
```

```
Filter type: slow
Command: /usr/lib/lp/postscript/postdaisy
Options: PAGES * = -o*
Options: COPIES * = -c*
Options: MODES group = -n2
Options: MODES group\=\([2-9]\) = -n\1
Options: MODES portrait = -pp
Options: MODES landscape = -pl
Options: MODES x\=\(\-*\[\.0-9]*\) = -x\1
Options: MODES y\=\(\-*\[\.0-9]*\) = -y\1
Options: MODES magnify\=\(\[\.0-9]*\) = -m\1
```

次の例は、daisytroff フィルタに関する情報をファイルにリダイレクトする方法を示しています(そのフィルタのフィルタ定義が作成されます)。これは、うっかりフィルタ定義を削除してしまった場合に便利です。

```
# lpfilter -f daisytroff -l > daisytroff.fd
```

次の例は、システムに追加されたすべての印刷フィルタを表示する方法と、それに応答して表示される情報を示しています。

```
# lpfilter -f all -l | grep Filter
(Filter "download")
Filter type: fast
(Filter "postio")
Filter type: fast
(Filter "postior")
Filter type: fast
(Filter "postreverse")
Filter type: slow
```

フォームの管理

「フォーム」は、あらかじめ決められている形式に従って情報が印刷されている用紙です。普通紙と違って、通常、フォームにはテキストまたはグラフィックスが前もって印刷されています。フォームの一般的な例としては、企業のレターヘッド、送り状、小切手、領収書、ラベルなどがあります。

「フォーム」という用語には2つの意味があります。一つは物理的な媒体(用紙)という意味で、もう一つはLP印刷サービスの形式を定義するソフトウェアという意味です。

LP印刷サービスを使用すると、フォームの使用方法を制御できます。この節では、フォームの追加、変更、削除、取り付けを行う方法と、フォームへのアクセスを制御する方法について説明します。

フォームの追加、変更、または削除

フォームを追加するときには、LP印刷サービスに指示を与えて、そのフォームを利用可能なフォームのリストに加えます。また、フォームの記述と定義に必要な情報を与えなければなりません。フォームを追加するとき、その定義を入力できますが、はじめに定義を作成しておいて、ファイルに保存しておくことをお勧めします。ファイルを編集すれば、フォーム定義を変更できます。

注-LP印刷サービスでは、フォーム定義は提供されません。

フォームを変更するには、異なる定義を持つフォームを追加し直さなければなりません。

LP印刷サービスでは、定義できるフォームの数に制限はありません。ただし、不要なフォームは削除してください。不要なフォームがあると、印刷サービスに余計な負担をかける可能性があります。

フォームの装着

フォームを印刷するには、プリンタに給紙し、コマンドを使用してフォームを「装着」し、これによって、プリンタに送られる印刷要求がこのフォーム定義を使用して印刷されることをLP印刷サービスに通知します。複数のフォームを使用する場合など、1台のプリンタで異なる種類の印刷を行う場合には、次の作業を実行します。

- 給紙してフォームを装着する前にプリンタを使用不可にします。
- フォームの準備ができたならプリンタを再び使用可能にします。そうしないと、LP印刷サービスはプリンタでそのフォームを必要としないファイルを印刷し続けます。

フォームを装着するときには、正しく揃っているかどうかを確認してください。揃え方がフォームに対して定義されている場合は、揃え方が正しくなるようにプリンタを調整し終わるまで、フォームを装着した後でパターン印刷を繰り返すように要求できます。

プリンタに取り付けられているフォームの使用を変更または中止したい場合は、フォームを取り外してLP印刷サービスに通知しなければなりません。

フォームの追跡

LP印刷サービスにより、各プリンタにどのフォームが装着されているかを確認できます。また、フォームに印刷するときに必要な記述がなければ、LP印刷サービスが通知します。フォームの記述を作成したり、各プリンタにフォームを装着したり取り外したりするのはシステム管理者の責任です。この作業はプリンタの設定時か、LP印刷サービスからの警告時に行います。

ユーザーは印刷ジョブを印刷したいフォームを指定します。管理者は特定のフォームを装着して、フォームが使用できる状態にあり、どのプリンタに装着されているかを LP 印刷サービスに伝えます。ユーザーは特定のフォームを指定することによって印刷要求を出すことができます。LP 印刷サービスが要求を受け取ると、フォームの装着要求を警告メッセージとして管理者に送信します。

フォームの装着に関する警告の定義

ほかの警告を指定するのと同じ方法で、LP 印刷サービスからのフォームを装着する際に出す警告を指定します。警告の概要については、99 ページの「障害の通知の設定」を参照してください。

フォームのチェック

LP 印刷サービスに対してフォームを定義し終わったら、チェックしたい情報に応じて 2 つのコマンドのどちらかでフォームの定義をチェックできます。

- `lpforms` コマンドを使用してフォームの属性を表示します。また、コマンドの出力をファイルにリダイレクトして将来の参照に備えて保存できます。
- `lpstat` コマンドを使用してフォームの現在の状態を表示します。内容を保護するため、位置揃えパターンは表示されません。

既存のフォーム名がわからない場合は、`/etc/lp/forms` ディレクトリの内容の一覧を表示して調べることができます。

フォームへのアクセスの制限

どのプリンタやユーザーが、ネットワーク上で利用可能な一部またはすべてのフォームを使用できるかを制御できます。たとえば、経理部に属するユーザーだけが小切手のフォームを印刷できるようにしたい場合があります。また、特定のプリンタだけで利用できる小切手のフォームが必要な場合もあります。

フォームへのユーザーアクセスを制限するには、138 ページの「フォームへのユーザーアクセスを制限する方法」、フォームへのプリンタアクセスを制限するには、139 ページの「フォームへのプリンタアクセスを制限する方法」を参照してください。

▼ フォームを追加する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. フォーム定義に基づくフォームを追加します。

```
# lpforms -f form-name -F /etc/lp/forms/form
```

-f form-name	フォーム用に選択した名前
-F /etc/lp/forms/form	フォーム定義名

詳細は、lpforms(1M)のマニュアルページを参照してください。

フォームは、プリンタサーバーの /etc/lp/forms/form-name/describe ファイルに追加されます。

3. 次のコマンドの出力に、フォームについての情報があるかをチェックして、フォームが追加されているか確認します。

```
# lpforms -f form-name -l
```

例 — フォームを追加する

次の例は、medical.fmd フォーム定義を使用する medical フォームを追加する方法を示しています。

```
# lpforms -f medical -F /etc/lp/forms/medical.fmd
```

注 – フォームを使用する前に、そのフォームへのアクセスを1つ以上のプリンタに与えておかなければなりません。詳細は、139 ページの「フォームへのプリンタアクセスを制限する方法」を参照してください。

▼ フォームを削除する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. フォームを削除します。

```
# lpforms -f form-name -x
```

-f form-name	削除するフォーム名
-x	指定したフォームを削除する

詳細は、lpforms(1M)のマニュアルページを参照してください。

フォームは /etc/lp/forms/form-name ファイルから削除されます。

3. フォームが削除されたことを確認します。

```
# lpforms -f form-name -l
```

指定したフォーム名がないことを示すエラーメッセージが表示されます。

例 — フォームを削除する

次の例は、medical フォームを削除する方法を示しています。

```
# lpforms -f medical -x
```

▼ フォームを取り外し、装着する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 現在のフォームを取り外そうとしているプリンタで印刷要求の受け入れを停止します。

```
# reject printer-name
```

プリンタ名 フォームを取り外すプリンタ名

新しい印刷要求 (フォームを必要としない場合もある) は、そのプリンタの待ち行列に入れられなくなります。

3. 現在のフォームを取り外します。

```
# lpadmin -p printer-name -M -f none
```

このコマンドの変数 *printer-name* は、[手順 2](#) と同じです。

現在のフォームは、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルから削除されます。

4. プリンタからフォーム用紙を取り外します。
5. 次の印刷要求のためにフォーム用紙を装着します。
6. フォームを装着します。

```
# lpadmin -p printer-name -M -f form-name [-a -o filebreak]
```

-p *printer-name* フォームを装着するプリンタ名

-M -f *form-name* 装着するフォーム名

```
-a -o filebreak
```

フォームに位置揃えパターンが定義されている場合は、そのコピーを印刷できるようにする

指定したフォームは、プリンタサーバーの `/etc/lp/printers/printer-name/configuration` ファイルに追加されます。

7. プリンタ上で印刷要求の受け付けを開始します。

```
# accept printer-name
```

これで、プリンタは新しく装着したフォームで印刷する準備ができました。

8. 次のコマンド出力の「**Form mounted**」見出しの下にあるフォーム名をチェックし、フォームが装着されていることを確認します。

```
$ lpstat -p printer-name -l
```

あるいは、新しいフォームを必要とする印刷要求を出して、プリンタの出力をチェックします。

例 — フォームを取り外し、装着する

次の例は、現在装着されているフォームをプリンタ luna から取り外すプロセスを示しています。

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f none
# accept luna
destination "luna" now accepting requests
```

次の例は、medical フォームをプリンタ luna 上に装着するプロセスを示しています。

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f medical -a -o filebreak
# accept luna
destination "luna" now accepting requests
```

▼ フォームの装着に関する警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. フォームの装着に関する要求警告を設定します。

```
# lpforms -f form-name -A alert [-Q requests] [-W minutes]
```

<code>-f form-name</code>	要求警告を設定したいフォーム名
<code>-A alert</code>	フォームが要求されるときに出す警告の種類を指定する。 <i>alert</i> に有効な値については、表 5-3 を参照。有効な値は <i>mail</i> 、 <i>write</i> 、 <i>quiet</i> など。 <i>mail</i> または <i>write</i> を選択すると、あらかじめ定義された警告メッセージが表示される。この警告メッセージは、指定されたフォームの装着を促すもので、そのフォームを使用するように設定されている 1 つ以上の複数のプリンタ名が含まれる
<code>-Q requests</code>	警告が出される前に、フォームが必要な印刷要求がいくつ待ち行列に入っていないかならなければならぬかを指定する。このオプションを指定しなければ、印刷要求が待ち行列に 1 つ入っただけで警告が出される
<code>-w minutes</code>	警告が出される間隔 (分単位) を指定する。このオプションを指定しなければ、警告は一度だけ送信される

要求警告は、プリンタサーバーの `/etc/lp/forms/form-name/alert.sh` ファイルに追加されます。

- 次のコマンドの出力をチェックして、そのフォームに関する警告が追加されていることを確認します。

```
# lpforms -f form-name -A list
```

あるいは、警告を出すために印刷要求の低い番号を設定した場合、最小限の要求を満たすために十分な印刷要求を出し、フォームの装着を促す警告を受け取ることを確認します。

例 — フォームの装着に関する警告を設定する

次の例は、印刷待ち行列に *letterhead* フォームに関する 10 の印刷要求があるときに、*letterhead* に関して 5 分ごとに電子メールで警告が送られるように設定する方法を示します。

```
# lpforms -f letterhead -A mail -Q 10 -W 5
```

次の例は、印刷待ち行列に *letterhead* フォームに関する 5 の印刷要求があるときに、*letterhead* に関して 10 分ごとにコンソールウィンドウに警告が送られるように設定する方法を示します。

```
# lpforms -f letterhead -A write -Q 5 -W 10
```

次の例は、*invoice* フォームに関して要求警告が送られないように設定する方法を示しています。

```
# lpforms -f invoice -A none
```


▼ フォームに関する情報を表示する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. フォームに関する情報を要求します。

```
# lpforms -f form-name -l
```

`-f form-name` 情報を表示したいフォーム名。利用できるすべてのフォームに関する情報を表示するには、`form-name` に `all` を指定する

`-l` 指定したフォームを表示する

指定した1つ以上のフォームに関する情報が表示されます。

例 — フォームに関する情報を表示する

次の例は、`medical` フォームに関する情報を表示する方法を示しています。

```
# lpforms -f medical -l
Page length: 62
Page width: 72
Number of pages: 2
Line pitch: 6
Character pitch: 12
Character set choice: pica
Ribbon color: black
Comment:
Medical claim form
```

次の例は、`medical` フォームに関する情報をファイルにリダイレクトするコマンドを示しています。(このコマンドは、そのフォームのフォーム定義を作成します)。これは、うっかりフォーム定義を削除してしまった場合に便利です。

```
# lpforms -f medical -l > medical.fmd
```

▼ フォームの現在の状態を表示する方法

1. プリンタサーバーにログインします。
2. フォームの現在の状態に関する情報を要求します。

```
$ lpstat -f form-name
```

`-f form-name` 現在の状態を表示したいフォーム名。すべてのフォームの現在の状態を表示したい場合は、`form-name` に `all` を指定する

指定した 1 つ以上のフォームの現在の状態に関する情報が表示されます。

例 — フォームの現在の状態を表示する

次の例は、`medical` フォームの状態を表示する方法を示しています。

```
$ lpstat -f medical
form medical is available to you
```

▼ フォームへのユーザーアクセスを制限する方法

1. プリントサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. フォームへのユーザーアクセスを許可または拒否します。

```
# lpforms -f form-name -u allow:user-list | deny:user-list
```

`-f form-name` ユーザーアクセスの許可または拒否リストを作成するためのフォーム名

`-u allow:user-list` ユーザーアクセスの許可リストに追加するユーザー名。複数のユーザーログイン名を指定する場合は、空白またはコンマで区切る。空白で区切る場合は、ID のリストを引用符で囲む。

`user-list` の有効な値については、[表 5-5](#) を参照

`deny:user-list` ユーザーアクセス拒否リストに追加するユーザー名。複数のユーザーログイン名を指定する場合は、空白またはコンマで区切る。空白で区切る場合は、ID のリストを引用符で囲む。
`user-list` の有効な値については、[表 5-5](#) を参照

プリントサーバーの次のどちらかのファイルの指定されたフォーム用の許可または拒否のユーザーアクセスリストに、指定した 1 人以上のユーザーが追加されます。

```
/etc/lp/forms/form-name/allow または
/etc/lp/forms/form-name/deny
```

3. ユーザーアクセスの許可および拒否リストを確認します。

```
# lpforms -f form-name -l
```

例 — フォームへのユーザーアクセスを制限する

次の例は、ユーザー `nathan` と `marcia` にのみ `check` フォームへのアクセスを許可する方法を示しています。

```
# lpforms -f check -u allow:nathan,marcia
```

次の例は、ユーザー `jones` と `smith` による `dental` フォームへのアクセスを拒否する方法を示しています。

```
# lpforms -f dental -u deny:"jones,smith"
```

▼ フォームへのプリンタアクセスを制限する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインするか、同等の役割になります。
2. プリンタでのフォームの使用を許可または拒否します。

```
# lpadmin -p printer-name -f allow:form-list | deny:form-list
```

<code>-p printer-name</code>	フォームの許可リストまたは拒否リストを作成するプリンタ名
<code>-f allow:form-list deny:form-list</code>	許可または拒否リストに追加されるフォーム名。複数のフォーム名は空白またはコンマで区切る。空白で区切る場合は、フォーム名のリストを引用符で囲む

指定した1つ以上のフォームは、プリンタサーバーの次のどちらかのファイルの許可または拒否フォームリストに追加されます。

```
/etc/lp/printers/printer-name/form.allow  
/etc/lp/printers/printer-name/form.deny
```

3. 許可および拒否フォームリストを確認します。

```
# lpstat -p printer-name -l
```

例 — フォームへのプリンタアクセスを制限する

次の例は、プリンタ `luna` に `medical`、`dental`、`check` の各フォームへのアクセスのみを許可する方法を示しています。

```
# lpadmin -p luna -f allow:medical,dental,check
```

次の例は、プリンタ `luna` による `medical`、`dental`、`check` の各フォームへのアクセスを拒否する方法を示しています。

```
# lpadmin -p luna -f deny:"medical dental payroll"
```

フォントの管理

レーザープリンタがある場合は、PostScript 用のフォントをインストールして管理する必要がある可能性があります。PostScript フォントをインストールするシステムと、その管理方法を決定する必要がある可能性があります。多くのプリンタの場合、プリンタのインストール作業の一部としてフォントを設定します。

PostScript フォントは、プリンタかプリンタと通信を行うシステムのどちらかに、アウトライン形式で格納されます。文書の印刷時に、PostScript インタプリタは、アウトライン記述から適切な大きさの各文字を必要に応じて生成します。文書に必要なフォントが使用するプリンタに格納されていない場合は、文書が印刷される前にそのフォントをプリンタに転送しなければなりません。この転送処理を「フォントのダウンロード」といいます。

フォントは、次のいくつかの方法で格納または使用されます。

- 「プリンタ常駐フォント」は、プリンタに常時格納されています。これらのフォントは、メーカーによってプリンタの読み取り専用メモリー (ROM) にインストールされている場合があります。プリンタがディスクを備えている場合は、そのディスクにフォントをインストールしなければならないことがあります。ほとんどの PostScript プリンタには、35 の標準フォントが付いています。
- 「常時ダウンロードフォント」は、PostScript の `exitserver` プログラムを使用してプリンタに転送されます。常時ダウンロードフォントは、プリンタの電源を切るまでプリンタのメモリーに残っています。ダウンロードフォントに割り当てられたメモリーによって、PostScript 印刷要求では利用可能なサーバーのメモリーが減少します。`exitserver` プログラムを使用するには、プリンタシステムのパスワードが必要で、プリンタ管理者が使用します。プリンタで出力する大部分の印刷要求に特定のフォントを使用するときは、そのフォントを常時ダウンロードするようにしてください。
- あまり頻繁に使用されないか、特殊な目的で使用されるフォントは、ユーザーのシステムに格納できます。ユーザーは、印刷要求を出すときにこれらのフォントを指定できます。そのフォントは印刷要求に追加されてプリンタに転送されます。印刷要求が処理されると、フォントに割り当てられたメモリー空間は、他の印刷要求が使用できるように解放されます。
- 「ホスト常駐フォント」は、多数のユーザーによって共有されるシステムに格納されます。フォントを格納するシステムは、プリンタサーバーでも印刷クライアントでもかまいません。各ユーザーは印刷する文書のフォントを要求指定できます。この方法は、多数のフォントを利用できるときや、これらのフォントがすべての印刷要求で使用されるとは限らないときに便利です。そのフォントがプリンタサーバーに接続されたプリンタでのみ使用される場合は、プリンタサーバーに格納してください。そのフォントが1つのシステム上で、ネットワーク上の複数のプリンタに要求を依頼する可能性があるユーザーによって使用される場合は、フォントはそのユーザーのシステムに格納してください。

LP 印刷サービスには、ホスト常駐フォントを管理するための特殊なダウンロード用のフィルタがあります。また、`troff` プログラムで使用するために、多くの PostScript プリンタに搭載された 35 の標準 PostScript フォント用の `troff` (1) のフォント幅テーブルも提供しています。

プリンタ常駐フォントの管理

ほとんどの PostScript プリンタは、プリンタ内蔵の ROM にフォントが搭載されています。プリンタによっては、追加フォントを格納するためのディスクが用意されています。プリンタをインストールするときに、そのプリンタ用のフォントリストにプリンタ常駐フォントを追加してください。プリンタ常駐フォントがわかっている場合は、フォントをネットワーク経由で必要以上に転送することがなくなります。各プリンタには搭載されているフォントの独自のリストがあります。

```
/etc/lp/printers/printer-name/residentfonts
```

プリンタをプリンタサーバーに接続するときには、プリンタサーバー上においてプリンタにダウンロードできるフォントが、`residentfonts` ファイル内のリストに含まれているかどうかを確認します。

プリンタ常駐フォントのリストが入っているファイルは、`vi` などのテキストエディタを使用して編集しなければなりません。

ホスト常駐フォントのダウンロード

PostScript の文書がプリンタにダウンロードされていないフォント指定を含んでいるときは、「ダウンロードフィルタ」がこの印刷要求を管理します。ダウンロードフィルタは PostScript の文書作成規則を使用して、ダウンロードするフォントを決定します。

LP 印刷フィルタには、高速フィルタと低速フィルタがあります。高速フィルタは、印刷するファイルをすばやく準備し、フィルタが処理している間にプリンタにアクセスしなければなりません。低速フィルタはファイルの変換に時間がかかり、フィルタが処理している間にプリンタにアクセスする必要はありません。低速フィルタの例には、ASCII ファイルから PostScript ファイルへのフィルタがあります。

ダウンロードフィルタは高速フィルタです。フォントがプリンタサーバー上にある場合は、フォントを自動的にダウンロードします。また、ダウンロードフィルタを使用して、プリンタサーバーにフォントを転送することもできます。そのためには、`lp -y` コマンドを指定して、ダウンロードフィルタを低速フィルタとして呼び出すための新しいフィルタテーブルのエントリを作成します。あるいは、入力タイプを変更して、このフィルタの選択を強制することもできます。

ダウンロードフィルタは、次の 5 つの作業を実行します。

1. PostScript の文書を検索して、要求されているフォントを判別します。これらの要求は、ヘッダコメントの PostScript 構造化コメント `%%DocumentFonts: font1 font2 ...` で指定されます。

2. プリンタ常駐フォントのリストを検索して、要求されたフォントをダウンロードしなければならないかどうかを判別します。
3. フォントがプリンタ上になれば、ダウンロードフィルタは(マップテーブルから適切なファイル名を読み取って) ホスト常駐フォントのディレクトリを検索し、要求されたフォントが利用可能かどうかを判別します。
4. そのフォントが利用可能であれば、フィルタはそのフォントのファイルを取り出し、印刷するファイルに追加します。
5. フォント定義ファイルとソースファイル(印刷するファイル)を PostScript プリンタに送ります。

ホスト常駐フォントのインストールと管理

フォントによっては、ホストシステムに格納されており、特定の印刷要求に応じてプリンタに転送されるものがあります。管理者は、システム上のすべてのユーザーが PostScript フォントを使用できるように管理する必要があります。そのためには、これらのフォントのインストール方法とインストール場所を知っておかなければなりません。フォントは名前で要求され、ファイルに格納されているので、LP 印刷サービスはフォント名とフォントを定義しているファイル名を対応付けるマップファイルを持っています。ホスト常駐フォントをインストールするときには、マップファイルとフォントリストの両方を更新しなければなりません。

PostScript プリンタで利用できるフォントは、ユーザーが作成した `/usr/share/lib/hostfontdir/typeface/font` ディレクトリに格納されます。この場合、*typeface* は *palatino* や *helvetica* などの名前に置き換えられ、*font* は *bold* や *italic* などの名前に置き換えられます。

▼ ダウンロードされた PostScript フォントをインストールする方法

1. プリンタサーバーまたは印刷クライアントにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. `/etc/lp/printers/printer-name` ディレクトリに変更します。

```
# cd /etc/lp/printers/printer-name
```

プリンタ名	ダウンロードされた PostScript フォントをインストールするプリンタ名
-------	---

3. **residentfonts** ファイルが存在しない場合は作成します。

```
# touch residentfonts
```

常駐させるダウンロードフォントを初めて追加する場合は、このファイルが存在しないことがあります。

4. **residentfonts** ファイルを編集して、すべてのプリンタ常駐フォントとダウンロードフォントを追加します。

▼ ホスト常駐 PostScript フォントをインストールする方法

1. プリンタサーバーまたは印刷クライアントにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。

2. **hostfontdir** ディレクトリが存在しない場合は作成します。

```
# cd /usr/share/lib
# mkdir hostfontdir
# chmod 775 hostfontdir
```

3. 新しい書体のディレクトリが存在しない場合は作成します。

```
# mkdir typeface
```

4. フォントファイルを適切なディレクトリにコピーします。

```
# cp filename /usr/share/lib/hostfontdir/typeface/font
```

5. マップテーブルに組み込むフォント名とファイル名を追加します。

- a. **/usr/share/lib/hostfontdir** ディレクトリに変更します。

- b. **vi** などのテキストエディタを使用して **map** ファイルを編集します。

テーブルに追加したいフォントごとに1行ずつエントリを追加します。エントリには、フォント名、スペース1個、フォントが常駐するファイル名の順に入力します。次に例を示します。

```
Palatino-Bold /usr/share/lib/hostfontdir/palatino/bold
```

- c. ファイルを保存します。

適切なシステム上のマップテーブルにサンプルエントリを入れておけば、ユーザーは各自の印刷ジョブに(たとえば、Palatino Bold などの)フォントを適用できます。このフォントを含む印刷要求を依頼すると、LP印刷サービスはそのファイルに **/usr/share/lib/hostfontdir/palatino/bold** のコピーを追加してから、プリンタに送信します。

6. **troff** を使用している場合は、このフォント用の新しいフォント幅テーブルを標準 **troff** フォントディレクトリ内で作成します。

第 7 章

LP 印刷サービスのカスタマイズ (手順)

この章では、LP 印刷サービスのカスタマイズについての概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 146 ページの「プリンタポート特性を調整する方法」
- 150 ページの「サポートされていないプリンタの `terminfo` エントリを追加する方法」
- 154 ページの「独自のプリンタインタフェースプログラムを設定する方法」
- 165 ページの「新しい印刷フィルタを作成する方法」
- 169 ページの「新しいフォーム定義を作成する方法」

プリンタの概要については、第 2 章を参照してください。

プリンタポート特性の調整

LP 印刷サービスによって設定されるプリンタポート特性には、プリンタの通信設定と互換性がなければなりません。LP 印刷サービスから提供されたデフォルトのプリンタポート設定値がプリンタで機能しない場合は、プリンタのマニュアルを参照し、そのプリンタが LP 印刷サービスに対してどのような設定値が必要かを調べてください。プリンタ通信設定を設定および表示するには、`stty` コマンドを使用します。

表 7-1 は、LP 印刷サービスに使用されるデフォルトの `stty` 設定値を示しています。

表 7-1 LP 印刷サービスに使用される `stty` のデフォルト設定値

オプション	説明
<code>-9600</code>	ボーレートを 9600 に設定する

表 7-1 LP 印刷サービスに使用される stty のデフォルト設定値 (続き)

オプション	説明
-cs8	8 ビットバイトを設定
-cstopb	1 バイト当たり 1 ストップビットを送信
-parity	パリティを生成しない
-ixon	XON/XOFF (START/STOP または DC1/DC3 ともいう) を使用可能にする
-opost	以下にリストされた設定値をすべて使用して「処理後出力」を実行する
-olcuc	小文字を大文字に割り当てない
-onlcr	改行をキャリッジリターン/改行に変更する
-ocrnl	キャリッジリターンを改行に変更しない
-onocr	カラム 0 でもキャリッジリターンを出力する
-n10	改行後の遅延なし
-cr0	キャリッジターン後の遅延なし
-tab0	タブ後の遅延なし
-bs0	バックスペース後の遅延なし
-vt0	垂直タブ後の遅延なし
-ff0	用紙送り後の遅延なし

▼ プリンタポート特性を調整する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. プリンタポートの特性を調整します。

```
# lpadmin -p printer-name -o "stty=options"
```

-p <i>printer-name</i>	ポート特性を調整するプリンタ名
-o "stty=options"	<i>options</i> で指定したポートの特性 (stty オプション) を設定する。このコマンドでは複数の stty オプションを変更できる。オプションを区切るには空白を使用し、オプションを一重引用符で囲む。オプションの全リストについては、stty(1) を参照。LP 印刷サービスで使用されるデフォルトの stty 設定については、表 7-1 を参照

3. プリンタポート特性が変更されたことを確認します。

```
# lpstat -p printer-name -l
```

例 — プリンタポート特性を調整する

次の例は、プリンタ luna のポート特性を設定する方法を示しています。parenb オプションはパリティチェック/生成を使用可能にし、parodd は奇数パリティの生成を設定し、cs7 は文字サイズを7ビットに設定します。

```
# lpadmin -p luna -o "stty='parenb parodd cs7'"
```

次の例は、プリンタ venus の端末ボーレートを 19200 に設定する方法を示しています。

```
# lpadmin -p venus -o "stty=19200"
```

サポートされていないプリンタの terminfo エントリを追加する

LP 印刷サービスは、インタフェースプログラムと terminfo データベースを使用して各プリンタを初期化し、選択されたページサイズ、文字ピッチ、行ピッチ、および文字セットを設定します。

各プリンタは、terminfo データベース内で短縮名を使用して識別されます。terminfo データベースで使用される名前は、TERM シェル変数を設定するのに使用する名前と同じです。また、この名前はプリンタを設定するときに指定するプリンタタイプでもあります。たとえば、各種の PostScript プリンタのエントリは、`/usr/share/lib/terminfo/P` ディレクトリに入っています。SunOS システムに組み込まれているデフォルトエントリは、PS (PostScript 用) と PSR (Reverse PostScript 用) です。

プリンタの terminfo エントリが見つからなくても、ページサイズ、ピッチ、および文字セットを自動選択しないで、LP 印刷サービスでプリンタを使用できます。ただし、印刷要求ごとに正しいモードでプリンタが設定されないという問題が起きることがあります。

使用中のプリンタタイプの terminfo エントリがないが、プリンタを正しいモードに設定しておきたい場合は、プリンタで使用するインタフェースプログラムをカスタマイズするか、terminfo データベースにエントリを追加できます。terminfo データベース内の端末エントリやプリンタエントリには、多数の項目が入っており、定義されています。しかし、LP 印刷サービスはそのうち 50 未満の項目しか使用しません。表 7-2 は、プリンタに必要な terminfo 項目を示しています。

表 7-2 プリンタに必要な terminfo 項目

項目	説明
ブール値	
cpix	文字ピッチを変更すると解像度が変わる
daisy	プリンタで文字セットを変更するには演算子が必要
lpix	行ピッチを変更すると解像度が変わる
数値	
bufsx	印刷前にバッファされるバイト数
cols	1 行の列数
cps	1 秒あたりの文字の平均印刷速度
it	最初はスペース n 個分ごとのタブ
lines	1 ページの行数
orc	1 文字あたりの水平解像度
orhi	1 インチあたりの水平解像度
orl	1 行あたりの垂直解像度
orvi	1 インチあたりの垂直解像度
文字列	
chr	水平解像度を変更する
cpi	1 インチあたりの文字数を変更する
cr	キャリッジリターン
csnm	文字セット名のリスト
cudl	1 行下げる
cud	キャリッジを n 行下に移動する
cuf	キャリッジを n 列右に移動する
cvr	垂直解像度を変更する
ff	紙送りする
hpa	水平絶対位置
ht	次の 8 スペースタブストップまでのタブ
if	初期化ファイル名
iprogr	初期化プログラムのパス名

表 7-2 プリンタに必要な terminfo 項目 (続き)

項目	説明
is1	プリンタ初期化文字列
is2	プリンタ初期化文字列
is3	プリンタ初期化文字列
文字列	
lpi	1 インチあたりの行数を変更する
mgc	マージン (上、下、横) をすべて消去する
rep	文字を n 回繰り返す
rwidm	倍幅の印刷を使用不可にする
scs	文字セットを選択する
scsd	文字セットの定義を開始する
slines	ページの長さを 1 ページあたり n 行に設定する
smgl	現在の列の左マージンを設定する
smglp	左マージンを設定する
smgr	現在の列の右マージンを設定する
smgrp	右マージンを設定する
smglr	左右のマージンを設定する
msgt	現在の行の上マージンを設定する
smgtp	上マージンを設定する
smgb	現在の行の下マージンを設定する
smgbp	下マージンを設定する
smgtb	上下のマージンを設定する
swidm	倍幅の印刷を使用可能にする
vpa	垂直絶対位置

▼ サポートされていないプリンタの terminfo エントリを追加する方法

注 - プリンタの terminfo エントリを作成する前に、まず、そのプリンタをサポートする既存の terminfo エントリがないことを確認してください。そのためには、類似するプリンタがあれば、そのエントリを使用してプリンタを設定してください。

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。

2. プリンタの **terminfo** エントリ名を決定します。

`/usr/share/lib/terminfo` ディレクトリ内のディレクトリには、有効な terminfo エントリがすべて入っています。それを参考にしてプリンタ名を選択してください。

3. プリンタの **terminfo** エントリファイルを作成します。

LP 印刷サービスに新しいプリンタを追加するために terminfo エントリ内で定義しなければならない項目については、表 7-2 を参照してください。terminfo データベースの構造については、terminfo(4) のマニュアルページを参照してください。

新しい terminfo エントリを作成しやすいように、`infocmp` コマンドを使用して既存の terminfo エントリをファイルに保存します。これは、作成したいエントリに似た terminfo エントリがある場合に便利な方法です。たとえば、次のコマンドで `ps` エントリを `ps_cust` ファイルに保存すると、新しい terminfo エントリになります。

```
infocmp ps > ps_cust
```

4. **terminfo** エントリをコンパイルして **terminfo** データベースに入れます。

```
# tic terminfo_entry
```

```
terminfo_entry          作成した terminfo エントリファイル
```

5. `/usr/share/lib/terminfo` ディレクトリ内で新しい **terminfo** エントリファイルをチェックします。

プリンタインタフェースプログラムのカスタマイズ

標準プリンタインタフェースプログラムでサポートされないプリンタを使用する場合は、独自のプリンタインタフェースプログラムを提供できます。標準プログラムをコピーし、指定したプリンタを使用するように LP 印刷サービスに指示できます。ただし、まず標準プログラムについて理解する必要があります。次の節では、標準プログラムについて説明します。

プリンタインタフェースプログラムの機能は次のとおりです。

- 必要に応じてプリンタポートを初期化する。標準プリンタインタフェースプログラムは、`stty` コマンドを使用してプリンタポートを初期化する。
- プリンタハードウェアを初期化する。標準プリンタインタフェースプログラムは、`terminfo` データベースと `TERM` シェル変数から制御シーケンスを取得する。
- 必要に応じてバナーページを印刷する。
- 印刷要求で指定された部数を印刷する。



注意 - リリース 3.2 より前の UNIX System V のプリンタインタフェースプログラムを使用している場合でも、そのプログラムは、SunOS 5.9 または互換 LP 印刷サービスで使用できます。ただし、一部の `-o` オプションは SunOS 5.9 または互換 LP 印刷サービスでは標準化されていて、すべてのプリンタインタフェースプログラムに渡されます。これらのオプションは、古いインタフェースで使用される同じ名前のオプションの妨げとなることがあります。

プリンタポートを開く処理は、プリンタインタフェースプログラムではなく LP 印刷サービスが受け持ちます。プリンタポートは標準出力としてプリンタインタフェースプログラムに与えられ、プリンタはプリンタインタフェースプログラムの「制御端末」として識別されるので、ポートが「ハング」するとプリンタインタフェースプログラムに `SIGHUP` 信号が送信されます。

標準プリンタインタフェースプログラム

LP 印刷サービスは、標準 (モデル) プリンタインタフェースプログラム `/usr/lib/lp/model/standard` を使用して、表 7-3 の印刷デフォルトを設定します。

表 7-3 デフォルトのプリンタポート特性

特性	デフォルト設定
デフォルトフィルタ	None
文字ピッチ	None
行ピッチ	None
ページ幅	None
ページ長	None
文字セット	None
stty オプション	9600 cs8 -cstopb -parenb -parodd ixon -ixany opost -olcuc onlcr -ocrnl -onocr -onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0
終了コード	0

stty モードのカスタマイズ

ボーレートや出力オプションなどの端末特性を変更する必要がある場合は、標準プリンタインタフェースプログラム内で、次のコメントから始まるセクションを探します。

```
## Initialize the printer port
```

終了コード

印刷し終わると、インタフェースプログラムは印刷ジョブの状態を示すコードを返して終了します。終了コードは、プリンタインタフェースプログラムの最後のエントリです。

次の表は、終了コードとそのコードが LP 印刷サービスでどのように解釈されるかを示しています。

表 7-4 プリンタインタフェースプログラムの終了コード

コード	LP 印刷サービスにとっての意味
0	印刷要求は正常に完了した。プリンタ障害が発生した場合は、クリアされた

表 7-4 プリンタインタフェースプログラムの終了コード (続き)

コード	LP 印刷サービスにとっての意味
1 から 127	要求の印刷中に問題が発生した (たとえば、印字できない文字が多すぎる、要求がプリンタの容量を超えているなど)。LP 印刷サービスは、その要求を依頼したユーザーに、印刷中にエラーが発生したことを通知する。このエラーはその後の印刷要求には影響しない。プリンタ障害が発生するとクリアされる
128	このコードは、LP 印刷サービスが内部で使用するために予約されている。インタフェースプログラムは、このコードを返して終了してはいけない
129	要求の印刷中にプリンタ障害が発生した。この障害は、その後の印刷要求に影響を及ぼす。プリンタの障害回復が LP 印刷サービスに管理者によって問題が解決されるまで待つように指示すると、LP 印刷サービスはプリンタを使用不可にする。障害回復後に印刷を続けようとする、LP 印刷サービスはプリンタを使用不可にしないが、数分後にそのまま印刷しようとする
>129	これらのコードは、LP 印刷サービスが内部で使用するために予約されている。インタフェースプログラムは、この範囲内のコードを返して終了してはいけない

プログラムがコード 129 を返して終了すると、root はプリンタ障害を警告されます。また、LP 印刷サービスは、障害がクリアされた後に要求を最初から印刷し直さなければなりません。要求全体を印刷し直したくない場合は、インタフェースプログラムに障害メッセージを LP 印刷サービスへ送信させることもできますが、障害がクリアされるまで待つこととなります。障害がクリアされると、インタフェースプログラムはファイルの印刷を再開できます。印刷が終了すると、プリンタインタフェースプログラムは障害が発生しなかった場合と同様に終了コード 0 を返すことができます。このアプローチには、障害が自動的にクリアされた場合に、それをインタフェースプログラムが検出できるので、管理者がプリンタを再び使用可能にする必要がないという利点もあります。

障害メッセージ

lp.tell プログラムを使用すると、LP 印刷サービスに障害メッセージを送信できます。このプログラムは、標準プリンタインタフェースコード内の LPTELL シェル変数によって参照されます。プログラムは標準入力を取り込んで LP 印刷サービスに送信し、LP 印刷サービスは管理者にプリンタ障害を警告するメッセージを出します。標準入力が空であれば、lp.tell は警告を開始しません。lp.tell プログラムの例として、次のコメントの直後の標準プリンタインタフェースコードを確認してください。

```
# Set up the $LPTELL program to capture fault messages here
```

特殊な終了コード 129 または `lp.tell` プログラムを使用すると、プリンタインタフェースプログラムはプリンタ自体を使用不可にする必要がありません。インタフェースプログラムは、プリンタを直接使用不可にできますが、その場合は障害警告メカニズムが無効になります。LP 印刷サービスがプリンタ障害を検出した場合にのみ警告が送信され、特殊終了コードと `lp.tell` プログラムはその主要検出ツールです。

LP 印刷サービスがいずれかの時点でファイルの印刷を中断しなければならない場合は、TERM 信号 (トラップ番号 15) を使用してインタフェースプログラムを終了させます。(詳細については、`kill(1)` および `signal(3C)` のマニュアルページを参照してください)。プリンタインタフェースプログラムが他の信号を受信して終了した場合、LP 印刷サービスはその後の印刷要求は影響されないものとみなし、そのプリンタを使用し続けます。LP 印刷サービスは、要求を依頼したユーザーに、その要求が正常に終了しなかったことを通知します。

インタフェースが最初に呼び出されると、信号 HUP、INT、QUIT、PIPE (トラップ番号 1、2、3、13) は無視されます。標準インタフェースは、信号が適切な時期にトラップされるように、この動作を変更します。標準インタフェースはこれらの信号の受信をプリンタの問題を示す警告として解釈し、信号を受信すると障害警告を発生します。

カスタマイズされたプリンタインタフェースプログラムの使用方法

カスタマイズされたプリンタインタフェースプログラムを作成し、プリンタサーバー上で標準プリンタインタフェースプログラムの代わりに使用できます。そのためには、`lpadmin` コマンドを使用して、プログラムを特定のプリンタの LP 印刷サービスに登録します。

▼ 独自のプリンタインタフェースプログラムを設定する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 独自のプリンタインタフェースプログラムがあるかどうかに基づいて次の手順を選択します。

カスタムプリンタインタフェースの状態	次の手順
独自のプリンタインタフェースプログラムがないので作成する必要がある	手順 3 へ

カスタムプリンタインタフェースの状態	次の手順
--------------------	------

すでに独自のプリンタインタフェースプログラムがある	手順 5へ
---------------------------	-------

- 標準プリンタインタフェースプログラムをコピーします。

```
# cp /var/spool/lp/model/standard custom-interface
```

- 標準プリンタインタフェースプログラムのコピーを必要に応じて変更します。
詳しくは、151 ページの「標準プリンタインタフェースプログラム」のプログラムの説明を参照して、何を変更する必要があるかを判断してください。

- 特定のプリンタ独自のプリンタインタフェースプログラムを設定します。

```
# lpadmin -p printer-name -i custom-interface
```

`-p printer-name` 独自のプリンタインタフェースプログラムを使用するプリンタ名

`-i custom-interface` 独自のプリンタインタフェースプログラム名

独自のプリンタインタフェースプログラムが LP 印刷サービスに登録され、ユーザーが印刷要求を出すと、そのプリンタに使用されます。

- 独自のプリンタインタフェースプログラムが `/etc/lp/printers/printer-name/configuration` ファイルに追加されたかどうかを確認します。

例 — 独自のプリンタインタフェースプログラムを設定する

次の例は、プリンタ `luna` の独自のプリンタインタフェースプログラム `custom` を設定する方法を示しています。

```
# lpadmin -p luna -i custom
```

次の例は、プリンタ `venus` からプリンタ `asteroid` に独自のプリンタインタフェースプログラムをコピーする方法を示しています。

```
# lpadmin -p asteroid -e venus
```

新しい印刷フィルタの作成

フィルタは、LP 印刷サービスがプリンタで解釈できないタイプのファイルを印刷する必要があるときに使用されます。新しい印刷フィルタを作成するのは簡単ではありません。通常は幅広い経験が必要です。新しい印刷フィルタを定義するには、次の2つの手順があります。

- 印刷フィルタプログラムを作成する
- 印刷フィルタ定義を作成する

印刷フィルタは、必要に応じて簡単なものでも複雑なものでもかまいません。フィルタには、入力形式、出力形式、そのフィルタ内でコマンド行引数を処理する言語を提供する複雑なオプションが入っています。

PostScript 以外のプリンタを使用する場合は、必要に応じて印刷フィルタを作成して追加する必要があります。まず、印刷フィルタの機能と、フィルタプログラムが満たさなければならない要件を理解しておく必要があります。

印刷フィルタプログラムの作成

LP 印刷サービスには、`/usr/lib/lp/postscript` ディレクトリにフィルタプログラムが組み込まれています。これらのフィルタは、宛先プリンタが PostScript 形式のデータを必要とするほとんどの PostScript 印刷の状況に対応します。印刷フィルタプログラムは、2進の実行可能プログラムでなければなりません。

フィルタのタイプ

印刷フィルタには、高速フィルタと低速フィルタの2種類があります。

高速フィルタは、ファイルの印刷準備にあまり処理時間がかかりません。また、実行するときにはプリンタにアクセスしなければなりません。印刷障害を検出する印刷フィルタは、高速フィルタでなければなりません。PRINTER キーワードをフィルタオプションとして使用するフィルタは、高速フィルタとしてインストールしなければなりません。

低速フィルタは、ファイルの印刷準備に大量の処理時間がかかります。また、低速フィルタは実行するときにはプリンタにアクセスする必要がありません。低速フィルタはバックグラウンドで実行されるので、プリンタと連結する必要がなく、低速フィルタを必要としないほかのファイルを印刷できます。

ファイルの変換

LP 印刷サービスは、印刷フィルタを使用して、ある内容形式から別の内容形式にファイルを変換します。プリンタごとに、受け付けられるファイル内容形式を指定できます。ユーザーは印刷要求を出すときにファイル内容形式を指定し、LP 印刷サービスはその内容形式のファイルを印刷できるプリンタを探します。多くのアプリケーションは各種プリンタに合わせてファイルを生成できるので、通常はこれで十分です。ただし、アプリケーションによっては、利用できるプリンタ上で印刷できないファイルを生成するものがあります。

LP 印刷サービスは、プリンタで直接受け付けられない形式のファイルの印刷要求を受信するたびに、印刷要求の内容形式を利用できる (または指定された) プリンタの内容形式と一致させようとします。一致すると、そのファイルはフィルタをかけなくてもプリンタに直接送信できます。一致する形式が見つからない場合や、内容形式でフィルタを使用するように指定されている場合は、LP 印刷サービスはファイルの内容形式を利用できるフィルタの入力内容形式と一致させ、フィルタの出力形式をプリンタの内容形式と一致させようとします。適切なフィルタが見つかったら、印刷要求はフィルタを通して渡されます。

特殊印刷モードの処理

印刷フィルタは、特殊モードを処理し特殊ページの印刷を要求します。特殊印刷モードは、カスタマイズされたフィルタが必要な印刷要求の特性を印刷するために必要です。フィルタは、次の特性を処理します。

- プリンタタイプ
- 文字ピッチ
- 行ピッチ
- ページ長
- ページ幅
- 印刷ページ数
- 文字セット
- フォーム名
- 部数

LP 印刷サービスは、これらの特性にデフォルト設定を提供します。ただし、一部の特性については、印刷フィルタの方が効率よく処理できます。たとえば、プリンタによっては、LP 印刷サービスより効率よく複数のコピーを処理できるものがあり、その場合は複数コピーのページ制御用フィルタを提供できます。

プリンタ障害の検出

各プリンタは、独自の 방법으로プリンタ障害を検出し、障害信号を LP 印刷サービスに送信します。LP 印刷サービスは、印刷中にハングアップ (キャリア喪失) と過剰な遅延の有無をチェックするだけです。

プリンタによっては、優れた障害処理能力を持ち、障害の原因を記述するメッセージを送信できるものがあります。また、キャリア信号の喪失やデータフローのシャットオフを示すものとは異なる信号を使用して障害を示すプリンタもあります。これらの付加的なプリンタ障害情報を解釈するには、フィルタが必要です。

また、フィルタは印刷要求を保留し、プリンタ障害がクリアされるまで待ってから印刷を再開します。この機能を使用すると、中断された印刷要求全体を印刷し直す必要がありません。プリンタに使用される制御シーケンスを認識するフィルタだけが、ファイルの改ページ位置を判別できます。したがって、その種のフィルタ以外は、障害がクリアされた後で印刷を再開する必要があるファイル内の位置を検出できません。

印刷フィルタがメッセージを生成すると、そのメッセージは LP 印刷サービスによって処理され、警告が使用可能になっている場合は、システム管理者に警告が送信されます。99 ページの「[障害の通知の設定](#)」を参照してください。

印刷フィルタプログラムの要件

印刷フィルタは単純なものでも複雑なものでもかまいませんが、次の要件を満たさなければなりません。

- フィルタは、標準入力からファイルの内容を取得して、変換後のファイルを標準出力に送信しなければなりません。
- 外部ファイルを参照するプログラムはフィルタとして使用できません。一般にユーザーは `troff`、`nroff` などのワープロプログラムをフィルタとして使用しがちです。LP 印刷サービスは、フィルタプログラムから「組み込みファイル」と呼ばれる他のファイルへの参照を認識しません。`troff` と `nroff` はファイルを組み込めるので、フィルタとして使用すると失敗することがあります。プログラムが処理を完了するために他のファイルを必要とする場合は、フィルタとして使用しないでください。
- フィルタは、一般にユーザーがアクセスできないファイルに依存しないでください。ユーザーが直接実行するときにはフィルタが失敗すると、LP 印刷サービスで実行しても失敗します。
- 低速フィルタは、ファイル内のエラーに関するメッセージを標準エラーに送信しますが、高速フィルタは送信しません。低速フィルタからのエラーメッセージは収集され、印刷要求を出したユーザーに送信されます。
- 低速フィルタが信号を受信したために終了すると、印刷要求が停止され、その要求を出したユーザーに通知されます。同様に、低速フィルタが 0 以外の終了コードを返して終了すると、印刷要求が停止され、ユーザーに通知されます。高速フィルタからの終了コードは、異なる方法で処理されます。

フィルタにプリンタ障害を検出させたい場合は、次の要件も満たしてください。

- フィルタは終了する前にできるだけ障害がクリアされるまで待つ必要があります。また、障害がクリアされたら、印刷を停止したページの先頭から印刷を再開してください。継続機能を使用したくない場合は、LP 印刷サービスは管理者に警告する前にフィルタを停止します。

- フィルタは、障害が認識されたらすぐにプリンタ障害メッセージを標準エラーに送信する必要があります。終了する必要はなく、障害がクリアされるまで待つことができます。
- フィルタは、ファイル内のエラーに関するメッセージを標準エラーに送信してはいけません。これらのメッセージは、ユーザーが読み込める標準出力に含める必要があります。
- フィルタは、ファイルの印刷が終了すると (ファイル内のエラーによって正常に印刷できなかった場合も) 終了コード 0 を返して終了する必要があります。
- フィルタは、プリンタ障害によって印刷要求を終了できなかった場合にのみ、0 以外の終了コードを返して終了する必要があります。
- フィルタをフィルタテーブルに追加する場合は、高速フィルタとして追加しなければなりません。

印刷フィルタ定義の作成

印刷フィルタ定義は、フィルタ、実行する印刷フィルタプログラム、実行する変換の種類などを LP 印刷サービスに指示します。/etc/lp/fd ディレクトリには、一連のフィルタ記述子ファイルが組み込まれています。これらのファイルは、フィルタの特性 (高速または低速フィルタなど) を記述し、フィルタプログラム (/usr/lib/lp/postscript/postdaisy など) を指します。

新しい印刷フィルタを定義するときには、フィルタプログラムを作成するだけでなく、印刷フィルタ定義を作成しなければなりません。印刷フィルタ定義には、LP 印刷サービスが使用する次の情報が入っています。

- 実行するフィルタプログラム名
- 受け付ける入力タイプ
- 生成する出力タイプ
- ジョブを送信できるプリンタタイプ
- ジョブを送信できる特定のプリンタ名
- フィルタタイプ (高速または低速)
- オプション

lpfilter コマンドへの直接入力として特性を入力できます。また、フィルタの特性を指定するファイルを作成し、ファイル名を lpfilter コマンドへの入力として使用することもできます。この種のファイルは「フィルタ記述子ファイル」と呼ばれ、/etc/lp/fd ディレクトリに配置してください。これらのファイルはフィルタそのものではなく、フィルタを指すものです。

情報をファイルに格納するかコマンド行に直接入力するかに関係なく、次の形式を使用してください。

```
Command: command-pathname [options]
Input types: input-type-list
Output types: output-type-list
Printer types: printer-type-list
Printers: printer-list
```

Filter type: fast or slow
Options: *template-list*

注 - Options 以外のフィルタ特性に複数の定義 (つまり複数行) を与えると、2 番目の定義のみが印刷サービスに使用されます。

情報はどんな順序で配置してもかまいません。また、すべての情報が必要とは限りません。値を指定しなければ、表 7-5 の値がデフォルトで割り当てられます。デフォルト値はあまり便利ではないので、明示的な値を指定するようにしてください。

表 7-5 lpfilter 引数のデフォルト値

項目	デフォルト
入力タイプ	任意
出力タイプ	任意
プリンタタイプ	任意
プリンタ	任意
フィルタタイプ	slow

コマンド

フィルタプログラムのフルパスを使用してください。プログラムが必ず必要とする固定オプションがある場合は、それらをこのコマンドに含めます。

入力タイプ

印刷フィルタが処理できるファイル内容形式のリストです。LP 印刷サービスには入力タイプ数の制限はありませんが、ほとんどのフィルタは 1 つのタイプしか受け付けません。いくつかのファイルタイプは、フィルタで処理できる程度に類似しています。14 文字以内の英数字とダッシュを使用した名前を指定できます。入力タイプ名の一部として下線 (_) は使用できません。

LP 印刷サービスは、一貫した命名規則に合致するように、これらの名前を使用してフィルタをファイルタイプと一致させます。たとえば、複数のフィルタが同じ入力タイプを受け付ける場合は、各フィルタ用に指定するときに、その入力タイプに同じ名前を使用します。ユーザーには、ファイルの印刷を依頼するときにファイルタイプの識別方法がわかるように、これらの名前を通知してください。

出力タイプ

出力タイプは、フィルタが出力として生成できるファイルタイプのリストです。入力タイプごとに、フィルタは1つずつ出力タイプを生成します。ただし、出力タイプはジョブごとに異なることがあります。出力タイプ名は、14文字以内の英数字とダッシュ(-)です。

出力タイプ名は、利用できる(ローカルまたはリモート)プリンタのタイプと一致するか、他のフィルタで処理される入力タイプと一致しなければなりません。LP印刷サービスは、ファイルを変換するために異なるフィルタを複数回かける必要があることがわかると、フィルタをシェルパイプラインにグループ化します。このように複雑な処理はほとんど必要ありませんが、LP印刷サービスではこの処理を実行できます。ユーザーが印刷したいあらゆるファイルを入力タイプにするフィルタの集合を探して、各ファイルをプリンタで処理できるファイルタイプに直接変換してください。

プリンタタイプ

プリンタタイプは、印刷フィルタがファイルを変換できるプリンタタイプのリストです。ほとんどのプリンタとフィルタの場合、これは出力タイプのリストと同じなので、フィルタ定義のこの部分はブランクにしておいてかまいません。しかし、別のタイプを使用することもできます。たとえば、初期化には1つのプリンタタイプを使用するプリンタでも、複数のファイル内容形式を認識できるものがあります。実際には、このプリンタには、各種ファイルタイプを、処理できるファイルタイプに変換する内部フィルタが付いています。したがって、1つのフィルタは、プリンタで処理できるファイルタイプに合った複数の出力タイプのいずれかを生成できます。印刷フィルタには、そのプリンタタイプで機能することを示すマークを付ける必要があります。

もう1つの例として、同じファイルタイプを受け付けるものとして表示される2つのモデルのプリンタを使用できます。ただし、内部に若干違いがあるので、一方のプリンタが生成する結果と異なります。それらのプリンタには、AとBなど、異なるプリンタタイプであることを示すラベルを付けます。この場合、Bはズレがある方のプリンタです。タイプBのプリンタで生成されるズレを考慮してファイルを調整するフィルタを作成します。このフィルタはこの種のプリンタタイプにのみ必要なもので、タイプBプリンタでのみ機能するものとして指定します。

プリンタ

一般に、印刷フィルタはその出力を受け付けるどのプリンタでも機能するので、通常はフィルタ定義のこの部分をスキップできます。

ただし、プリンタによっては、フィルタが生成する出力に適したものと適さないものがあります。たとえば、1台のプリンタを高速出力専用にして、フィルタを通す必要がないファイルのみをそのプリンタに送信できます。同じタイプの他のプリンタは、印刷前に広範囲にフィルタを通す必要があるファイルに使用できます。

フィルタタイプ

LP 印刷サービスは、156 ページの「フィルタのタイプ」で説明しているように、高速フィルタと低速フィルタを認識します。

印刷モードで (lp -y コマンドを使用して) 呼び出される低速フィルタは、印刷要求が出されたシステム上で実行しなければなりません。LP 印刷サービスはモード値をプリンタサーバーに渡せません。ただし、ファイル内容形式 (lp コマンドの -T オプションの後で指定) をプリンタサーバー上の内容形式と一致させることはできます。したがって、プリンタサーバー上で特殊モードを有効にしたい場合は、LP 印刷サービスが入力タイプと出力タイプを一致することができる内容形式を指定しなければなりません。

オプション

オプションは、各種の情報をフィルタコマンドのコマンド行引数にどのように変換するかを指定します。この情報には、ユーザーからの (印刷要求を伴う) 指定、プリンタ定義、要求の処理に使用されるフィルタによって実装される仕様などを含めることができます。

テンプレートによる印刷フィルタオプションの定義

印刷フィルタオプションの定義には 13 個の情報源があり、それぞれが「キーワード」で表されています。各オプションは「テンプレート」内で定義されます。テンプレートとは、フィルタのいずれかの特性値に基づいてフィルタコマンドに渡されるオプションを定義する、フィルタ定義内のステートメントです。

フィルタ定義で指定するオプションには、13 個のキーワードを使用しなくても、すべて使用しても、そのサブセットを使用してもかまいません。また、完全なフィルタ定義に複数の定義が必要な場合は、1 つのキーワードを複数回指定することもできます。印刷フィルタ定義における Options を定義するための 13 個のキーワードについては、表 7-6 を参照してください。

表 7-6 印刷フィルタオプションのキーワード

特性	キーワード	考えられるパターン	例
内容形式 (入力)	INPUT	内容形式	troff
内容形式 (出力)	OUTPUT	内容形式	postscript, impress
プリンタタイプ	TERM	プリンタタイプ	att495
プリンタ名	PRINTER	プリンタ名	lp1

表 7-6 印刷フィルタオプションのキーワード (続き)

特性	キーワード	考えられるパターン	例
文字ピッチ	CPI	四捨五入された 10 進数	10
行ピッチ	LPI	四捨五入された 10 進数	6
ページ長	LENGTH	四捨五入された 10 進数	66
ページ幅	WIDTH	四捨五入された 10 進数	80
印刷ページ数	PAGES	ページリスト	1-5, 13-20
文字セット	CHARSET	文字セット	finnish
フォーム名	FORM	フォーム名	invoice2
部数	COPIES	整数	3
特殊モード	MODES	モード	landscape

印刷フィルタ定義には、複数のテンプレートを含めることができます。複数のテンプレートを指定する場合は、1 行にコンマで区切って入力するか、先頭に `Options:` を付けて複数行に入力します。

テンプレートの形式は次のとおりです。

keywordpattern = replacement

`keyword` は、フィルタの特定の特性に関して登録されたオプションのタイプを識別します。

`pattern` は、キーワードの特定のオプションです。

`replacement` は、キーワードが指定した値のときに発生する動作です。

特定のフィルタのオプションを定義する方法を示す例として、印刷サービススケジューラを使って次の条件でフィルタに印刷要求を割り当てたいものと仮定します。

- フィルタで生成される OUTPUT のタイプが `impress` の場合は、フィルタに `-I` オプションを渡す。
- フィルタで生成される OUTPUT のタイプが `postscript` の場合は、フィルタに `-P` オプションを渡す。

上記の条件を指定するには、`lpfilter` コマンドのオプションとして次のテンプレートを与えます。

`Options: OUTPUT impress=-I, OUTPUT postscript=-P`

Options 行が長くなりすぎる場合は、次のように各テンプレートを別々の行に入れません。

```
Options: OUTPUT impress=-I
Options: OUTPUT postscript=-P
```

どちらのテンプレートでも、*keyword* は OUTPUT として定義されています。1 番目のテンプレートでは、パターンは *impress* で、*replacement* の値は -I です。2 番目のテンプレートでは、*pattern* の値は *postscript* で、*replacement* の値は -P です。

各種テンプレート (つまり、各キーワードの *pattern* 引数と *replacement* 引数) に与える値を見つけるには、次のことに注意する必要があります。

- INPUT テンプレートの値は、フィルタによる変換が必要なファイル内容形式からとっています。
- OUTPUT テンプレートの値は、フィルタが生成すべき出力タイプからとっています。
- TERM テンプレートの値はプリンタタイプです。
- PRINTER テンプレートの値は、最終出力を印刷するプリンタ名です。
- CPI、LPI、LENGTH、WIDTH の各テンプレートの値は、ユーザーの印刷要求、使用するフォーム、またはプリンタのデフォルト値からとっています。
- PAGES テンプレートの値は、印刷すべきページのリストです。通常、これはコマンドで区切ったページ範囲のリストです。各ページ範囲は、ダッシュで区切った 1 対の数値、または単一の数字からなります (たとえば、1-5,6,8,10 は 1 ページから 5 ページまでと、6 ページ、8 ページ、10 ページを示します)。ただし、印刷要求に -P オプションで値を与えると、印刷要求は変更されずに渡されます。
- CHARSET テンプレートの値は、使用する文字セット名です。
- FORM テンプレートの値は、lp -f コマンド (印刷要求を出すのに使用するコマンド) で要求したフォーム名です。
- COPIES テンプレートの値は、ファイルの印刷部数です。フィルタがこのテンプレートを使用する場合、この「1 部」にはフィルタによって生成される複数のコピーが含まれるので、LP 印刷サービスはフィルタがかけられたファイルの印刷部数から 1 を引きます。
- MODES テンプレートの値は、lp -y コマンドからとっています。ユーザーは複数の -y オプションを指定できるので、MODES テンプレートは複数の値をとることができます。値はユーザーが与えた順に左から右に適用されます。

テンプレートの *replacement* 部は、テンプレートの値をフィルタプログラムに与える方法を示します。通常はリテラルオプションですが、値の位置を示すブレースホルダのアスタリスク (*) が付いていることがあります。*pattern* と *replacement* には、ユーザー入力オプションからフィルタオプションへの複雑な変換を行うために、ed コマンドの正規表現の構文を使用することもできます。ed の正規表現の構文はすべて使用できます。たとえば、*(... \)* や *n* 構成を使用すると、*pattern* の各部を抽出して *replacement* にコピーできます。また、& を使用すると、*pattern* 全体を *replacement* にコピーできます。詳細については、ed(1) のマニュアルページを参照してください。

注 - *pattern* または *replacement* にコンマまたは等号 (=) を含める場合は、その前にバックスラッシュ (\) を付けてください。これらの文字の前にバックスラッシュが付いていると、*pattern* または *replacement* が使用されるときに削除されます。

▼ 新しい印刷フィルタを作成する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. 印刷フィルタプログラムを作成します。
印刷フィルタプログラムの概要については、156 ページの「印刷フィルタプログラムの作成」を参照してください。使用しやすいように、通常、フィルタ定義は `/usr/lib/lp/postscript` ディレクトリに入っています。作成したプログラムは、選択したディレクトリ内の `/usr/lib/lp` ディレクトリの下に入れる必要があります。
3. 印刷フィルタ定義を作成します。
印刷フィルタ定義の概要については、159 ページの「印刷フィルタ定義の作成」を参照してください。印刷フィルタ定義はテキストファイルに保存する必要があります。使用しやすいように、通常、フィルタ定義は `/etc/lp/fd` ディレクトリに入っており、接尾辞 `.fd` で識別されます。
4. 印刷フィルタをプリンタサーバーに追加します。
詳細は、128 ページの「印刷フィルタを追加する方法」を参照してください。

例 — 新しい印刷フィルタを作成する

次の例は、N37 または Nlp を simple に変換する印刷フィルタ定義を示します。

```
Input types: N37, Nlp, simple
Output types: simple
Command: /usr/bin/col
Options: MODES expand = -x
Options: INPUT simple = -p -f
```

次の例で、印刷フィルタプログラム名は `col` です。新しい印刷フィルタをプリンタサーバーに追加すると、ユーザーの印刷要求は次のように処理されます。

- ユーザーが次のコマンドを入力した場合

```
$ lp -y expand report.doc
```

印刷プログラムは次の引数を使用して実行され、ファイルが変換されます。

```
/usr/bin/col -x -p -f
```

- ユーザーが次のコマンドを入力した場合

```
$ lp -T N37 -y expand report.doc
```

印刷プログラムは次の引数を使用して実行され、ファイルが変換されます。

```
/usr/bin/col -x
```

次の例は、troff から PostScript に変換する印刷フィルタ定義を示します。

```
Input types: troff
```

```
Output types: postscript
```

```
Printer types: PS
```

```
Filter type: slow
```

```
Command: /usr/lib/lp/postscript/dpost
```

```
Options: LENGTH * = -l*
```

```
Options: MODES port = -pp, MODES land = -pl
```

```
Options: MODES group \=\([1-9]\) = -n\1
```

次の例で、フィルタプログラム名は dpost です。このプログラムは入力タイプ troff をとり、postscript 出力を生成し、タイプ PS (PostScript) のプリンタに機能します。ユーザーは、用紙方向を縦モードにするか横モードにするかを尋ねるプロンプトが表示されたときに、それぞれの省略形 port または land を指定するだけで済みます。これらのオプションは LP 印刷サービスに固有ではないので、ユーザーは lp -y コマンドを使用して指定しなければなりません。

新しい印刷フィルタをプリンタサーバーに追加すると、印刷要求は次のように処理されます。

- ユーザーが次のコマンドを入力し、横方向、ページ長 60 行で、troff ファイルタイプを PostScript プリンタ (タイプ PS) で印刷するように要求した場合

```
$ lp -T troff -o length=60 -y land -d luna ch1.doc
```

次の引数が指定された印刷フィルタプログラム dpost コマンドは、ファイルを変換します。

```
/usr/lib/lp/postscript/dpost -l60 -pl
```

- ユーザーが次のコマンドを入力した場合

```
$ lp -T troff -y group=4 -d luna ch1.doc
```

次の引数が指定された印刷フィルタプログラム dpost コマンドは、ファイルを変換します。

```
/usr/lib/lp/postscript/dpost -n4
```

新しいプリンタフォームの作成

新しいフォームを提供したい場合は、lpforms コマンドへの入力として9個の必須特性 (ページ長とページ幅など) に関する情報を入力し、その特性を定義しなければなりません。LP 印刷サービスは、この情報を次の2つの目的に使用します。

- フォーム上に正しく印刷されるようにプリンタを初期化する。
- フォームの処理方法に関する留意事項をシステム管理者に送信する。

フォーム名は、14文字以内の英数字と下線であれば、任意に選択して使用できます。情報は次の形式でなければなりません。

```
Page length: scaled number
Page width: scaled number
Number of pages: integer
Line pitch: scaled number
Character pitch: scaled number
Character set choice: character-set-name [,mandatory]
Ribbon color: ribbon-color
Comment:
informal notes about the form
Alignment pattern: [content-type] alignment pattern
```

省略可能な句、[,mandatory] は、ユーザーがフォームの文字セット選択を無効にできないことを意味します。content-type は位置揃えパターンを使用して指定できますが省略可能です。この属性を指定すると、印刷サービスは必要に応じて使用し、ファイルにフィルタをかけて印刷する方法を決定します。

2つの例外がありますが、情報は任意の順序で指定できます。例外は、Alignment pattern (位置揃えパターン) (必ず最後に配置しなければなりません) と comment (コメント) (必ず Comment: プロンプトの行に続かなければなりません) です。コメントにキー句 (Page length, Page width など) で始まる行が含まれている場合は、キー句が行頭にならないように、その行を> 文字で始めます。先頭の> 文字は、コメントから取り除かれて表示されません。

すべての情報を与えなければならぬわけではありません。表 7-7 の項目の値を指定しないときは、デフォルト値が割り当てられます。lpforms コマンドを実行する前に、新しいフォームに関して次の情報を収集してください。

表 7-7 フォームのデフォルト値

項目	デフォルト	説明
ページ長	66 行	フォームの長さ、または複数ページのフォームの場合は各ページの長さ。この情報は、行数でもインチ単位やセンチメートル単位でもかまわない

表 7-7 フォームのデフォルト値 (続き)

項目	デフォルト	説明
ページ幅	80 列	文字数、インチ数、またはセンチメートル数によるフォームの幅
ページ数	1	複数ページのフォームのページ数。LP 印刷サービスは、この数値と印刷フィルタ (利用できる場合) を使用して、位置揃えパターンを 1 つのフォームの長さに制限する。「位置揃えパターン」の説明を参照。フィルタが利用できない場合は、LP 印刷サービスは出力を打ち切らない
行ピッチ	1 インチあたり 6 行	フォーム上の行間隔。これは「リーディング」とも呼ばれる。2 行間の間隔、つまりベースラインからベースラインまでの間隔を 1 インチまたは 1 センチあたりの行数で表す
文字ピッチ	1 インチあたり 10 文字	フォームに表示される文字間隔。文字の間隔を 1 インチまたは 1 センチあたりの文字数で表す
文字セット選択肢	任意	このフォームに使用しなければならない文字セット、印字ホイール、またはフォントカートリッジ。ユーザーは、このフォームを使用するときに独自の印刷要求に別の文字セットを選択できる。また、単一の文字セットのみを使用するように指示できる
リボンの色	任意	フォームを常に特定のカラーリボンで印刷しなければならない場合、LP 印刷サービスはどの色を使用すべきかを示す装着警告メッセージを表示できる
コメント	(デフォルトなし)	ユーザーがフォームを理解する上で参考になる任意の情報。たとえば、フォーム名、そのバージョン、用途、または使用上の制限を示すことができる
位置揃えパターン	(デフォルトなし)	LP 印刷サービスが 1 枚のブランクフォームを埋めるために使用するサンプルファイル。フォームを装着するときに、このパターンを印刷して正しく位置揃えすることができる。また、印刷サービスに印刷方法が認識されるように、このパターンの内容形式を定義することもできる

注 - LP 印刷サービスは、位置揃えパターン内の重要な情報にマスクをかけようとしません。小切手を位置揃えするときなど、サンプルフォームに重要な情報を印刷したくない場合は、該当するデータにマスクをかける必要があります。LP 印刷サービスは、スーパーユーザーまたは lp としてログインしたユーザー以外は読み取れないように、位置揃えパターンを安全な場所に格納します。

フォーム情報を収集し終わったら、フォームを lpforms コマンドに入力します。lpforms コマンドに入力する前にこの情報を編集できるように、まず、この情報を別のファイルに記録してください。そうすれば、プロンプトの後で個々の情報を入力しなくても、そのファイルを入力として使用できます。

▼ 新しいフォーム定義を作成する方法

1. プリンタサーバーにスーパーユーザーまたは **lp** としてログインするか、同等の役割になります。
2. フォーム定義ファイルを作成します。
印刷フォームの作成方法の概要については、[167 ページの「新しいプリンタフォームの作成」](#)を参照してください。プリンタ定義はテキストファイルに保存してください。
3. フォームを LP 印刷サービスに追加します。

```
# lpadmin -p printer-name -M -f form-name
```
4. フォームをプリンタサーバーに追加します。
手順については、[132 ページの「フォームを追加する方法」](#)を参照してください。

第 8 章

LP 印刷サービス (リファレンス)

この章では、LP 印刷サービスの内容説明を提供します。

- 172 ページの「LP 印刷サービスの構造」
- 180 ページの「LP 印刷サービスのコマンド」
- 181 ページの「LP 印刷サービスの機能」
- 181 ページの「LP によるファイルの管理とローカル印刷要求のスケジューリングの方法」
- 182 ページの「ネットワーク印刷要求のスケジューリング」
- 183 ページの「印刷ファイルにフィルタを適用する」
- 183 ページの「プリンタインタフェースプログラムの機能」
- 183 ページの「lpsched デーモンによる印刷ジョブ状態の確認」
- 184 ページの「ログファイルの消去」

プリンタ関連作業の情報の参照箇所

プリンタ関連作業	参照箇所
Solaris プリンタマネージャと lp コマンドによるプリンタのセットアップ	第 4 章
セットアップ後の lp コマンドによるプリンタの管理	第 5 章
文字セット、フィルタ、フォーム、フォントの管理	第 6 章
LP 印刷サービスのカスタマイズ	第 7 章

LP 印刷サービス

「LP 印刷サービス」とは、ユーザーが作業を続けながらファイルを印刷できるようにするソフトウェアユーティリティの集合です。当初、印刷サービスは LP スプーラと呼ばれていました (LP はラインプリンタの意味ですが、現在ではレーザプリンタなどのさまざまな種類のプリンタも含まれます。スプール (Spool) は、system peripheral operation off-line の頭文字です)。

印刷サービスは、LP 印刷サービスソフトウェア、システム管理者が提供する印刷フィルタ、ハードウェア (プリンタ、システム、およびネットワーク接続) からなっています。

LP 印刷サービスの構造

この節では、LP 印刷サービスのディレクトリ構造、ファイル、ログ、およびコマンドについて説明します。

LP 印刷サービスのディレクトリ

LP 印刷サービスのファイルは、表 8-1 のように 7 つのディレクトリに分散されています。

表 8-1 LP 印刷サービスのディレクトリ

ディレクトリ	内容
/usr/bin	LP 印刷サービスのユーザーコマンド
/etc/lp	LP 構成ファイルの階層
/usr/share/lib	terminfo データベースディレクトリ
/usr/lib/print	lp 変換スクリプト、in.lpd デーモン、printd デーモン。 printd デーモンは /var/spool/print ディレクトリのすべての保留ジョブを 1 分間に 1 回転送する。転送するジョブがなくなると終了する
/usr/sbin	LP 印刷サービスの管理コマンド
/usr/lib/lp	lpsched プログラム、バイナリファイル、PostScript フィルタ、および model ディレクトリ。model ディレクトリには、標準プリンタインタフェースプログラムが入っている

表 8-1 LP 印刷サービスのディレクトリ (続き)

ディレクトリ	内容
/var/lp/logs	LP ログファイル。lpsched. <i>n</i> には lpsched からのメッセージが記録され、requests. <i>n</i> には完了した印刷要求についての情報が記録される
/var/spool/lp	ファイルが印刷待ち行列に入れられるスプーリングディレクトリ
/var/spool/print	LP 印刷サービスのクライアント側要求格納域

LP 印刷サービスの構成ファイル

lpsched デーモンが/etc/lp ディレクトリに格納する構成情報を、次の表に示します。



注意 - 表 8-2 の構成ファイルはプライベートインタフェースで、将来のリリースでは変更される可能性があります。現在の場所にあるファイルに依存する、または現在使用している形式のデータに依存するソフトウェアを構築しないようにしてください。

表 8-2 /etc/lp ディレクトリの内容

ファイル	種類	説明
classes	ディレクトリ	lpadmin -c コマンドで与えたクラスを識別するファイル
fd	ディレクトリ	既存のフィルタの記述
filter.table	ファイル	印刷フィルタ照合テーブル
forms	ディレクトリ	各フォームのファイルを格納する場所。最初は、このディレクトリは空になっている
interfaces	ディレクトリ	プリンタインタフェースプログラムファイル
logs	/var/lp/logs へのリンク	印刷動作のログファイル
model	/usr/lib/lp/model へのリンク	標準プリンタインタフェースプログラムのリンク
printers	ディレクトリ	各ローカルプリンタのディレクトリ。各ディレクトリには、個々のプリンタの構成情報と警告ファイルが入っている
pwheels	ディレクトリ	印字ホイールまたはカートリッジファイル

これらの構成ファイルは、lpd ベースのプリンタサーバー上の /etc/printcap ファイルと同じ機能を提供します。

注 - これらのファイルの内容を確認できますが、ファイルを直接編集しないでください。代わりに、lpadmin コマンドを使用して構成を変更します。変更内容は /etc/lp ディレクトリ内の構成ファイルに入力されます。lpsched デーモンは、構成ファイルを管理して構成します。

/etc/lp/printers ディレクトリには、システムに認識されるローカルプリンタごとに1つずつサブディレクトリが入っています。次の例は、プリンタ sparc1 と luna の /etc/lp/printers サブディレクトリを示します。

```
$ ls -l /etc/lp/printers
drwxrwxr-x 2 lp lp 512 Jan 23 23:53 luna
drwxrwxr-x 2 lp lp 512 Jan 11 17:50 sparc1
```

プリンタ固有の各ディレクトリ内では、次のファイルを使用してプリンタを記述できます。

ファイル名	説明
alert.sh	警告に応答して実行するシェル
alert.vars	警告変数
configuration	構成ファイル
users.deny	プリンタアクセスが拒否されるユーザーのリスト
comment	プリンタの説明 (備考欄)

通常、プリンタ luna の構成ファイル /etc/lp/printers/luna/configuration は、次のようになっています。

```
Banner: on: Always
Content types: PS
Device: /dev/term/b
Interface: /usr/lib/lp/model/standard
Printer type: PS
Modules: default
```

terminfo データベース

/usr/share/lib ディレクトリには、terminfo データベースのディレクトリが入っており、そのディレクトリには多数のタイプの端末とプリンタに関する定義が入っています。LP 印刷サービスは、terminfo データベース内の情報を使用してプリンタを初期設定し、選択されたページサイズ、文字ピッチ、行ピッチ、および文字セットを設定し、一連のコードをプリンタに送ります。

各プリンタは、terminfo データベース内で短縮名を使用して識別されます。terminfo データベースの構造については、51 ページの「プリンタタイプ」を参照してください。必要であれば、terminfo データベースにエントリを追加できますが、これは煩雑で時間のかかる作業です。詳細は、147 ページの「サポートされていないプリンタの terminfo エントリを追加する」を参照してください。

デーモンと LP 内部ファイル

/usr/lib/lp ディレクトリには、表 8-3 に示すような LP 印刷サービスに使用されるデーモンとファイルが入っています。

表 8-3 /usr/lib/lp ディレクトリの内容

ファイル	種類	説明
bin	ディレクトリ	印刷警告、低速フィルタ、待ち行列管理プログラムを生成するファイルが入っている
lpsched	デーモン	LP 印刷要求のスケジューリングを管理する
model	ディレクトリ	標準プリンタインタフェースプログラムが入っている
postscript	ディレクトリ	LP 印刷サービスで提供されているすべての PostScript フィルタプログラムが入っている。これらのフィルタには、フィルタの特性とその格納場所を LP 印刷サービスに指示する /etc/lp/fd ディレクトリ内の記述子ファイルが含まれる

LP 印刷サービスのログファイル

LP 印刷サービスは、次の 2 組のログファイルを管理します。

ログファイル名	説明
syslogd (1M)	/etc/syslog.conf の lpr.debug を設定して LP 印刷サービスロギングを有効にする

ログファイル名	説明
<code>/var/spool/lp</code>	印刷待ち行列に入っている現在の待ち行列のリスト
<code>/var/lp/logs/requests</code>	進行中の印刷要求の履歴

印刷待ち行列ログ

各システムのスケジューラは、`/var/spool/lp/tmp/system` ディレクトリと `/var/spool/lp/requests/system` ディレクトリ内で印刷要求ログを保管します。各印刷要求は、要求に関する情報を含んだファイルを2つ(各ディレクトリに1つずつ)持っています。`/var/spool/lp/requests/system` ディレクトリ内の情報には、スーパーユーザーまたは `lp` しかアクセスできません。`/var/spool/lp/tmp/system` ディレクトリ内の情報には、その要求を出したユーザー、スーパーユーザー、または `lp` しかアクセスできません。

次の例では、`/var/spool/lp/tmp/starbug` ディレクトリの内容を示します。

```
$ ls /var/spool/lp/tmp/starbug
5      5-0
# cat 5-0
C 1
D print1
F /etc/profile
P 20
T /etc/profile
t simple
U root
s 0000
v 2
```

これらのファイルは、印刷要求が待ち行列に入っている限り、そのディレクトリ内に残っています。要求が完了すると、ファイル内の情報は組み合わせられ、`/var/lp/logs/requests` ファイルに追加されます。このファイルについては、次の節で説明します。

現在待ち行列に入っている印刷要求の状態を追跡したい場合は、`/var/spool/lp/logs` ディレクトリの情報を使用します。

履歴ログ

LP 印刷サービスは、`lpsched` と `requests` という2つのログファイルに印刷サービスの履歴を記録します。これらのログファイルは、`/var/lp/logs` ディレクトリに入っています。これらのログ内の情報を使用し、印刷の問題を診断して解決できます。次の例は、`/var/lp/logs` ディレクトリの内容を示します。

```
# cd /var/lp/logs
# ls
lpsched.1      requests      requests.2
```



```
lpsched      lpsched.2  requests.1
#
```

問題の解決に最も重要な2つのログファイルは、ローカルの印刷要求に関する情報が入っている `lpsched` ログと、すでに完了してプリンタの待ち行列にない印刷要求の情報が入っている `requests` ログです。

`requests` ログの構造は単純なので、共通の UNIX シェルコマンドを使用してデータを抽出できます。要求は、出力された順番に要求 ID を示す行で区切って表示されます。= で始まる区切り行より下の各行には、その行に含まれる情報の種類を識別する1つの文字が付いています。各文字は、空白文字1つでデータと区切られます。

次の例は、`requests` ログの内容を示しています。

```
# pwd
/var/lp/logs
# tail requests.2
= print1-3, uid 0, gid 1, size 206662, Wed Mar 14 08:56:30 MST 2003
z print1
C 1
D print1
F /usr/dict/words
P 20
T /usr/dict/words
t simple
U root
s 0x0014
v 2
#
```

表 8-4 に、文字コードとそれに対応する LP `requests` ログ内の行を示します。

表 8-4 LP `requests` ログ内の文字コード

文字	行の内容
=	区切り行。ユーザーの要求 ID、ユーザー ID (UID)、グループ ID (GID)、フィルタを通す前のファイルサイズの総バイト数、および要求が待ち行列に入れられた時刻が入っている
z	プリンタ名
C	印刷部数
D	出力先のプリンタクラス、または any
F	印刷されたファイル名。この行は印刷されたファイルごとに区切られ、ファイルが印刷された順番で表示される
f	(省略可能) 使用するフォーム名。
H	(省略可能) 特殊処理。resume、hold、または immediate

表 8-4 LP requests ログ内の文字コード (続き)

文字	行の内容
N	(省略可能) 印刷要求が正常に完了したときに使用された警告のタイプ。ユーザーが電子メールで通知を受けたときは m タイプ、端末へのメッセージで通知を受けた場合は w タイプ
O	(省略可能) プリンタ依存の -o オプション。たとえば、nobanner
P	印刷要求の優先順位
p	印刷されたページのリスト
r	(省略可能) ユーザーが lp -r コマンドを使用してファイルの「raw」処理を要求したときに含まれる 1 文字の行
S	(省略可能) 使用された文字セット、印字ホイール、またはカートリッジ
T	バナーページに印刷されるタイトル
t	ファイル内で見つかった内容形式
U	印刷要求を出したユーザー名
s	16 進数形式の各ビットの組み合わせで表される要求の結果。印刷サービスの内部では複数のビットが使用される。ビットとその意味については、下表に記載
x	(省略可能) 印刷要求に使用された低速フィルタ
Y	(省略可能) 要求の印刷に使用された印刷フィルタの特殊モードのリスト
z	(省略可能) 要求に使用したプリンタ。要求がプリンタまたはプリンタクラスの待ち行列に入れられた場合や、要求が別の出力先に移動された場合は、このプリンタは出力先 (d 行) とは異なる

次の表に、LP requests ログ内の結果コードとその説明を示します。

表 8-5 LP requests ログ内の結果コード

結果コード	説明
0x0001	要求は保留され再開を待機中
0x0002	低速フィルタを実行中
0x0004	低速フィルタを正常に完了
0x0008	要求はプリンタ上にある
0x0010	印刷を正常に完了
0x0020	要求は保留されユーザーによる変更を待機中
0x0040	要求は取り消し済み

表 8-5 LP requests ログ内の結果コード (続き)

結果コード	説明
0x0080	要求は次に印刷される
0x0100	フィルタ処理または印刷要求に失敗
0x0200	要求はリモートプリンタに転送中 (現在は使用されない)
0x0400	ユーザーに通知
0x0800	通知が動作中
0x1000	リモートシステムが要求を受け付け済み (現在は使用されない)
0x2000	管理者が要求を保留した
0x4000	プリンタのフィルタを変更しなければならなかった
0x8000	要求は一時的に停止された

スプーリングディレクトリ

印刷待ち行列に入れられたファイルは、印刷されるまで `/var/spool/lp` ディレクトリに格納されますが、それがわずかに数秒の場合があります。表 8-6 は、`/var/spool/lp` ディレクトリの内容を示します。

表 8-6 `/var/spool/lp` ディレクトリの内容

ファイル	種類	説明
SCHEDLOCK	ファイル	スケジューラのロックファイル。スケジューラが停止し、再起動されない場合は、このファイルをチェックする
admins	ディレクトリ	<code>/etc/lp</code> へのリンク
bin	ディレクトリ	<code>/usr/lib/lp/bin</code> へのリンク
logs	リンク	完了した印刷要求のログが記録される <code>../lp/logs</code> へのリンク
model	リンク	<code>/usr/lib/lp/model</code> へのリンク
requests	ディレクトリ	印刷要求が印刷されるまで記録される構成済みプリンタごとのサブディレクトリが入ったディレクトリ。ユーザーはこのログにアクセスできない
system	ディレクトリ	システムの印刷状態ファイル
temp	リンク	スプールされた要求が入っている <code>/var/spool/lp/tmp/hostname</code> へのリンク

表 8-6 /var/spool/lp ディレクトリの内容 (続き)

ファイル	種類	説明
tmp	ディレクトリ	印刷要求が印刷されるまでログが記録される構成済みの各プリンタのディレクトリ。既存の印刷要求を変更した場合も、このログに記録される

LP 印刷サービスのコマンド

表 8-7 に、頻繁に使用する LP 印刷サービスのコマンドを示します。(1M) のコマンドを使用するには、スーパーユーザーまたは lp になるか、同等の役割になる必要があります。

表 8-7 LP 印刷サービスコマンド早見表

コマンド	マニュアルページ	作業
enable	enable (1)	プリンタを使用可能にする
cancel	cancel (1)	印刷要求を取り消す
lp	lp (1)	1 つ以上のファイルをプリンタに送る
lpstat	lpstat (1)	LP 印刷サービスの状態を出力する
disable	enable (1)	1 台以上のプリンタを無効にする
accept	accept (1M)	印刷要求を特定の出力先の待ち行列に入れられるようにする
reject	accept (1M)	印刷要求が特定の出力先の待ち行列に入れられないようにする
lpadmin	lpadmin (1M)	プリンタの構成を設定または変更する
lpfilter	lpfilter (1M)	フィルタの定義を設定または変更する
lpforms	lpforms (1M)	あらかじめ印刷されたフォームを設定または変更する
lpadmin	lpadmin (1M)	フォームを取り付ける
lpmove	lpmove (1M)	ある出力先から別の出力先に出力要求を移動する
lpsched	lpsched(1M)	LP 印刷サービススケジューラを起動する
lpshut	lpshut (1M)	LP 印刷サービススケジューラを停止する
lpusers	lpusers (1M)	デフォルトの優先順位と、LP 印刷サービスのユーザーが要求できる優先順位の制限を設定または変更する

LP 印刷サービスの機能

LP 印刷サービスは、次の機能を実行します。

- ファイルを管理してローカル印刷要求をスケジュールする。
- ネットワーク要求を受け取り、スケジュールする。
- ファイルが正しく印刷されるように、必要に応じてフィルタを通す。
- プリンタとインタフェースするプログラムを起動する。
- ジョブの状態を追跡する。
- プリンタに取り付けられたフォームを追跡する。
- 現在装着されている印字ホイールを追跡する。
- 新しいフォームや別の印字ホイールを取り付け、装着するよう警告を発する。
- 印刷問題に関する警告を発する。

LP によるファイルの管理とローカル印刷要求のスケジュールリングの方法

LP 印刷サービスには、`lpsched` というスケジューラデーモンが組み込まれています。スケジューラデーモンは、プリンタの設定と構成に関する情報を使用して LP システムファイルを更新します。

また、`lpsched` デーモンは、ユーザーが要求をアプリケーションから出すかコマンド行から出すかに関係なく、[図 8-1](#) のように、プリンタサーバー上のすべてのローカル印刷要求をスケジュールします。さらに、スケジューラはプリンタとフィルタの状態を追跡します。プリンタが要求を印刷し終わると、プリンタサーバー上の待ち行列に残っているものがあれば、スケジューラは次の要求をスケジュールします。

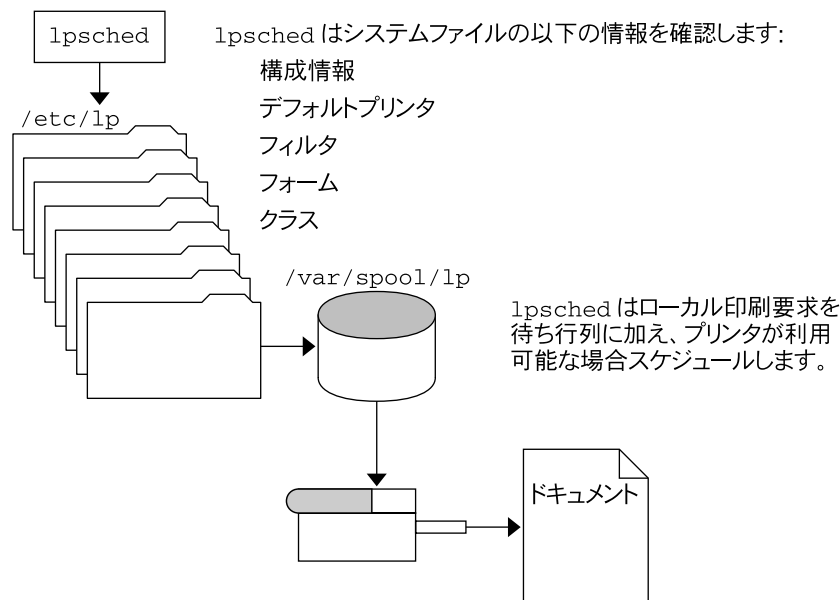


図 8-1 lpsched デーモンによるローカル印刷要求のスケジューリング方法

各プリンタサーバーは、LP スケジューラを 1 つだけ実行していなければなりません。スケジューラは、システムのブート時に (または実行レベル 2 を入力すると)、制御スクリプト `/etc/rc2.d/S80lp` によって起動されます。システムを再起動しなくても、`/etc/init.d/lp stop` コマンドを使用してスケジューラを停止し、`/etc/init.d/lp start` コマンドを使用して再起動できます。各システムのスケジューラは、`lp` コマンドによってシステムに出された要求を管理します。

ネットワーク印刷要求のスケジューリング

各印刷クライアントは、ネットワーク経由で直接、プリンタサーバーと通信します。通信は、要求コマンド (`lp`、`lpstat`、`cancel`、`lpr`、`lpq`、または `lprm`) とプリンタサーバー上の印刷サービス間で行われます。これによって、クライアント専用システムにおける印刷システムのオーバーヘッドが軽減され、その結果、拡張性、パフォーマンス、およびデータの正確性が向上します。

プリンタサーバーは、現在、インターネットサービスデーモン (`inetd`) で印刷要求を待機します。ネットワークから印刷サービスへ要求があると、`inetd` デーモンは「プロトコルアダプタ」と呼ばれるプログラム (`in.lpd`) を起動します。プロトコルアダプタは、印刷要求を翻訳して、印刷スプーラに送信し、その結果を要求依頼元に戻します。このプロトコルアダプタは、要求の発生時に起動し、ネットワーク要求の処理が完了すると終了します。このプロセスによって、印刷のためのアイドル状態のシステムのオーバーヘッドが解消されます。また、Solaris の以前の印刷機能にあった、ネットワークに接続された印刷サポート用の余分なシステム構成が不要になります。

印刷ファイルにフィルタを適用する

印刷フィルタは、待ち行列内のファイルの内容をある形式から別の形式に変換するプリンタサーバー上のプログラムです。

印刷フィルタは、必要に応じて簡単なものでも複雑なものでもかまいません。SunOS リリースの場合は、出力先プリンタが PostScript 形式へのデータ変換を必要とするほとんどの PostScript 印刷状況に対応する印刷フィルタが、`/usr/lib/lp/postscript` ディレクトリに組み込まれています。PostScript 以外のプリンタ用のフィルタが必要な場合は、そのフィルタを作成し、目的のシステムに追加しなければなりません。

一連の「印刷フィルタ記述子ファイル」が `/etc/lp/fd` ディレクトリに用意されています。これらの記述子ファイルは、フィルタの特性 (高速フィルタや低速フィルタなど) を記述し、フィルタプログラム (`/usr/lib/lp/postscript/postdaisy` など) を指します。

プリンタインタフェースプログラムの機能

LP 印刷サービスは、オペレーティングシステムの他の部分と情報をやり取りします。また、標準プリンタインタフェースプログラムを使用して、次の作業を実行します。

- 必要に応じてプリンタポートを初期化する。標準プリンタインタフェースプログラムは、`stty` コマンドを使用してプリンタポートを初期化する。
- プリンタを初期化する。標準プリンタインタフェースプログラムは、`terminfo` データベースと `TERM` シェル変数を使用して、適切な制御シーケンスを見つける。
- 必要に応じてバナーページを印刷する。
- 印刷要求で指定された部数だけ印刷する。

LP 印刷サービスは、別のプログラムが指定されなければ、標準インタフェースプログラム (`/usr/lib/lp/model` ディレクトリに入っています) を使用します。独自のインタフェースプログラムを作成することもできますが、独自のプログラムがプリンタへの接続を終了させないことや正しいプリンタの初期設定を妨げないことを確認しなければなりません。

lp sched デーモンによる印刷ジョブ状態の確認

lp sched デーモンは、処理する各印刷要求のログを保持し、印刷処理中に発生したエラーを記録します。このログは `/var/lp/logs/lpsched` ファイルに保管されます。lp cron ジョブは毎晩、`/var/lp/logs/lpsched` ファイルの名前を新しい `lpsched.n` という名前に変更して、新しいログファイルを開始します。エラーが発生したり、印刷待ち行列からジョブが消えたりした場合は、ログファイルを使用して lp sched デーモンで実行された印刷ジョブへの処理を判別できます。

ログファイルの消去

/var/lp/logs ディレクトリ内の lpsched および requests ログファイルは、情報が追加されるにつれて大きくなります。LP 印刷サービスは、デフォルトの cron ジョブを使用してログファイルを消去します。lp cron ジョブは /var/spool/cron/crontabs/lp ファイルに入っています。このジョブはログファイルの内容を定期的に移動します。ログの内容は log.1 に移動され、log.1 の内容は log.2 に移動されるというようになります。log.2 が上書きされると、その内容は失われます(つまり、前の log.1 の内容が代わりに書き込まれます)。

▼ プリンタ要求のログの交換間隔を変更する方法

プリンタサーバー上の requests ログファイルは、毎日ではなく、毎週交換されます。プリンタサーバーの使用頻度が高い場合は、交換間隔を毎日に再設定できます。

1. プリンタサーバー上でスーパーユーザーまたは lp になります。
2. EDITOR 環境変数を設定します。

```
# EDITOR=vi
# export EDITOR
```

3. lp の crontab ファイルを編集します。

```
# crontab -e lp
```

4. requests ログファイルの交換間隔を指定するファイルの先頭行を、毎日曜日 (0) から、毎日を示すアスタリスク (*) に変更します。

```
13 3 * * * cd /var/lp/logs; if [ -f requests ]; then if
[ -f requests.1 ]; then /bin/mv requests.1 requests.2; fi; /usr/bin/cp
requests requests.1; >requests; fi
```

5. ファイルを保存して、終了します。

ローカル印刷の処理スケジュール

印刷要求は、クライアントとサーバーが異なるシステムにある場合と同じ経路をたどります。要求は常に同じ経路をたどり、クライアントからサーバーに流れます。

次の図に、ユーザーがローカルプリンタ上に PostScript ファイルを印刷する要求を出したときに実行される処理を示します。ローカルプリンタとは、ユーザーのシステムに接続されたプリンタです。ローカルシステム上の lpsched デーモンがすべての処理を実行します。これらの処理には、プリンタと内容形式の照合、デフォルトプリンタの特定、要求のフィルタ処理、指定されたプリンタインタフェースプログラムの起動などが含まれます。

プリンタインタフェースプログラムは、次のことを行います。

- パナーページの印刷
- プリンタ障害の検出
- プリンタ障害ポリシーを、リセット、継続、または印刷ジョブのやり直しのどれかに決定する
- 最後に、インタフェースプログラムは `lpcat` プログラムを使用して、印刷要求をローカルプリンタのシリアルポートにダウンロードする

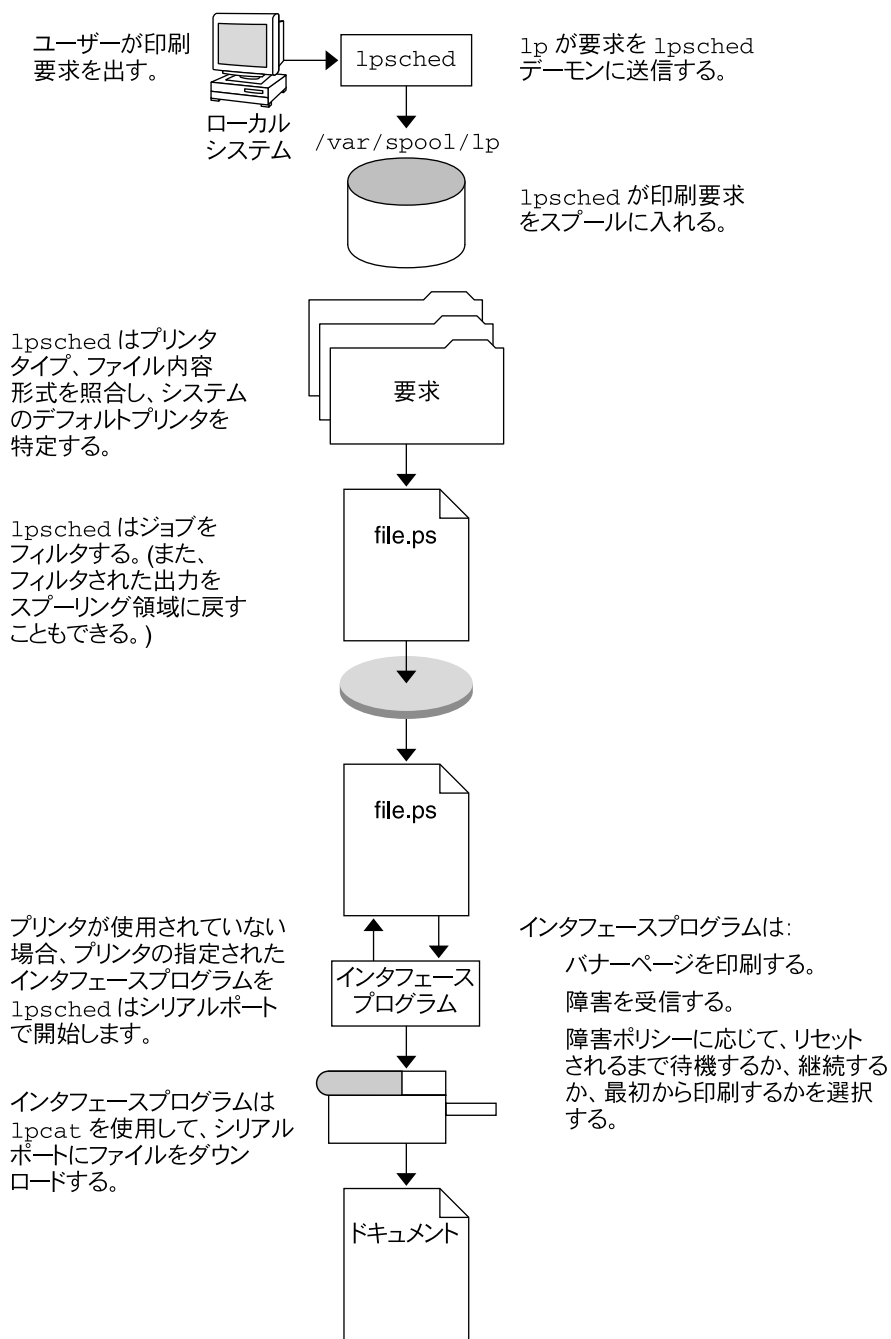


図 8-2 ローカルでの印刷処理

リモート印刷の処理スケジュール

次の図は、Solaris 印刷クライアントのユーザーが lpd ベースのプリンタサーバーに印刷要求を出したときに実行される処理を示します。このコマンドは、直接プリンタサーバーと接続して、自身の通信を処理します。

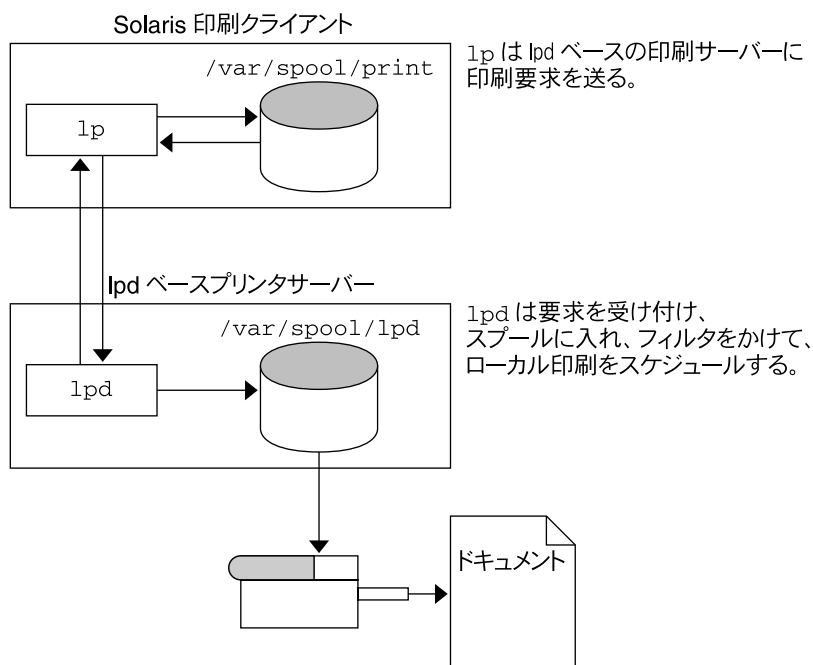


図 8-3 Solaris 印刷クライアントと lpd ベースのプリンタサーバー間の印刷

次の図は、lpd ベースの印刷クライアントが Solaris プリンタサーバーに印刷要求を出す様子を示しています。lpd デーモンは、印刷要求のローカル部分とプリンタサーバーへの接続を処理します。プリンタサーバー上の inetd プロセスは、ネットワーク印刷要求を待って、プロトコルアダプタを起動して要求を処理します。プロトコルアダプタは lpsched デーモンと通信し、このデーモンがプリンタサーバー上で要求を処理します。

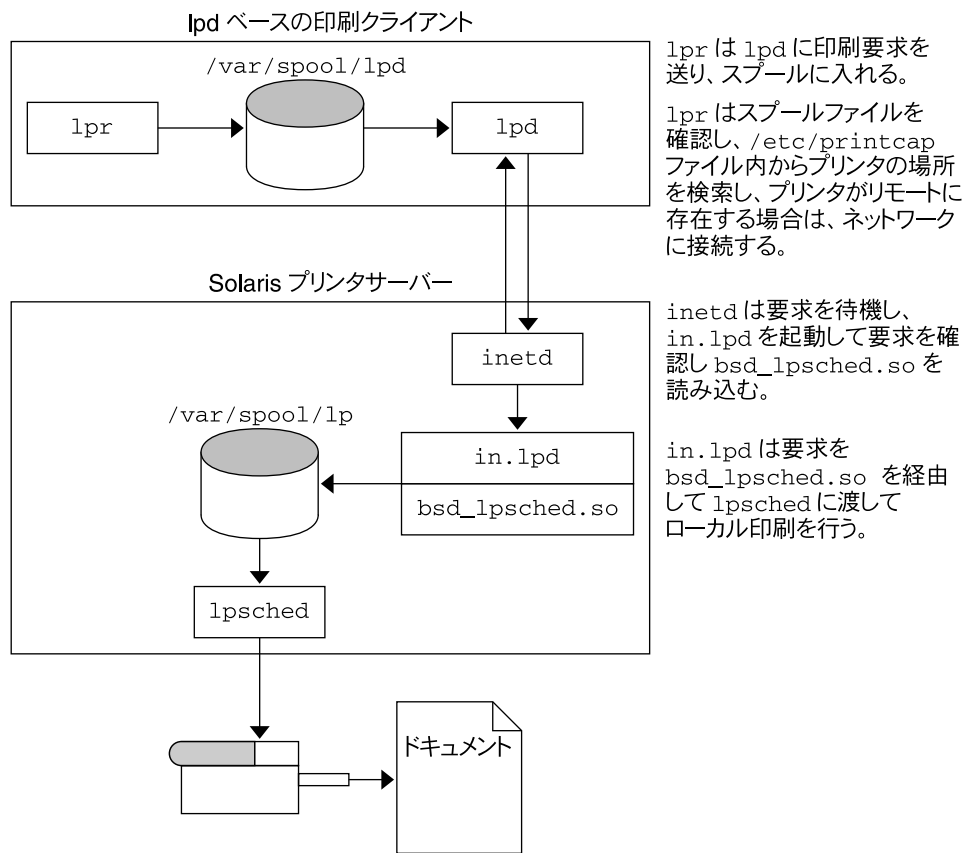


図 8-4 lpd ベースの印刷クライアントと Solaris プリンタサーバー間の印刷

次の図は、Solaris 印刷クライアントのユーザーが Solaris プリンタサーバーに印刷要求を出したときに実行される処理を示します。印刷クライアント上の印刷コマンドは、プリンタサーバーと直接通信することにより、各印刷要求のローカル部分を処理します。

プリンタサーバー上の `inetd` プロセスは、ネットワーク上の印刷要求を監視し、プロトコルアダプタを起動して、プリンタサーバー上の `lpsched` デーモンと通信します。このデーモンが印刷要求を処理します。

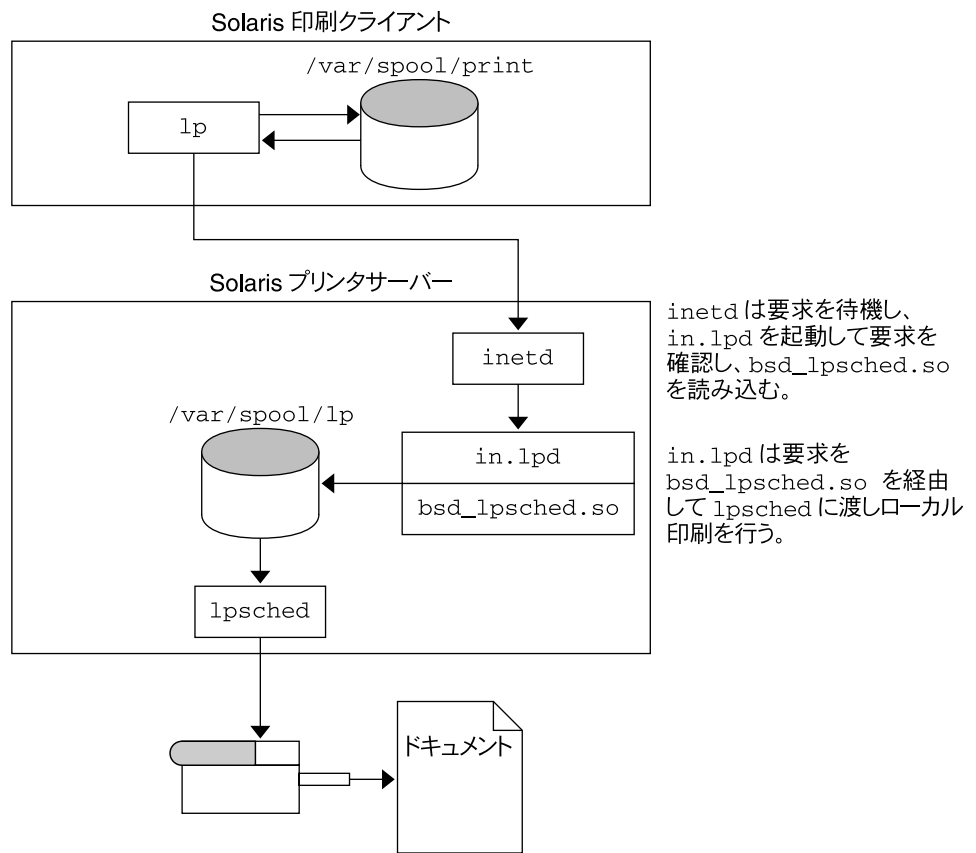


図 8-5 Solaris 印刷クライアントと Solaris プリンタサーバー間の印刷

第 9 章

端末とモデムの管理 (トピック)

以下の各章で、端末とモデムの管理について説明します。

第 10 章	端末とモデムの概要について説明します。
第 11 章	端末とモデムを設定する手順について説明します。
第 12 章	SAF コマンドを使用して端末とモデムを設定する手順について説明します。

第 10 章

端末とモデムの管理 (概要)

この章では、端末やモデムを管理する場合の概要を説明します。この章の内容は以下のとおりです。

- 193 ページの「端末、モデム、ポート、サービス」
- 196 ページの「端末とモデムを管理するツール」
- 196 ページの「シリアルポートツール」
- 196 ページの「サービスアクセス機能 (SAF)」

シリアルポートツールで端末とモデムを設定する手順については、[第 11 章](#)を参照してください。

SAF で端末とモデムを設定する手順については、[第 12 章](#)を参照してください。

端末とモデムの管理に関する新機能

Solaris 管理コンソールは、端末とモデムを設定するためのシリアルポートツールを提供します。Solaris 管理コンソールの起動については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。端末やモデムを設定する手順については、Solaris 管理コンソールのオンラインヘルプを参照してください。

端末、モデム、ポート、サービス

端末とモデムは、システム資源とネットワーク資源へのローカルおよびリモートのアクセスを提供します。端末とモデムの設定は、システム管理者の重要な作業です。この節では、Solaris 環境におけるモデムと端末の管理についての概要を説明します。

端末

システムのビットマップグラフィックスディスプレイは、シリアルポートに接続され、テキストしか表示できない英数字端末とは異なります。グラフィックスディスプレイは、特別な手順に従って管理する必要はありません。

モデム

モデムには次の3つの基本構成があります。

- 発信専用
- 着信専用
- 発着信両用

家庭用コンピュータに接続されるモデムの中には、「発信専用」サービス向けに設定されていることがあります。その場合、ユーザーは家から他のコンピュータにアクセスできますが、外からは誰もユーザーのコンピュータにアクセスできません。

「着信専用」サービスは発信専用のちょうど逆です。つまり、リモートサイトからはシステムにアクセスできますが、そのシステムから外側には呼び出しができません。

「発着信両用」アクセスは、その名前が示すとおり、着信専用、発信専用の両機能を持っています。

ポート

「ポート」とは、装置がオペレーティングシステムと通信するためのチャンネルのことです。具体的には、端末やモデムのケーブルを物理的に接続する「コンセント」と考えると一番わかりやすいでしょう。

ただし、ポートは厳密には物理的なコンセントではなく、その実体はハードウェア（ピンとコネクタ）とソフトウェア（デバイスドライバ）からなっています。多くの場合、1つの物理的なコンセントが複数のポートを備えており、複数の装置を接続できます。

一般的なポートとして、シリアル、パラレル、SCSI (Small Computer Systems Interface)、Ethernet などがあります。

「シリアルポート」は、標準の通信プロトコルを使用し、1本の信号線で1バイト単位の情報を1ビットずつ送信します。

RS-232-C または RS-423 規格に準拠して設計されている装置（大部分のモデム、英数字端末、プロッタ、一部のプリンタ）は、同様に設計されたコンピュータのシリアルポートにはどれにでも、標準ケーブルを使用して接続できます。

1 台のコンピュータに多数のシリアルポートを接続する場合は、システムに「アダプタボード」を追加する必要があります。アダプタボードは、ドライバソフトウェアを使用することにより、より多くの装置を接続できるための追加のシリアルポートを提供します。

サービス

モデムや端末を使用すると、シリアルポートのソフトウェアを介してコンピュータ資源にアクセスできます。シリアルポートソフトウェアは、ポートに接続する装置向けに特定の「サービス」を提供するように設定しなければなりません。たとえば、モデムに対してはシリアルポートは発着信両用サービスを提供するように構成できます。

ポートモニター

特定のサービスへのアクセスは、主に「ポートモニター」を通じて行います。ポートモニターとは、ログイン要求や、プリンタまたはファイルのアクセス要求を常に監視しているプログラムのことです。

ポートモニター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装置間の通信を確立するのに必要なすべてのパラメータを設定します。次に、必要なサービスを提供する他のプロセスに制御を移します。

表 10-1 に、Solaris 環境で提供されている 2 つのタイプのポートモニターとその説明を示します。

表 10-1 ポートモニターのタイプ

ポートモニター (Port Monitor)	マニュアルページ	説明
listen	listen (1M)	Solaris 2.6 より前のシステムからのリモート印刷要求の処理など、ネットワークサービスへのアクセスを制御する。デフォルトの Solaris オペレーティング環境では、このタイプのポートモニターは使用されていない。
ttymon	ttymon (1M)	モデムや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する。シリアルポートツールは、それらの装置からのログイン要求を処理するように、ttymon ポートモニターを自動的に設定する。

getty (1M) という従来のポートモニターに慣れているユーザーも、新しい ttymon を使用してください。新しい ttymon はさらに強力なツールとなっています。1 つの ttymon で複数の getty に相当する処理が行えます。それ以外の点では、どちらのプログラムも同じ機能を提供します。

端末とモデムを管理するツール

次の表は、端末とモデムを管理するツールを列挙したものです。

表 10-2 端末とモデムを管理するツール

ツールの説明	ツール	参照箇所
管理作業全般	サービスアクセス機能 (SAF) のコマンド	196 ページの「サービスアクセス機能 (SAF)」
簡易設定	Solaris 管理コンソールのシリアルポートツール	第 11 章と Solaris 管理コンソールオンラインヘルプ
簡易設定	Admintool	Admintool オンラインヘルプ

シリアルポートツール

シリアルポートツールは、`pmadm` コマンドを呼び出すことにより、シリアルポートソフトウェアを設定して端末やモデムを管理します。シリアルポートツールはまた、次の機能も提供します。

- 共通の端末およびモデム構成用テンプレート
- 複数ポートの設定、変更、または削除
- 各ポートの状態の簡易表示

サービスアクセス機能 (SAF)

SAF は、端末、モデム、その他のネットワーク装置の管理用のツールです。SAF では特に次の設定を行います。

- `ttymon` および `listen` ポートモニター (`sacadm` コマンドを使用)
- `ttymon` ポートモニターサービス (`pmadm`、`ttyadm` コマンドを使用)
- `listen` ポートモニターサービス (`pmadm`、`nlsadmin` コマンドを使用)
- `tty` 装置に関する問題の解決
- ネットワークからの印刷サービス要求に関する問題の解決
- サービスアクセスコントローラに関する問題の解決 (`sacadm` コマンドを使用)

SAF は、`tty` 装置やローカルエリアネットワーク (LAN) を通して行われるシステム資源やネットワーク資源へのアクセスを制御するオープンシステムソリューションです。SAF はプログラムではなく、バックグラウンドプロセスと管理用コマンドの階層構造になっています。

第 11 章

端末とモデムの設定 (手順)

この章では、Solaris 管理コンソールのシリアルポートツールを使用して、端末とモデムを設定する手順を説明します。

この章で説明する手順は次のとおりです。

- 200 ページの「端末を設定する方法」
- 201 ページの「モデムを設定する方法」
- 202 ページの「ポートを初期化する方法」

端末とモデムの概要については、第 10 章を参照してください。

シリアルポートツールによる端末とモデムの設定

Solaris 管理コンソールのシリアルポートツールを使用すると、シリアルポートを設定できます。「シリアルポート (Serial Ports)」メニューからシリアルポートを選択し、「アクション (Action)」メニューから「構成 (Configure)」オプションを選択して、次の内容を設定します。

- 端末
- モデム - 着信
- モデム - 発信
- モデム - 着信 / 発信
- 初期化のみ - 接続なし

「構成 (Configure)」オプションは、上記サービスを設定するテンプレートへのアクセスを提供します。各シリアルポートの詳細は 2 つのレベルで表示できます。つまり、基本と拡張機能です。シリアルポートを選択して、「アクション (Action)」メニューから「プロパティ (Properties)」オプションを選択し、シリアルポートを構成し

た後、各シリアルポートの詳細レベルにアクセスすることができます。シリアルポートを構成したら、SAF コマンドでポートを使用可能または使用不可にすることができます。SAF コマンドの使用については、第 12 章を参照してください。

シリアルポートのコマンド行インタフェースについては、`smserialport (1M)` のマニュアルページを参照してください。

端末の設定

表 11-1 にシリアルポートを使用して端末を設定する際のメニュー項目 (およびそれらのデフォルト値) を示します。

表 11-1 端末のデフォルト値

認定レベル	項目	デフォルト値
基本 (Basic)	ポート名 (Port Name)	—
	備考欄 (Description)	Terminal
	サービスの状態 (Service Status)	使用可能
	ボーレート (Baud Rate)	9600
	端末の種類 (Terminal Type)	tvi925
	ログインプロンプト (Login Prompt)	ttyn login:
拡張機能 (Advanced)	キャリア検出 (Carrier Detection)	ソフトウェア
	選択: キャリア検出時に接続 (Connect on Carrier)	いいえ
	選択: 双方向 (Bidirectional)	はい
	選択: 初期化のみ (Initialize Only)	いいえ
	タイムアウト (秒) (Timeout)	なし
	ポートモニター (Port Monitor)	zsmon
	サービスプログラム (Service Program)	/usr/bin/login

モデムの設定

表 11-2 に、シリアルポートでモデムを設定する際に使用できる 3 つのモデム用テンプレートを示します。

表 11-2 モデム用テンプレート

モデム構成	説明
着信専用	モデムに着信はできるが、発信はできない。
発信専用	モデムから発信はできるが、着信はできない。
発着信両用	モデムへ着信も、モデムから発信もできる。

表 11-3 に各テンプレートのデフォルト値を示します。

表 11-3 モデム用テンプレートのデフォルト値

認定レベル	項目	モデム - 着信専用	モデム - 発信専用	モデム - 着信と発信
基本 (Basic)	ポート名 (Port Name)	—	—	—
	備考欄 (Description)	Modem – Dial In Only	Modem – Dial Out Only	Modem – Dial In and Out
	サービスの状態 (Service Status)	使用可能	使用可能	使用可能
	ボーレート (Baud Rate)	9600	9600	9600
	ログインプロンプト (Login Prompt)	ttyn login:	ttyn login:	ttyn login:
拡張機能 (Advanced)	キャリア検出 (Carrier Detection)	ソフトウェア	ソフトウェア	ソフトウェア
	選択: キャリア検出時に接続 (Connect on Carrier)	いいえ	いいえ	いいえ
	選択: 双方向 (Bidirectional)	いいえ	いいえ	はい
	選択: 初期化のみ (Initialize Only)	いいえ	はい	いいえ
	タイムアウト (秒) (Timeout)	なし	なし	なし
	ポートモニター (Port Monitor)	zsmon	zsmon	zsmon
	サービスプログラム (Service Program)	/usr/bin/login	/usr/bin/login	/usr/bin/login

表 11-4 では、「初期化のみ」テンプレートの各デフォルト値を示します。

表 11-4 「初期化のみ (Initialize Only)」のデフォルト値

認定レベル	項目	デフォルト値
基本 (Basic)	ポート名 (Port Name)	—
	備考欄 (Description)	Initialize Only - No Connection
	サービスの状態 (Service Status)	有効
	ボーレート (Baud Rate)	9600
	ログインプロンプト (Login Prompt)	ttyn login:
拡張機能 (Advanced)	キャリア検出 (Carrier Detection)	ソフトウェア
	選択: キャリア検出時に接続 (Connect on Carrier)	いいえ
	選択: 双方向 (Bidirectional)	できる
	選択: 初期化のみ (Initialize Only)	できる
	タイムアウト (秒) (Timeout)	なし
	ポートモニター (Port Monitor)	zsmon
	サービスプログラム (Service Program)	/usr/bin/login

▼ 端末を設定する方法

1. Solaris 管理コンソールが実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

2. ナビゲーション区画で「コンピュータ (Computer)」アイコンをクリックします。
3. 「デバイスとハードウェア (Devices and Hardware)」、「シリアルポート (Serial Ports)」をクリックします。
シリアルポートメニューが表示されます。
4. 端末に使用するポートを選択します。
5. 「アクション (Action)」メニューから「構成 (Configure)」、「端末 (Terminal)」を選択します。
「シリアルポート n を構成 (Configure Serial Port)」ウィンドウが「基本 (Basic)」モードで表示されます。
「端末 (Terminal)」メニュー項目については、表 11-1 を参照してください。

6. 「了解 (OK)」をクリックします。
7. 各項目を設定するには、端末として設定されたポートを選択します。次に、「アクション (Action)」メニューから「プロパティ (Properties)」を選択します。
8. 必要な場合は、テンプレートエントリの値を変更します。
9. 「了解 (OK)」をクリックしてポートを設定します。
10. 端末サービスが追加されていることを確認します。

```
$ pmadm -l -s tty#
```

▼ モデムを設定する方法

1. Solaris 管理コンソールが実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

2. ナビゲーション区画で「コンピュータ (Computer)」アイコンをクリックします。
3. 「デバイスとハードウェア (Devices and Hardware)」、「シリアルポート (Serial Ports)」をクリックします。
シリアルポートメニューが表示されます。
4. モデムに使用するポートを選択します。
5. 「アクション (Action)」メニューから、次の「構成 (Configure)」オプションのいずれかを選択します。
 - a. 「構成 (Configure)」、「モデム (着信) (Modem (Dial In))」
 - b. 「構成 (Configure)」、「モデム (発信) (Modem (Dial Out))」
 - c. 「構成 (Configure)」、「モデム (着信/発信) (Modem (Dial In/Out))」
「シリアルポート n を構成 (Configure Serial Port)」ウィンドウが「基本 (Basic)」モードで表示されます。
「モデム (Modem)」メニュー項目については、表 11-3 を参照してください。
6. 「了解 (OK)」をクリックします。
7. 各項目を設定するには、モデムとして設定されたポートを選択します。次に、「アクション (Action)」メニューから「プロパティ (Properties)」を選択します。
8. 必要な場合は、テンプレートエントリの値を変更します。
9. 「了解 (OK)」をクリックしてポートを設定します。

10. モデムのサービスが設定されたことを確認します。

```
$ pmadm -l -s ttyn
```

▼ ポートを初期化する方法

1. Solaris 管理コンソールが実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

2. ナビゲーション区画で「コンピュータ (Computer)」アイコンをクリックします。

3. 「デバイスとハードウェア (Devices and Hardware)」、「シリアルポート (Serial Ports)」をクリックします。

シリアルポートメニューが表示されます。

4. 初期化するポートを選択します。

5. 「構成 (Configure)」→「初期化のみ – 接続なし (Initialize Only – No Connection)」を選択します。

「シリアルポート *n* を構成 (Configure Serial Port)」ウィンドウが「基本 (Basic)」モードで表示されます。

「初期化のみ (Initialize Only)」メニュー項目については、表 11-4 を参照してください。

6. 「了解 (OK)」をクリックします。

7. 各項目を設定するには、初期化のみで設定されたポートを選択します。次に、「アクション (Action)」メニューから「プロパティ (Properties)」を選択します。

8. 必要な場合は、テンプレートエントリの値を変更します。

9. 「了解 (OK)」をクリックしてポートを設定します。

10. モデムのサービスが初期化されたことを確認します。

```
$ pmadm -l -s ttyn
```

端末とモデムの問題を解決する方法

端末またはモデムを追加し、適切なサービスを設定したにもかかわらず、シリアルポート回線を通してログインできない場合は、次のような順序で問題を解決してください。

- ユーザーを確認します。

端末やモデムが正しく動作しないという報告は、多くの場合、ログインや着信ができなかったユーザーから寄せられます。したがって、まず、デスクトップに問題がないかどうかを確認することから始めてください。

ログインできない主な原因は、次のとおりです。

- ログイン ID またはパスワードが正しくない
 - 端末が X-ON フロー制御キー (Control-q) の入力を待っている
 - シリアルケーブルの接続が緩んでいるか外れている
 - 端末の設定が正しくない
 - 端末の電源が切られたか、端末に電源が入っていない
- 端末の設定を確認します。

次に、端末またはモデムの設定を調べます。端末またはモデムとの通信の正しい tty 名を調べ、それぞれの設定が tty 名の設定と一致することを確認します。

- 端末サーバーの設定を確認します。

端末に問題のないことがわかったら、端末またはモデムのサーバーに問題がないかどうかを調べます。pmadm コマンドを使用して、ポートモニターが端末またはモデムをサービスするように設定されていて、関連する tty 名が正しいことを確認します。

```
$ pmadm -l -t ttymon
```

/etc/ttydefs を調べ、ラベル定義を端末設定と照合してチェックします。sacadm を使用してポートモニターの状態を調べます。pmadm を使用して、端末が使用するポートのサービスを調べます。

- シリアル接続を確認します。

サービスアクセスコントローラが TTY ポートモニターを起動し、pmadm が端末のポートに対するサービスが有効になっていると報告し、さらに端末の設定がポートモニターの設定と一致する場合は、シリアル接続を調べて問題の原因を探します。シリアル接続は、シリアルポート、ケーブル、端末から構成されています。これらの構成部分のうち 2 つを、信頼性が確認されている他のものに取り替えて、1 箇所ずつテストしてください。

次の構成部分をすべてテストします。

- シリアルポート
 - モデム
 - ケーブル
 - コネクタ
- シリアルポートをコンソールとして使用している場合は、シリアルポートツールからシリアルポートの設定を変更しないでください。コンソール設定を正しく変更するには、/etc/inittab ファイルの次の行を変更してください。

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console
login: " -T terminal_type -d /dev/console -l console -m
ldterm,ttcompat
```


第 12 章

サービスアクセス機能によるシリアルポートの管理 (手順)

この章では、サービスアクセス機能 (SAF) によるシリアルポートサービスの管理方法を説明します。

この章で説明する手順は次のとおりです。

- 213 ページの「ttymon ポートモニターを追加する方法」
- 213 ページの「ttymon ポートモニターの状態を表示する方法」
- 214 ページの「ttymon ポートモニターを停止する方法」
- 214 ページの「ttymon ポートモニターを起動する方法」
- 215 ページの「ttymon ポートモニターを無効にする方法」
- 215 ページの「ttymon ポートモニターを有効にする方法」
- 215 ページの「ttymon ポートモニターを削除する方法」
- 216 ページの「サービスを追加する方法」
- 217 ページの「TTY ポートサービスの状態を表示する方法」
- 219 ページの「ポートモニターサービスを有効にする方法」
- 219 ページの「ポートモニターサービスを無効にする方法」

この章の内容は以下のとおりです。

- 206 ページの「サービスアクセス機能 (SAF) の概要」
- 207 ページの「SAF 全体の管理 (sacadm)」
- 208 ページの「ポートモニターサービス管理 (pmadm)」
- 210 ページの「TTY モニターとネットワークリスナーポートモニター」

SAF については、220 ページの「サービスアクセス機能の管理 (リファレンス)」を参照してください。

サービスアクセス機能 (SAF) の概要

端末とモデムの設定は、Solaris 管理コンソールのシリアルポートツール、Admintool、SAF コマンドで行うことができます。

SAF は、端末、モデム、その他のネットワーク装置を管理するためのツールです。SAF プログラムの最上位には、サービスアクセスコントローラ (SAC) があります。SAC は、`sacadm` コマンドを使用して管理するポートモニターを制御します。各ポートモニターは 1 つ以上のポートを管理できます。

管理者は `pmadm` コマンドを使用して、ポートに対応するサービスを管理します。SAC が提供するサービスはネットワークによって異なりますが、SAC と管理コマンド `sacadm` と `pmadm` はネットワークには依存しません。

表 12-1 に SAF の制御階層を示します。 `sacadm` コマンドを使用すると、`ttymon` および `listen` ポートモニターを制御する SAC を管理できます。

また、`ttymon` と `listen` のサービスは `pmadm` コマンドにより制御されます。`ttymon` の 1 つのインスタンスは複数のポートにサービスを提供できます。また、`listen` の 1 つのインスタンスは、ネットワークインタフェース上で複数のサービスを提供できます。

表 12-1 SAF の制御階層

機能	プログラム	説明
全体の管理	<code>sacadm</code>	ポートモニターの追加および削除用コマンド
サービスアクセスコントローラ	<code>sac</code>	SAF のマスタープログラム
ポートモニター	<code>ttymon</code> <code>listen</code>	シリアルポートのログイン要求を監視する ネットワークのサービス要求を監視する
ポートモニターサービスの管理	<code>pmadm</code>	ポートモニターのサービス制御用コマンド
サービス	ログイン、リモートプロシージャコール、その他	SAF がアクセスを可能にするサービス
コンソールの管理	コンソールログイン	コンソールは、 <code>/etc/inittab</code> ファイル中のエントリにより、 <code>ttymon</code> の直接モードを使用して、自動的に設定される。 <code>pmadm</code> か <code>sacadm</code> を使用して、コンソールを直接管理しないこと。詳細は、211 ページの「 <code>ttymon</code> とコンソールポート」を参照

SAF 全体の管理 (sacadm)

sacadm コマンドは SAF 階層の最上位のコマンドです。sacadm コマンドは主に、ttymon および listen などのポートモニターを追加または削除するのに使用します。このコマンドにはそれ以外に、ポートモニターの現在の状態の表示、ポートモニターの構成スクリプトの管理などの機能があります。

サービスアクセスコントローラ (SAC プログラム)

サービスアクセスコントローラ (SAC) プログラムはすべてのポートモニターを管理します。システムはマルチユーザーモードになると自動的に SAC を起動します。

SAC は、起動されるとまず、各システムの構成スクリプトを探して解釈します。構成スクリプトを使用すると、SAC の環境をカスタマイズできます。このスクリプトは、デフォルトでは空の状態です。ここで行われる SAC の環境に対する変更は、SAC のすべての「子プロセス」に継承されます。継承された環境は継承した子プロセスで変更できます。

SAC プログラムは、システムごとの構成スクリプトの解釈が終わると、SAC の管理ファイルを読み取り、指定されたポートモニターを起動します。各ポートモニターについて、SAC はそれ自身のコピーを実行します (技術的には、SAC が子プロセスをフォークします)。次に、各子プロセスは、それぞれのポートモニターごとの構成スクリプトがあればそれを解釈します。

各ポートモニターの構成スクリプトに指定されている環境を変更すると、それぞれのポートモニターが影響を受け、さらにそれがポートモニターのすべての子プロセスに継承されます。最後に、子プロセスは SAC 管理ファイル内のコマンドを使用して親であるポートモニタープログラムを実行します。

SAC の初期化プロセス

次に、SAC を最初に起動したときに行われる一連の処理を要約します。

1. init が実行レベル 2 で SAC プログラムを生成します。
2. SAC プログラムがシステムごとの構成スクリプト /etc/saf/_sysconfig を読み取ります。
3. SAC プログラムが SAC 管理ファイル /etc/saf/_sactab を読み取ります。
4. SAC プログラムが起動する各ポートモニターの子プロセスをフォークします。
5. 各ポートモニターがポートモニターごとの構成スクリプト /etc/saf/pmtag/_config を読み取ります。

ポートモニターサービス管理 (pmdm)

pmdm コマンドでポートモニターのサービスを管理できます。pmdm コマンドは特にサービスを追加または削除したり、サービスを有効または無効にしたりする場合に使用します。このコマンドでは、さらに、各サービスの構成スクリプトをインストールしたり置き換えたり、サービスに関する情報を出力したりすることもできます。

サービスの各インスタンスは、ポートモニター別、ポート別に一意に識別できなければなりません。pmdm コマンドを使用してサービスを管理する場合、*pmtag* 引数で特定のポートモニターを、また *soctag* 引数で特定のポートをそれぞれ指定します。

ポートモニターのタイプごとに、SAF はポートモニター固有の構成データのフォーマットを定義するための特別なコマンドを必要とします。この構成データは pmdm コマンドで使用します。ttymon および listen ポートモニター用の特別なコマンドは、それぞれ ttyadm と nlsadmin です。

ttymon ポートモニター

直結モデムまたは英数字端末を通してログインしようとするたびに、ttymon は動作を開始します。

図 12-1 に示すように、init プロセスがブート時に最初に起動されるプロセスです。init プロセスは、その管理ファイル (/etc/inittab) を参照して、必要に応じて他のプロセスを起動します。それらのプロセスの 1 つに SAC があります。

SAC が起動されると、今度は SAC がその管理ファイル (/etc/saf/_sactab) に指定されているポートモニターを自動的に起動します。次の図は、1 つの ttymon ポートモニターだけを示しています。

ttymon ポートモニターが起動されると、シリアルポート回線を監視してサービス要求がないかどうかを調べます。

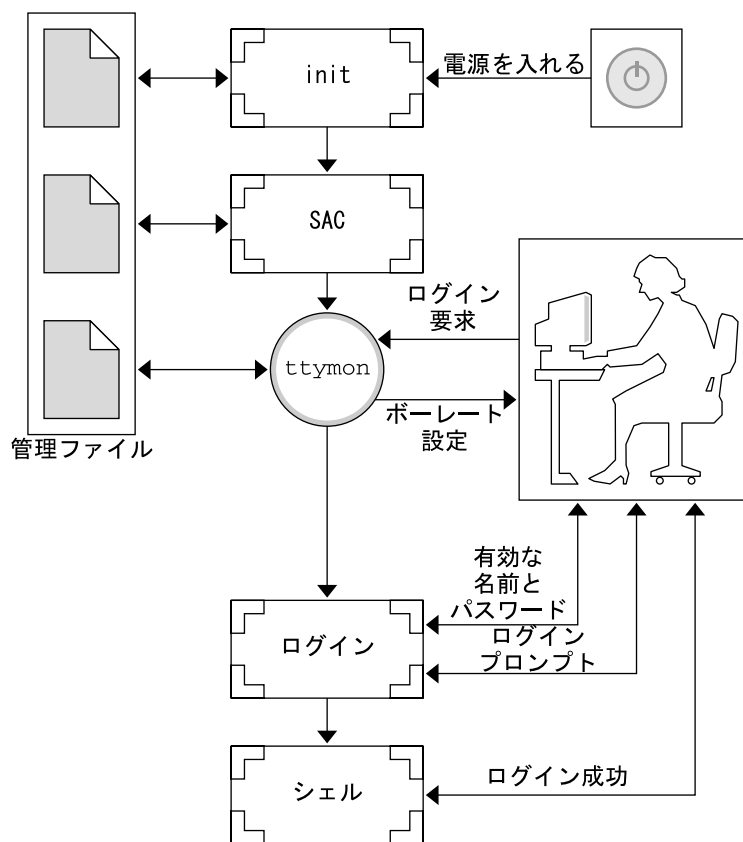


図 12-1 ttymon によるログイン要求の処理

ユーザーが英数字端末やモデムを通してログインしようとする時、シリアルポートドライバはその操作をオペレーティングシステムに伝えます。ttymon ポートモニターはシリアルポートの操作を監視し、通信リンクを確立しようとします。つまり、装置との通信に必要なデータ転送速度、回線制御手順、ハンドシェイクプロトコルを決定します。

モデムや端末との通信用の正しいパラメータの設定が終わると、ttymon ポートモニターはそれらのパラメータをログインプログラムに渡し、制御を移します。

ポートの初期化プロセス

ttymon ポートモニターのインスタンスが SAC によって実行されると、ttymon はポートの監視を始めます。ttymon ポートモニターは、各ポートについて指定されている場合、まず回線制御手順を初期化し、次に回線速度と端末の設定を初期化します。初期化に使用される値は、`/etc/ttydefs` の該当するエントリから得られます。

ttymon ポートモニターは、次に、プロンプトを表示してユーザーからの入力を待ちます。ユーザーが Break キーを押して回線速度が不適當であるという指示を与えると、ttymon ポートモニターは次の速度を設定して、再びプロンプトを表示します。

「自動ボーレート」がポートで有効な場合は、ttymon ポートモニターはそのポートのボーレートを自動的に決めようとします。ttymon ポートモニターがボーレートを認識してプロンプトを表示する前に、ユーザは Return キーを押す必要があります。

有効な入力を受け取ると、ttymon ポートモニターは次のタスクを実行します。

- ポートのサービスごとの構成ファイルを解釈する
- 必要に応じて /etc/utmpx エントリを作成する
- サービス環境を設定する
- ポートに対応したサービスを起動する

サービスが終了すると、ttymon は /etc/utmpx 中にエントリがあれば削除し、ポートを初期状態に戻します。

発着信両用サービス

ポートが発着信両用サービスに設定されている場合、ttymon ポートサービスは次のように行います。

- ユーザーをサービスに接続可能にする。
- uucico、cu、または ct コマンドが、(空いていれば) ポートを発信専用モードで使用できるようにする。
- 文字を読み取ってからプロンプトを表示する。
- 接続要求があると (connect-on-carrier フラグが設定してある場合)、プロンプトメッセージを送らないでポートの対応サービスを起動する。

TTY モニターとネットワークリスナー ポートモニター

SAF は、将来のモニターや他社製のポートモニターの管理に対応するために総合的な管理方法を提供していますが、Solaris 環境では ttymon と listen の 2 つだけが実装されています。

TTY ポートモニター (ttymon)

ttymon ポートモニターは、STREAMS をベースにしており、以下を実行します。

- ポートの監視

- 端末モード、ボーレート、回線制御手順の設定
- ログインプロセスの起動

ttymon ポートモニターは、以前のバージョンの SunOS 4.1 ソフトウェアのもとで `getty` が提供していたのと同じサービスを Solaris ユーザーに提供します。

ttymon ポートモニターは SAC プログラムで実行され、`sacadm` コマンドを使用して構成します。ttymon の各インスタンスはそれぞれに複数のポートを監視できます。それらのポートはポートモニターの管理ファイル内に指定します。また、この管理ファイルは `pmadm` および `ttymax` コマンドを使用して構成します。

ttymon とコンソールポート

コンソールサービスの管理は、サービスアクセスコントローラや、明示的な ttymon 管理ファイルの実行によるものではありません。/etc/inittab ファイル内のエントリを使用し、ttymon を直接モードで使用して、コンソールポートの管理を行います。直接モードとは、ttymon の特別なモードのことで、ログインサービスが必要なコマンドによって直接呼び出されます。

/etc/inittab ファイル内のデフォルトのコンソールエントリは次のようになります。

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
-T terminal_type -d /dev/console -l console -m ldterm,ttcompat
```

co:234:respawn:

co は、エントリをコンソールとして指定する。234 は動作の実行レベルを指定する。respawn は、コンソールエントリが失敗した場合、あるいは実行レベル 2、3、4 に存在しない場合に、コンソールエントリを再起動する必要があることを示す

/usr/lib/saf/ttymon -g -h

-g オプションを使用するため、正しいボーレートと正しい端末設定をポート上で設定でき、SAC による事前構成なしに、ログインサービスに接続できる。-h オプションは、デフォルトまたは指定した速度に設定する前に、回線速度をゼロに設定することにより、回線をハングアップさせる

-p "`uname -n` console login:

コンソールポート用のプロンプト文字列を指定する

-t terminal_type

コンソールの端末タイプを指定する

```
-d /dev/console -l console
-m ldterm,ttcompat
```

-d オプションは、コンソールデバイスを指定する。-l オプションは、`/etc/ttydefs`ファイル内の `ttylabel` を指定する。-m オプションは、プッシュする STREAMS モジュールを指定する

ttymon 固有の管理コマンド (ttyadm)

ttymon の管理ファイルは、`sacadm` および `pmadm` の他に `tttyadm` コマンドによっても更新できます。tttyadm コマンドは、ttymon 固有の情報を書式化し、それらの情報を標準出力に書き出し、書式化された ttymon 固有のデータを `sacadm` および `pmadm` コマンドに提示する手段を提供します。

したがって、tttyadm は ttymon を直接管理するのではなく、一般的な管理用コマンドである `sacadm` および `pmadm` を補足するものです。詳細は、`tttyadm(1M)` のマニュアルページを参照してください。

ネットワークリスナーサービス (listen)

listen ポートモニターは SAC プログラムで動作し、以下を実行します。

- ネットワークを監視してサービス要求がないかを調べる
- サービス要求があれば受け付ける
- それらのサービス要求に応答してサーバーを呼び出す

listen ポートモニターは `sacadm` コマンドを使用して構成します。listen の各インスタンスはそれぞれに複数のサービスを提供できます。それらのサービスは listen ポートモニターの管理ファイルに指定します。この管理ファイルは `pmadm` および `nlsadmin` コマンドを使用して構成します。

ネットワークリスナープロセスは、トランスポート層インタフェース (TLI) 仕様に準拠する任意の接続型トランスポートプロバイダで使用できます。Solaris 環境では、listen ポートモニターは、`inetd` サービスが提供しない追加ネットワークサービスを提供できます。

listen 固有の管理コマンド (nlsadmin)

listen ポートモニターの管理ファイルは、`sacadm` および `pmadm` の他に、`nlsadmin` コマンドでも更新できます。`nlsadmin` コマンドは、listen 固有の情報の書式を定義して標準出力に書き込み、書式付きの listen 固有のデータを `sacadm` および `pmadm` コマンドに提示する手段を提供します。

したがって、`nlsadmin` は listen を直接管理するのではなく、一般的な管理用コマンドである `sacadm` および `pmadm` を補足するものです。

個別に構成される各ネットワークには、ネットワークリスナープロセスのインスタンスが少なくとも1つは存在します。nlsadmin コマンドは listen ポートモニターの動作状態を制御します。

nlsadmin コマンドは、与えられたネットワークに対して listen ポートモニターを設定し、そのポートモニターの固有の属性を構成し、そのモニターを起動したり、強制終了させたりすることができます。さらに、マシン上にある listen ポートモニターについて報告することもできます。

詳細は、nlsadmin(1M) のマニュアルページを参照してください。

ttymon ポートモニターの管理

sacadm コマンドを使用して ttymon ポートモニターを追加、表示、削除、終了、起動、あるいは有効または無効にすることができます。

▼ ttymon ポートモニターを追加する方法

1. スーパーユーザーになります。
2. **ttymon** ポートモニターを追加します。

```
# sacadm -a -p mbmon -t ttymon -c /usr/lib/saf/ttymon -v `ttyadm  
-V` -y "TTY Ports a & b"
```

- a ポートモニター追加用のオプションを指定する
- p mbmon をポートモニタータグとして指定する
- t ポートモニタータイプを ttymon として指定する
- c ポートモニターを起動するのに使用するコマンド文字列を定義する
- v ポートモニターのバージョン番号を指定する
- y ポートモニターのインスタンスを説明するコメントを指定する

▼ ttymon ポートモニターの状態を表示する方法

1. スーパーユーザーになります。
2. **ttymon** ポートモニターの状態を表示します。

```
# sacadm -l -p mbmon
```

- l ポートモニター状態表示用のフラグを指定する
- p mbmon をポートモニタータグとして指定する

例 - ttymon ポートモニターの状態を表示する

```
# sacadm -l -p mbmon
PMTAG  PMTYPE  FLGS RCNT STATUS  COMMAND
mbmon  ttymon  -    0    STARTING /usr/lib/saf/ttymon #TTY Ports a & b
```

PMTAG	ポートモニター名が mbmon であることを示す
PMTYPE	ポートモニターのタイプが ttymon であることを示す
FLGS	次のフラグが設定されているかどうかを示す <ul style="list-style-type: none"> ■ d は、新しいポートモニターを有効にしない ■ x は、新しいポートモニターを起動しない ■ ダッシュ (-) — フラグの設定なし
RCNT	戻りカウント値を示す。0 の戻りカウントは、ポートモニターが失敗した場合でも再起動しないことを示す
STATUS	ポートモニターの現在の状態を示す
COMMAND	ポートモニターを起動するコマンドを示す
#TTY Ports a & b	ポートモニターを説明するコメントを示す

▼ ttymon ポートモニターを停止する方法

1. スーパーユーザーになります。
2. **ttymon** ポートモニターを停止します。

```
# sacadm -k -p mbmon
```

- k ポートモニターを終了状態にするフラグを指定する
- p mbmon をポートモニタータグとして指定する

▼ ttymon ポートモニターを起動する方法

1. スーパーユーザーになります。
2. 終了した **ttymon** ポートモニターを起動します。

```
# sacadm -s -p mbmon
```

- s ポートモニターを起動状態にするフラグを指定する
- p mbmon をポートモニタータグとして指定する

▼ ttymon ポートモニターを無効にする方法

ポートモニターを無効にすると、以前から存在しているサービスをそのまま有効にするため、新しいサービスが起動できなくなります。

1. スーパーユーザーになります。
2. **ttymon** ポートモニターを無効にします。

```
# sacadm -d -p mbmon
```

- d ポートモニターを無効状態にするフラグを指定する
- p mbmon をポートモニタータグとして指定する

▼ ttymon ポートモニターを有効にする方法

ttymon ポートモニターを有効にすると、そのモニターが新しい要求にサービスを提供できるようになります。

1. スーパーユーザーになります。
2. **ttymon** ポートモニターを有効にします。

```
# sacadm -e -p mbmon
```

- e ポートモニターを有効状態にするフラグを指定する
- p mbmon をポートモニタータグとして指定する

▼ ttymon ポートモニターを削除する方法

1. スーパーユーザーになります。
2. **ttymon** ポートモニターを削除します。

```
# sacadm -r -p mbmon
```

- r ポートモニター削除用のフラグを指定する
- p mbmon をポートモニタータグとして指定する

注 - ポートモニターを削除すると、それに関連するすべての構成ファイルが削除されます。ポートモニター構成ファイルは `sacadm` コマンドでは更新や変更ができません。ポートモニターを再構成するには、古いポートモニターを削除してから新しいポートモニターを追加してください。

ttymon サービスの管理

`pmadm` コマンドを使用してサービスを追加したり、ポートモニターに関連付けられている1つ以上のポートのサービスを表示したり、サービスを有効または無効にしたりできます。

▼ サービスを追加する方法

1. スーパーユーザーになります。
2. 標準の端末サービスを `mbmon` ポートモニターに追加します。

```
# pmadm -a -p mbmon -s a -i root -v `ttyadm -V` -m "`ttyadm -i 'Terminal
disabled' -l contty -m ldterm,ttcompat -S y -d /dev/term/a
-s /usr/bin/login`"
```

注 - 上記の例では入力が自動的に次の行にまたがっていますが、実際には Return キーを押さずに (改行なしに) 入力します。

- a ポートモニター追加用のフラグを指定する
- p mbmon をポートモニタータグとして指定する
- s a をポートモニターサービスタグとして指定する
- i 識別情報を、サービス実行中にポートモニターサービスタグに割り当てられるように指定する
- v ポートモニターのバージョン番号を指定する

-m ttyadm により書式化された ttymon 固有の構成データを指定する

上記の pmadm コマンドには ttyadm コマンドが組み込まれています。その組み込みコマンドの中の指定項目の意味は次のとおりです。

-b ポートを双方向用にするフラグを指定する
-i 無効応答メッセージを指定する
-l /etc/ttydefs ファイルにあるどの TTY 名を使用するか指定する
-m サービスを起動する前にプッシュする STREAMS モジュールを指定する
-d TTY ポートに使用する装置へのフルパス名を指定する
-s 接続要求を受信したとき起動するサービスへのフルパス名を指定する。引数が必要な場合、コマンドと引数を二重引用符 (") で囲む

▼ TTY ポートサービスの状態を表示する方法

pmadm コマンドを次に示すように使用して、特定のポートモニターに設定されている 1 つまたはすべての TTY ポートの状態を表示します。

1. スーパーユーザーになります。
2. ポートモニターの 1 つのサービスを表示します。

```
# pmadm -l -p mbmon -s a
```

-l システム上のサービス情報を表示する
-p mbmon をポートモニタータグとして指定する
-s a をポートモニターサービスタグとして指定する

例 - 1 つのポートモニターのすべてのサービスを表示する

この例は、ポートモニターのすべてのサービスを表示します。

```
# pmadm -l -p mbmon
PMTAG PMTYPE SVCTAG FLAGS ID <PMSPECIFIC>
mbmon ttymon a - root /dev/term/a - - /usr/bin/login - contty
ldterm,ttcompat login: Terminal disabled tvi925 y #
```

PMTAG	pmadm -p コマンドを使用して設定された、ポートモニター名 mbmon を示す
PMTYPE	ポートモニターのタイプが ttymon であることを示す
SVCTAG	pmadm -s コマンドを使用して設定された、サービスタグ値を示す
FLAGS	次のフラグが pmadm -f コマンドを使用して設定されているかどうかを示す <ul style="list-style-type: none"> ■ x — サービスを有効にしない ■ u — サービス用の utmpx エントリを作成する ■ ダッシュ (-) — フラグの設定なし
ID	起動時にサービスに割り当てられた ID を示す。この値は、pmadm -i コマンドを使用して設定される
<PMSPECIFIC> の情報	
/dev/term/a	ttyadm -d コマンドを使用して設定された、TTY ポートパス名を示す
-	次のフラグが ttyadm -c -b -h -I -r コマンドを使用して設定されているかどうかを示す。 <ul style="list-style-type: none"> ■ c — キャリア検出時に接続するよう、ポートを設定する ■ b — ポートが双方向性である (着信トラフィックと発信トラフィックの両方を許可する) と設定する ■ h — 着呼が受信された直後の自動ハンガアップを抑制する ■ I — ポートを初期化する ■ r — login: メッセージを出力する前に、ポートから文字を受信するまで、ttymon を待機させる ■ ダッシュ (-) — フラグの設定なし
-	ttyadm -r <i>count</i> オプションを使用して設定された値を示す。このオプションは、ポートからデータを受信後に、ttymon がプロンプトを表示するときを決定する。 <i>count</i> が 0 の場合、ttymon は任意の文字を受信するまで待機する。 <i>count</i> が 0 より大きい場合、ttymon は <i>count</i> の数だけ新規行を受信するまで待機する。この例では、値は設定されていない
/usr/bin/login	接続を受信したときに呼び出されるサービスのフルパス名を示す。この値は、ttyadm -s コマンドを使用して設定される
-	ttyadm -t コマンドのタイムアウト値を示す。このオプションは、ポートを開くことに成功して、かつ入力データが <i>timeout</i> 秒内に受信されていない場合に、ttymon がポートを閉じることを意味する。この例では、タイムアウト値は設定されていない

contty	/etc/ttydefs ファイル中の TTY 名を示す。この値は、 <code>ttyadm -l</code> コマンドを使用して設定される
ldterm, ttcompat	プッシュする STREAMS モジュールを示す。これらのモジュールは、 <code>ttyadmin -m</code> コマンドを使用して設定される
login: Terminal disabled	ポートが無効であるときに表示される、アクティブでないメッセージを示す。このメッセージは、 <code>ttyadm -i</code> コマンドを使用して設定される
tvi925	<code>ttyadm -T</code> コマンドを使用して設定されている場合、端末タイプを示す。この例では、端末タイプは、 <code>tvi925</code>
y	<code>ttyadm -S</code> コマンドを使用して設定されたソフトウェアキャリア値を示す。 <code>n</code> は、ソフトウェアキャリアをオフにする。 <code>y</code> は、ソフトウェアキャリアをオンにする。この例では、ソフトウェアキャリアはオン
#	<code>pmadm -y</code> コマンドで指定した任意のコメントを示す。この例では、コメントは存在しない

▼ ポートモニターサービスを有効にする方法

1. スーパーユーザーになります。
2. 無効になっているポートモニターサービスを有効にします。

```
# pmadm -e -p mbmon -s a
```

```
-e      有効フラグを指定する
-p      mbmon をポートモニタータグとして指定する
-s      a をポートモニターサービスタグとして指定する
```

▼ ポートモニターサービスを無効にする方法

1. スーパーユーザーになります。
2. ポートモニターサービスを無効にします。

```
# pmadm -d -p mbmon -s a
```

```
-d      無効フラグを指定する
-p      mbmon をポートモニタータグとして指定する
```

-s a をポートモニターサービスタグとして指定する

サービスアクセス機能の管理 (リファレンス)

SAF の関連ファイル

SAF は構成ファイルを使用しますが、このファイルは `sacadm` および `pmadm` コマンドを使用して変更できます。構成ファイルを手作業で編集する必要はありません。

ファイル名	説明
<code>/etc/saf/_sysconfig</code>	システムごとの構成スクリプト
<code>/etc/saf/_sactab</code>	SAC の管理ファイル。SAC が制御するポートモニターの構成データが入っている
<code>/etc/saf/pmtag</code>	ポートモニター <code>pmtag</code> のホームディレクトリ
<code>/etc/saf/pmtag/_config</code>	存在する場合、ポートモニター <code>pmtag</code> のポートモニターごとの構成スクリプト
<code>/etc/saf/pmtag/_pmtab</code>	ポートモニター <code>pmtag</code> の管理ファイル。 <code>pmtag</code> が提供するサービスのポートモニター固有の構成データが入っている
<code>/etc/saf/pmtag/svctag</code>	サービス <code>svctag</code> のサービスごとの構成スクリプト
<code>/var/saf/log</code>	SAC のログファイル
<code>/var/saf/pmtag</code>	<code>pmtag</code> によって作成されるファイルのディレクトリ。たとえば、ログファイルのディレクトリなど

`/etc/saf/_sactab` ファイル

`/etc/saf/_sactab` は次のようになります。

```
# VERSION=1
zsmon:ttymon::0:/usr/lib/saf/ttymon #
```

# VERSION=1	サービスアクセス機能のバージョン番号を示す
zsmon	ポートモニター名
ttymon	ポートモニターのタイプ
::	次の2つのフラグが設定されているかどうかを示す
	d — ポートモニターを有効にしない
	x — ポートモニターを起動しない。この例では、 どちらのフラグも設定されていない
0	戻りコード値を示す。0の戻りカウントは、ポート モニターが失敗した場合でも再起動しないこと を示す
/usr/lib/saf/ttymon	ポートモニターのパス名を示す

/etc/saf/pmtab/_pmtab ファイル

/etc/saf/pmtab/_pmtab ファイル (/etc/saf/zsmon/_pmtab など) は、次のようになります。

```
# VERSION=1
ttya:u:root:reserved:reserved:/dev/term/a:I::/usr/bin/login::9600:
ldterm,ttcompat:ttya login\: ::tvi925:y:#
```

# VERSION=1	サービスアクセス機能のバージョン番号を示す
ttya	サービスタグを示す
x,u	次のフラグが設定されているかどうかを示す x — サービスを有効にしないことを意味する u — サービス用の utmpx エントリを作成することを意味する。
root	サービスタグに割り当てられた ID を示す
reserved	このフィールドは予約されている
reserved	このフィールドは予約されている
reserved	このフィールドは予約されている
/dev/term/a	TTY ポートパス名を示す
/usr/bin/login	接続を受信したときに呼び出されるサービスのフルパス名を指定する

<code>:c,b,h,I,r:</code>	次のフラグが設定されているかどうかを示す c — キャリア検出時に接続するよう、ポートを設定する b — ポートが双方向性である (着信トラフィックと発信トラフィックの両方を許可する) と設定する h — 着呼が受信された直後の自動ハングアップを抑制する I — ポートを初期化する r — ポートから文字を受信するまで、 <code>ttymon</code> が <code>login:</code> メッセージを出力しないよう待機させる
<code>9600</code>	<code>/etc/ttydefs</code> ファイルに定義されている TTY 名を指定する
<code>ldterm,ttcompat</code>	プッシュする STREAMS モジュールを指定する
<code>ttya login\:</code>	表示するプロンプトを指定する
<code>:y/n:</code>	
<code>message</code>	任意のアクティブでない (無効な) 応答メッセージを指定する
<code>tvi925</code>	端末タイプを指定する
<code>y</code>	ソフトウェアキャリアが設定されているかどうかを示す (y/n)

サービスの状態

`sacadm` コマンドはサービスの状態を制御します。次の表は、サービスの起こりうる状態について説明したものです。

状態	説明
有効	デフォルト状態 - ポートモニターを追加したとき、サービスが有効になる
無効	デフォルト状態 - ポートモニターを削除したとき、サービスは停止する

特定のサービスの状態を確認するには、次のように入力します。

```
# pmadm -l -p portmon_name -ssvctag
```

ポートモニターの状態

sacadm コマンドは、ttymon および listen ポートモニターの状態を制御します。次の表は、起こりうるポートモニターの状態について説明したものです。

状態	説明
起動	デフォルト状態 - ポートモニターは追加されると自動的に起動される
有効	デフォルト状態 - ポートは追加されると自動的にサービス要求を受け付け可能になる
停止	デフォルト状態 - ポートモニターは削除されると自動的に停止する
無効	デフォルト状態 - ポートモニターは削除されると自動的に提供中であったサービスを続行し、新しいサービスの追加を拒否する
起動中	中間状態 - ポートモニターの起動が進行中
停止中	中間状態 - ポートモニターは手作業で終了過程に入っているが、まだシャットダウン手続きは完了していない。ポートモニターが停止状態になるまでの途中の状態
非動作中	アクティブではない状態 - ポートモニターが強制終了された状態。前の動作状態のときに監視していたすべてのポートがアクセス不可になる。外部のユーザーからはポートが無効なのか、非動作状態なのか区別できない
障害	アクティブではない状態 - ポートモニターを起動して動作状態を維持できない

特定のポートモニターの状態を確認するには、次のように入力します。

```
# sacadm -l -p portmon_name
```

ポートの状態

ポートは、ポートを制御するポートモニターの状態によって、有効または無効にできます。

状態	説明
シリアル (ttymon) ポートの状態	
有効	ttymon ポートモニターはポートにプロンプトメッセージを送り、ログインサービスを提供する

状態	説明
無効	ttymon が強制終了されているか、無効の場合のすべてのポートのデフォルト状態。このように指定した場合、接続要求を受け取ると、ttymon は「disabled」メッセージを送信する

第 13 章

システム資源の管理 (トピック)

以下の各章で、システム資源の管理について説明します。

第 14 章	ディスク割り当て制限、cron および at コマンド、アカウントティングプログラムによるシステム資源の管理に役立つ Solaris コマンドとユーティリティの概要を説明します。
第 15 章	システム情報を表示したり、変更したりする手順を説明します。
第 16 章	使用されていないファイルや大きなディレクトリを見つけて、ディスク空間を最適化するための手順を説明します。
第 17 章	ディスク割り当て制限を設定し、管理する手順を説明します。
第 18 章	crontab および at コマンドを使用して、繰り返しのまたは 1 度限りのシステムイベントのスケジュールを設定する手順を説明します。
第 19 章	アカウントティングを設定し、管理する手順を説明します。
第 20 章	システムアカウントティングソフトウェアに関するリファレンス情報を提供します。その中には、さまざまなアカウントティング機能の操作方法、さまざまなアカウントティングプログラムの使用方法などが含まれます。

第 14 章

システム資源の管理 (概要)

この章では、Solaris 9 の新しい機能とシステム資源の管理に役立つ機能について概説します。

これらの機能を使用して、一般的なシステム情報を表示し、ディスク容量を監視し、ディスク割り当て制限を設定し、アカウントティングプログラムを使用し、`crontab` と `at` コマンドをスケジュールして、日常的に使用するコマンドを自動的に実行することができます。

この節では、柔軟な方法でシステム資源の割り当て、監視、制御を可能にする Solaris の資源管理機能については言及しません。

Solaris 資源管理機能を使用しないでシステム資源を管理する手順については、[229 ページの「システム資源管理の概要 \(作業マップ\)」](#)を参照してください。

Solaris 資源管理機能を使用したシステム資源の管理については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「Solaris 9 リソースマネージャの紹介」を参照してください。

システム資源管理の新機能

df、du、ls の新オプション

df、du、および ls コマンドの新しい `-h` オプションは、ディスクの使用状況とファイルやファイルシステムのサイズを 1024 の累乗で表示します。ファイルやディレクトリのサイズが 1024 バイトを上回る場合、このオプションは、ディスク容量をキロバイト、メガバイト、ギガバイト、またはテラバイト単位で表示するため、df、du、および ls `-l` コマンドの出力が分かりやすくなります。

詳細については、df(1B)、du(1)、およびls(1)を参照してください。

このガイドの詳細は、240 ページの「ファイルとディスク容量の情報を表示する方法」を参照してください。

システムパラメータの変更

このマニュアルの前バージョンには、ユーザー 1 人あたりのプロセス数と共有メモリのセグメントを増やすといった、システムパラメータの変更について記述されている節がありました。その節は削除されました。調整可能なシステムパラメータの変更については、『Solaris カーネルのチューンアップ・リファレンスマニュアル』を参照してください。

また、ネットワーク時間サーバーの設定については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「システム時刻関連サービス」に移動しました。

Solaris 資源管理

Solaris 資源管理を使用すると、アプリケーションがシステム資源を利用する状況を制御できます。

また、拡張アカウンティング機能を設定して、タスクやプロセスごとのリソースの消費状況を監視したり記録することができます。

拡張アカウンティングでは、アカウンティングデータの一般グループをあらわす、新しい可変長の汎用アカウンティングファイル形式を導入しています。これによって、カーネルによってさまざまなアカウンティングファイルに記録される資源の利用率を設定することができます。

Solaris 資源管理機能には、次のものがあります。

- タスク — リソースの使用状況を追跡するための新しいプロセス集合
- プロジェクト — リソースの使用状況を請求するための新しい管理データベース。実行されたタスクに基づいて、資源をプロジェクトに充てることができます。
- acctadm — 拡張アカウンティング機能の様々な属性を構成するための新しいツール。たとえば、システムのアカウンティングによって追跡された資源は、システム全体に基づいて構成することができます。

デフォルトの新しいアカウンティング構成は管理を必要としません。ただし、拡張アカウンティング機能を使用しなくても、/etc/project ファイルを削除しないでください。このファイルが存在しないと、ユーザーはシステムにログインできません。

このリリースで拡張されたアカウンティング機能についての詳細な情報は、次の表を参照してください。

Solaris 資源管理機能	参照箇所
拡張アカウントिंगのオンとオフ	『Solaris のシステム管理 (資源管理とネットワークサービス)』の「拡張アカウントング」および acctadm (1M) のマニュアルページ
プロジェクトデータベースの記述について	『Solaris のシステム管理 (資源管理とネットワークサービス)』の「プロジェクトとタスク」および project (4) のマニュアルページ
拡張されたアカウントング機能のデータの処理について	libexacct (3LIB)、getacct (2)、putacct (2)、および wracct (2)

システム資源管理の概要 (作業マップ)

作業	説明	参照先
システム情報の表示と変更	さまざまなコマンドを使用して、一般的なシステム情報、日付と時間、システムのホスト名の変更など、システム情報を表示したり変更することができます。	第 15 章
ディスク使用状況の管理	ディスク容量の使用状況を識別し、古い使用されていないファイルを削除します。	第 16 章
割り当て制限の管理	UFS ファイルシステム割り当て制限を使用して、どのくらいのディスク容量がユーザーによって使用されるのかを管理します。	第 17 章
システムイベントのスケジュール	cron および at ジョブを使用して、古い使用されていないファイルの整理などのスケジュールシステムの日常的な定型作業に役立てます。	第 18 章
システムアカウントングの管理	システムアカウントングを使用して、ユーザーとアプリケーションが使用するシステム資源を識別します。	第 19 章

第 15 章

システム情報の表示と変更 (手順)

この章では、最も一般的なシステム情報を表示および変更するために必要な手順を示します。この章で説明する手順は次のとおりです。

- 232 ページの「システムが 64 ビット Solaris オペレーティングシステムを実行できるか調べる方法」
- 233 ページの「64 ビット Solaris 機能が有効になっているか調べる方法」
- 234 ページの「システムのプロセッサタイプを表示する方法」
- 235 ページの「システムのホスト ID を表示する方法」
- 236 ページの「日付と時刻を表示する方法」
- 236 ページの「システムの日付と時刻を手作業で設定する方法」
- 237 ページの「その日のメッセージを設定する方法」
- 237 ページの「システムのホスト名の変更方法」

システム情報の表示

表 15-1 に、一般的なシステム情報を表示するためのコマンドを示します。

表 15-1 システム情報を表示するためのコマンド

コマンド	マニュアルページ	表示されるシステム情報
date	date(1)	日付と時刻
hostid	hostid(1)	ホスト ID
isainfo	isainfo(1)	動作しているシステムのネイティブアプリケーションによってサポートされるビット数を報告する。ビット数は、トークンとしてスクリプトに渡すことができる

表 15-1 システム情報を表示するためのコマンド (続き)

コマンド	マニュアルページ	表示されるシステム情報
isalist	isalist (1)	x86 のプロセッサタイプを識別する
prtconf	prtconf (1M)	インストールされているメモリー量
psrinfo	psrinfo (1M)	プロセッサタイプ
showrev	showrev (1M)	ホスト名、ホスト ID、リリース、カーネルアーキテクチャ、アプリケーションアーキテクチャ、ハードウェアプロバイダ、ドメイン、およびカーネルのバージョン
uname	uname (1)	オペレーティングシステム名、リリース、バージョン、ノード名、ハードウェア名、およびプロセッサタイプ

システムが 64 ビット Solaris オペレーティングシステムを実行できるか調べる方法

現在、64 ビット Solaris オペレーティングシステムをサポートするプラットフォームは UltraSPARC システムだけです。システムが UltraSPARC システムかどうかを確認するには、次のコマンドを使用します。

```
$ uname -m
sun4u
```

uname -m コマンドの出力が sun4u の場合、そのマシンは UltraSPARC システムです。

マシンが UltraSPARC システムであるかどうかは、次のように psrinfo コマンドを使用して確認できます。

```
# psrinfo -v
Status of processor 0 as of: 05/30/01 13:48:46
  Processor has been on-line since 05/18/01 10:02:40.
  The sparcv9 processor operates at 333 MHz,
    and has a sparcv9 floating point processor.
```

プロセッサタイプが sparcv9 であれば、そのプラットフォームで 64 ビット Solaris OS が実行できます。

64 ビット Solaris 機能が有効になっているか調べる方法

システムで 64 ビット Solaris 機能が有効になっているか調べるには `isainfo` コマンドが使用できます。有効になっていれば、システムは 64 ビットカーネルでブートされています。

例— 64 ビット Solaris 機能が有効になっているか調べる

32 ビットカーネルを実行している UltraSPARC システムの出力は、次のようになります。

```
$ isainfo -v
32-bit sparc applications
```

この出力は、システムが 32 ビットアプリケーションだけをサポートすることを示します。

64 ビットカーネルを実行している UltraSPARC システムの出力は、次のようになります。

```
$ isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

この出力は、システムが 32 ビットと 64 ビットのアプリケーションを両方サポートすることを示しています。

動作しているシステムのネイティブアプリケーションによってサポートされるビット数を表示するには、`isainfo -b` コマンドを使用します。

32 ビット Solaris OS が動作する SPARC、x86、UltraSPARC システムは、次のように表示されます。

```
$ isainfo -b
32
```

64 ビット Solaris OS が動作する 64 ビット UltraSPARC システムは、次のように表示されます。

```
$ isainfo -b
64
```

コマンドは 64 だけを返します。64 ビット UltraSPARC システムでは 32 ビットと 64 ビットのアプリケーションが両方動作しますが、64 ビットシステムで実行するには 64 ビットアプリケーションが最適です。

システムのプロセッサタイプを表示する方法

システムのプロセッサタイプを確認するには、`psrinfo -v` コマンドを使用します。次の例に示すように、このコマンドはプロセッサの数も表示します。

```
#psrinfo -v
Status of processor 0 as of: 02/24/2003 09:11:05
  Processor has been on-line since 02/03/2003 10:21:46.
  The sparcv9 processor operates at 600 MHz,
    and has a sparcv9 floating point processor.
Status of processor 1 as of: 02/24/2003 09:11:05
  Processor has been on-line since 02/24/2003 07:02:28.
  The sparcv9 processor operates at 600 MHz,
    and has a sparcv9 floating point processor.
```

x86 システムでは、`isalist` コマンドを使用してプロセッサタイプを確認します。

```
#isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

Solaris リリース情報を表示する方法

Solaris リリースバージョンを特定するには、`/etc/release` ファイルの内容を表示します。

```
$ more /etc/release
Solaris 9 s9_58shwpl3 SPARC
Copyright 2002 Sun Microsystems, Inc. All Rights Reserved.
Use is subject to license terms.
Assembled 15 April 2002
```

一般的なシステム情報を表示する方法

一般的なシステム情報を表示するには、`showrev` コマンドを使用します。

例 — 一般的なシステム情報を表示する

次の例は、`showrev` コマンドの出力を示します。-a オプションは、提供可能なあらゆるシステム情報を表示します。

```
$ showrev -a
Hostname: starbug
Hostid: nnnnnnnn
Release: 5.9
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: solar.com
```

```
Kernel version: SunOS 5.9 May 2002
```

```
OpenWindows version:  
X11 Version 6.6.1 5 April 2002
```

```
No patches are installed  
$
```

システム情報の表示には、`uname` コマンドも使用できます。次の例は、`uname` コマンドの出力を示します。`-a` オプションを使用すると、オペレーティングシステム名とともに、システムノード名、オペレーティングシステムのリリース、オペレーティングシステムのバージョン、ハードウェア名、プロセッサタイプを表示します。

```
$ uname  
SunOS  
$ uname -a  
SunOS starbug 5.9 Generic sun4u sparc SUNW,Ultra-5_10  
$
```

システムのホスト ID を表示する方法

ホスト ID を 16 進形式で表示するには、`hostid` コマンドを使用します。

例 — システムのホスト ID を表示する

次の例は、`hostid` コマンドの出力を示します。

```
$ hostid  
80a5d34c
```

システムにインストールされているメモリーを表示する方法

システムにインストールされているメモリー量を表示するには、`prtconf` コマンドを使用します。

例 — システムにインストールされているメモリーを表示する

次の例は、`prtconf` コマンドの出力例を示します。`grep Memory` コマンドは、`prtconf` コマンドの出力からメモリーに関する情報だけを表示します。

```
# prtconf | grep Memory  
Memory size: 128 Megabytes
```

日付と時刻を表示する方法

システムクロックに従った現在の日付と時刻を表示するには、`date` コマンドを使用します。

例 — 日付と時刻を表示する

次の例は、`date` コマンドの出力例を示します。

```
$ date
Thu May 31 17:44:58 MDT 2003
$
```

システム情報の変更

この節では、一般的なシステム情報を変更するコマンドを説明します。

▼ システムの日付と時刻を手作業で設定する方法

1. スーパーユーザーになります。
2. 次のように新しい日付と時刻を入力します。

```
# date mmddHHMM[[cc]yy]
```

<i>mm</i>	月。2桁を使用
<i>dd</i>	日。2桁を使用
<i>HH</i>	時。2桁で24時間制を使用
<i>MM</i>	分。2桁を使用
<i>cc</i>	世紀。2桁を使用
<i>yy</i>	年。2桁を使用

詳細は、`date(1)` のマニュアルページを参照してください。

3. オプションを付けずに `date` コマンドを使用し、システム日付が正しく設定されたことを確認します。

例 — システムの日付と時刻を手作業で設定する

次の例は、`date` コマンドを使用して手作業でシステムの日付と時刻を設定する方法を示します。

```
# date
Thu Jun 21 13:59:15 MDT 2003
# date 0621141003
Thu Jun 21 14:10:00 MDT 2003
```

▼ その日のメッセージを設定する方法

その日のメッセージファイル `/etc/motd` を編集して、システムの全ユーザーに対して、ログイン時に通知または問い合わせる内容を書き込みます。ただし、この機能を使用するときは、必要なメッセージだけを送るようにします。メッセージファイルは定期的に編集して、不用になったメッセージを削除することをお勧めします。

1. スーパーユーザーになります。
2. エディタを使って `/etc/motd` ファイルを開き、必要なメッセージを追加します。テキストを編集して、ユーザーのログイン時に表示されるメッセージを記述します。スペース、タブ、リターンも入れてください。
3. `/etc/motd` ファイルの内容を表示して、変更内容を確認します。

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

例 — その日のメッセージを設定する

Solaris ソフトウェアのインストール時に、デフォルトのその日のメッセージが設定されます。メッセージの内容は次のような SunOS バージョン情報です。

```
$ cat /etc/motd
Sun Microsystems Inc.   SunOS 5.9           Generic   May 2002
```

次の例は、編集後の `/etc/motd` ファイルの内容を示します。このファイルは、ログインする各ユーザーに対してシステムの利用度に関する情報を提供します。

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m. on
Saturday, July 7, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you.
```

▼ システムのホスト名の変更方法

システムのホスト名は、複数の異なる場所に指定します。

ネームサービスデータベースを更新して、新しいホスト名を反映させる必要があることを覚えておいてください。

システム名の変更は、次の手順で行なってください。

1. スーパーユーザーになります。
2. 次のファイルでシステムのホスト名を変更します。
 - /etc/nodename
 - /etc/hostname.*xy*
 - /etc/inet/hosts
 - /etc/net/ticlts/hosts
 - /etc/net/ticots/hosts
 - /etc/net/ticotsord/hosts
3. (省略可能) ネームサービスを使用している場合は、**host** ファイルでシステムのホスト名を変更します。
4. システムをリブートして、新しいホスト名を起動します。

```
# init 6
```

第 16 章

ディスク使用の管理 (手順)

この章では、使用していないファイルや大きなディレクトリを見つけることにより、ディスク容量を最適化する方法を示します。この章で説明する手順は次のとおりです。

この章で説明する手順は次のとおりです。

- 240 ページの「ファイルとディスク容量の情報を表示する方法」
- 242 ページの「ファイルサイズを表示する方法」
- 243 ページの「サイズの大きなファイルを見つける方法」
- 244 ページの「指定されたサイズ制限を超えるファイルを見つける方法」
- 245 ページの「ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する方法」
- 246 ページの「ローカル UFS ファイルシステムのユーザーの所有状況を表示する方法」
- 247 ページの「最新ファイルのリストを表示する方法」
- 248 ページの「古いファイルと使用されていないファイルを見つけて削除する方法」
- 249 ページの「一時ディレクトリを一度にクリアする方法」
- 250 ページの「core ファイルを見つけて削除する方法」
- 250 ページの「クラッシュダンプファイルを削除する方法」

ファイルとディスク容量の情報の表示

次の表は、サイズとディスク容量の情報を表示するコマンドをまとめたものです。

コマンド	マニュアルページ	説明
df	df (1M)	空きディスクブロック数とファイル数を表示する。

コマンド	マニュアルページ	説明
du	du(1)	各サブディレクトリに割り当てられたディスク容量を表示する。
find -size	find(1)	-size オプションで指定したサイズに基づいて、ディレクトリを再帰的に検索する。
ls -lh	ls(1)	ファイルのサイズを 1024 の累乗でリストする。

ファイルとディスク容量の情報を表示する方法

次のように df コマンドを入力して、ディスク容量の利用状況に関する情報を表示します。

```
$ df [directory] [-h] [-t] [-t]
```

df	オプションを指定しない場合、マウントされている全ファイルシステム、それらの装置名、使用されている 512 バイトのブロック数、ファイル数のリストを表示する
directory	チェックするファイルシステムのディレクトリを指定する
-h	ディスク容量を 1024 の累乗で表示する
-t	マウントされている全ファイルシステムの合計ブロック数と使用されているブロック数を表示する

例 — ファイルサイズおよびディスク容量に関する情報を表示する

次の例では、/usr/dist を除き、すべてのファイルシステムがローカルでマウントされています。/usr/dist はシステム venus からリモートにマウントされています。

```
$ df
/                (/dev/dsk/c0t0d0s0 ): 287530 blocks   92028 files
/usr             (/dev/dsk/c0t0d0s6 ): 1020214 blocks  268550 files
/proc           (/proc                ):      0 blocks     878 files
/dev/fd         (fd                   ):      0 blocks     0 files
/etc/mnttab     (mnttab               ):      0 blocks     0 files
/var/run        (swap                 ): 396016 blocks   9375 files
/tmp            (swap                 ): 396016 blocks   9375 files
/opt            (/dev/dsk/c0t0d0s5 ): 381552 blocks   96649 files
/export/home    (/dev/dsk/c0t0d0s7 ): 434364 blocks   108220 files
/usr/dist       (venus:/usr/dist      ):14750510 blocks 2130134 files
```


次の例は、ファイルシステム情報が1024バイト単位で表示されています。

```
$ df -h
Filesystem      size  used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0 1.9G   58M   1.8G     4%      /
/dev/dsk/c0t0d0s6 2.5G  765M   1.7G     1%     /usr
/proc           0K     0K     0K     0%     /proc
fd              0K     0K     0K     0%     /dev/fd
mnttab          0K     0K     0K     0%     /etc/mnttab
swap            2.1G   24K   2.1G     1%     /var/run
swap            2.1G    0K   2.1G     0%     /tmp
/dev/dsk/c0t0d0s5 1.9G   12M   1.8G     1%     /opt
/dev/dsk/c0t0d0s7 1.9G   10K   1.8G     1%     /export/home
```

注- /proc と /tmp は、ローカルファイルシステムですが、UFS ファイルシステムではありません。 /proc は PROCFS ファイルシステムであり、 /var/run と /tmp は TMPFS ファイルシステムであり、 /etc/mnttab は MNTFS ファイルシステムです。

次の例は、マウントされているすべてのファイルシステム、装置名、使用されている512バイトブロックの合計数、ファイル数を示しています。2行構成の各エントリの2行目は、それぞれのファイルシステムに割り当てられているブロックの合計数とファイルの合計数を示します。

```
$ df -t
/                (/dev/dsk/c0t0d0s0 ): 287530 blocks  92028 files
                  total: 385614 blocks  96832 files
/usr             (/dev/dsk/c0t0d0s6 ): 1020214 blocks 268550 files
                  total: 2381102 blocks 300288 files
/proc           (/proc           ): 0 blocks      879 files
                  total: 0 blocks      924 files
/dev/fd         (fd              ): 0 blocks      0 files
                  total: 0 blocks      72 files
/etc/mnttab     (mnttab         ): 0 blocks      0 files
                  total: 0 blocks      1 files
/var/run       (swap           ): 396112 blocks  9375 files
                  total: 396112 blocks  9395 files
/tmp           (swap           ): 396112 blocks  9375 files
                  total: 396128 blocks  9395 files
/opt           (/dev/dsk/c0t0d0s5 ): 381552 blocks  96649 files
                  total: 385614 blocks  96832 files
/export/home   (/dev/dsk/c0t0d0s7 ): 434364 blocks 108220 files
                  total: 434382 blocks 108224 files
/usr/dist      (venus:/usr/dist ): 14750510 blocks 2130134 files
                  total: 41225162 blocks 2482176 files
```

ファイルサイズの確認

ls コマンドを使用して、ファイルサイズを調べたりソートしたりできます。また、find コマンドを使用して、サイズの制限を超えているファイルを探することができます。詳細については、ls(1) と find(1) を参照してください。

▼ ファイルサイズを表示する方法

1. 確認したいファイルがあるディレクトリに移動します。
2. 次のように入力して、ファイルのサイズを表示します。

```
$ ls [-lh] [-s]
```

-l	長形式でファイルとディレクトリのリストを表示し、それぞれのサイズをバイト単位で示す(次の例を参照)
-h	ファイルやディレクトリのサイズが 1024 バイトより大きい場合、ファイルとディレクトリのサイズをキロバイト、メガバイト、ギガバイト、テラバイト単位で示す。このオプションは、-o、-n、-@、-g オプションで表示された出力を修正して、ファイルやディレクトリのサイズを新しい形式で表示する。詳細は ls(1) を参照してください。
-s	ファイルとディレクトリのリストを表示し、それぞれのサイズをブロック単位で示す

例 — ファイルサイズを表示する

次の例は、lastlog と messages が /var/adm ディレクトリ内のその他のファイルよりも大きいことを示します。

```
$ cd /var/adm
$ ls -lh
total 148
drwxrwxr-x  5 adm      adm      512 Nov 26 09:39 acct/
-rw-----  1 uucp     bin           0 Nov 26 09:25 aculog
drwxr-xr-x  2 adm      adm      512 Nov 26 09:25 exacct/
-r--r--r--  1 root     other    342K Nov 26 13:56 lastlog
drwxr-xr-x  2 adm      adm      512 Nov 26 09:25 log/
-rw-r--r--  1 root     root     20K Nov 26 13:55 messages
drwxr-xr-x  2 adm      adm      512 Nov 26 09:25 passwd/
drwxrwxr-x  2 adm      sys      512 Nov 26 09:39 sa/
drwxr-xr-x  2 root     sys      512 Nov 26 09:49 sm.bin/
-rw-rw-rw-  1 root     bin           0 Nov 26 09:25 spellhist
drwxr-xr-x  2 root     sys      512 Nov 26 09:25 streams/
```

```
-rw-r--r-- 1 root bin 3.3K Nov 26 13:56 utmpx
-rw-r--r-- 1 root root 0 Nov 26 10:17 vold.log
-rw-r--r-- 1 adm adm 19K Nov 26 13:56 wtmpx
```

次の例は、lpsched.1 ファイルが2ブロックを使用していることを示します。

```
$ cd /var/lp/logs
$ ls -s
total 2          0 lpsched          2 lpsched.1
```

▼ サイズの大きなファイルを見つける方法

1. サイズの大きなファイルを検索したいディレクトリに移動します。
2. 次のように入力して、ファイルのサイズをブロック単位に、最も大きいものから降順に表示します。

```
$ ls -s | sort -nr | more
```

```
sort -nr
```

ファイルのリストをブロックサイズの最も大きなものから降順に並べる

例 — サイズの大きなファイルを見つける

次の例では、lastlog と messages が /var/adm ディレクトリ内で最も大きなファイルです。

```
$ cd /var/adm
$ ls -s | sort -nr | more
48 lastlog
30 messages
24 wtmpx
18 pacct
8 utmpx
2 vold.log
2 sulog
2 sm.bin/
2 sa/
2 passwd/
2 pacct1
2 log/
2 acct/
0 spellhist
0 aculog
total 144
```

指定されたサイズ制限を超えるファイルを見つける方法

次のように `find` コマンドを使用して、指定したサイズを超えるファイルを見つけてファイル名を表示します。

```
$ find directory -size +nnn
```

<code>directory</code>	検索するディレクトリを指定する
<code>-size +<i>nnn</i></code>	512 バイトブロック数。このサイズを上回るファイルがリストされる

例 — 指定されたサイズ制限を超えるファイルを見つける

次の例は、作業中のカレントディレクトリ内の 400 ブロックを超えるファイルをどのように見つけるかを示します。

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
./Config/configPrintsetup.doc
./Config/configMailappx.doc
./Config/configMailconcepts.doc
./snapshot.rs
```

ディレクトリサイズの確認

`du` コマンドとそのオプションを使用してディレクトリのサイズを表示できます。さらに `quot` コマンドを使用すれば、ユーザーアカウントによって占められるローカル UFS ファイルシステム上のディスク容量のサイズを知ることができます。これらのコマンドについては、`du(1)` および `quot(1M)` のマニュアルページを参照してください。

ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する方法

次のように `du` コマンドを入力して、1つ以上のディレクトリ、サブディレクトリ、ファイルのサイズを表示します。サイズは 512 バイトブロック単位で表示されます。

```
$ du [-as] [directory ...]
```

<code>du</code>	指定した各ディレクトリとそれらの下の各サブディレクトリのサイズを合わせて表示する
<code>-a</code>	指定したディレクトリ内の各ファイルと各サブディレクトリのサイズと合計ブロック数を表示する
<code>-s</code>	指定したディレクトリ内の合計ブロック数を表示する
<code>-h</code>	各ディレクトリを 1024 バイト単位で表示する
<code>-H</code>	各ディレクトリを 1000 バイト単位で表示する
<code>directory ...</code>	調べたい 1 つ以上のディレクトリを指定する

例 — ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する

次の例は、2つのディレクトリの合計サイズを示しています。

```
$ du -s /var/adm /var/spool/lp
130    /var/adm
40     /var/spool/lp
```

次の例は、2つのディレクトリとそれらのすべてのサブディレクトリとファイルのサイズ、および各ディレクトリ内の合計ブロック数を表示します。

```
$ du /var/adm /var/spool/lp
2      /var/adm/log
2      /var/adm/passwd
2      /var/adm/acct/fiscal
2      /var/adm/acct/nite
2      /var/adm/acct/sum
8      /var/adm/acct
2      /var/adm/sa
2      /var/adm/sm.bin
130    /var/adm
4      /var/spool/lp/admins
2      /var/spool/lp/fifos/private
2      /var/spool/lp/fifos/public
6      /var/spool/lp/fifos
2      /var/spool/lp/requests/starbug
4      /var/spool/lp/requests
2      /var/spool/lp/system
```

```
2      /var/spool/lp/tmp/starbug
2      /var/spool/lp/tmp/.net/tmp/starbug
4      /var/spool/lp/tmp/.net/tmp
2      /var/spool/lp/tmp/.net/requests/starbug
4      /var/spool/lp/tmp/.net/requests
10     /var/spool/lp/tmp/.net
14     /var/spool/lp/tmp
40     /var/spool/lp
```

次の例は、ディレクトリのサイズを 1024 バイト単位で示しています。

```
du -h /usr/share/audio
796K  /usr/share/audio/samples/au
797K  /usr/share/audio/samples
798K  /usr/share/audio
```

▼ ローカル UFS ファイルシステムのユーザーの所有状況を表示する方法

1. スーパーユーザーになります。
2. 次のように入力して、ユーザー、ディレクトリまたはファイルシステム、**1024** バイト単位のブロック数を表示します。

```
# quot [-a] [filesystem]
```

-a	マウントされている各 UFS ファイルシステムの全ユーザーと 1024 バイト単位のブロック数を表示する
filesystem	UFS ファイルシステムを指定する。ユーザーと使用されているブロック数が表示される

注 - quot コマンドは、ローカル UFS ファイルシステムに対してだけ使用できません。

例 — ローカル UFS ファイルシステムのユーザーの所有状況を表示する

次の例では、ルート (/) ファイルシステムのユーザーが表示されています。次に、マウントされた全 UFS ファイルシステムのユーザーが表示されています。

```
# quot /
/dev/rdisk/c0t0d0s0:
43340  root
```

```
3142 rimmer
47 uucp
35 lp
30 adm
4 bin
4 daemon
# quot -a
/dev/rdsk/c0t0d0s0 (/):
43340 root
3150 rimmer
47 uucp
35 lp
30 adm
4 bin
4 daemon
/dev/rdsk/c0t0d0s6 (/usr):
460651 root
206632 bin
791 uucp
46 lp
4 daemon
1 adm
/dev/rdsk/c0t0d0s7 (/export/home):
9 root
```

古いファイルまたは使用されていないファイルの検索と削除

非常に負荷がかかったファイルシステムを整理する作業として、最近使用されていないファイルの特定と削除があります。使用されていないファイルは `ls` または `find` コマンドを使用して見つけることができます。詳細は、`ls(1)` と `find(1)` のマニュアルページを参照してください。

ディスク容量を節約するその他の方法としては、`/var/tmp` または `/var/spool` 内にあるような一時ファイルを空にしたり、`core` ファイルやクラッシュダンプファイルを削除したりするなどが含まれます。クラッシュダンプファイルの詳細については、[第 28 章](#)を参照してください。

最新ファイルのリストを表示する方法

次のように `ls -t` コマンドを使用して、最も新しく作成または変更されたファイルから順番にファイルのリストを表示します。

```
$ ls -t [directory]
```

-t	最新タイムスタンプのファイルを最初にしてソートする
directory	検索するディレクトリを指定する

例 — 最新ファイルのリストを表示する

次の例は、`ls -tl` コマンドを使用して、`/var/adm` ディレクトリ内で作成または変更された最新のファイルを検索する方法を示しています。 `sudo` が最も新しく作成または変更されたファイルです。

```
$ ls -tl /var/adm
total 134
-rw----- 1 root    root      315 Sep 24 14:00 sulog
-r--r--r-- 1 root    other    350700 Sep 22 11:04 lastlog
-rw-r--r-- 1 root    bin      4464 Sep 22 11:04 utmpx
-rw-r--r-- 1 adm     adm      20088 Sep 22 11:04 wtmpx
-rw-r--r-- 1 root    other    0 Sep 19 03:10 messages
-rw-r--r-- 1 root    other    0 Sep 12 03:10 messages.0
-rw-r--r-- 1 root    root     11510 Sep 10 16:13 messages.1
-rw-r--r-- 1 root    root     0 Sep 10 16:12 vold.log
drwxr-xr-x 2 root    sys      512 Sep 10 15:33 sm.bin
drwxrwxr-x 5 adm     adm      512 Sep 10 15:19 acct
drwxrwxr-x 2 adm    sys      512 Sep 10 15:19 sa
-rw----- 1 uucp    bin      0 Sep 10 15:17 aculog
-rw-rw-rw- 1 root    bin      0 Sep 10 15:17 spellhist
drwxr-xr-x 2 adm    adm      512 Sep 10 15:17 log
drwxr-xr-x 2 adm    adm      512 Sep 10 15:17 passwd
```

▼ 古いファイルと使用されていないファイルを見つけて削除する方法

1. スーパーユーザーになります。
2. 次のように入力して、指定した日数の間アクセスのないファイルを見つけて、ファイルにそれらのリストを書き込みます。

```
# find directory -type f[-atime + nnn] [-mtime + nnn] -print > filename
```

directory	検索するディレクトリを指定する このディレクトリの下にあるディレクトリも検索する。
-atime +nnn	指定した日数 (nnn) の間アクセスのないファイルを見つける
-mtime +nnn	指定した日数 (nnn) の間変更のないファイルを見つける

filename

使用されないファイルリストを書き込むファイルを指定する

3. 上の手順でリストに書き込んだ使用されていないファイルを削除します。

```
# rm `cat filename`
```

filename は、前の手順で作成した、使用されていないファイルのリストが入っているファイルです。

例 — 古いファイルと使用されていないファイルを見つけて削除する

次の例は、`/var/adm` ディレクトリとサブディレクトリ中の過去 60 日にわたってアクセスされていないファイルを示しています。`/var/tmp/deadfiles` ファイルには、使用されていないファイルのリストが含まれます。`rm` コマンドは、これらの使用されていないファイルを削除します。

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
# rm `cat /var/tmp/deadfiles`
#
```

▼ 一時ディレクトリを一度にクリアする方法

1. スーパーユーザーになります。
2. 整理するディレクトリに移動します。

```
# cd directory
```



注意 – 手順 3 を実行する前に、正しいディレクトリにいることを確認してください。手順 3 はカレントディレクトリ内のすべてのファイルを削除します。

3. カレントディレクトリ内のファイルとサブディレクトリを削除します。

```
# rm -r *
```

4. 不要な一時サブディレクトリやファイル、または古いサブディレクトリやファイルがある他のディレクトリに移動して、手順 3 を繰り返してサブディレクトリやファイルを削除します。

例 — 一時ディレクトリを一度にクリアする

次の例は、どのように mywork ディレクトリを整理するかを示し、すべてのファイルとサブディレクトリが削除されたことを確認する方法を示しています。

```
# cd mywork
# ls
filea.000
fileb.000
filec.001
# rm -r *
# ls
#
```

▼ core ファイルを見つけて削除する方法

1. スーパーユーザーになります。
2. **core** ファイルを検索したいディレクトリに移動します。
3. 次のように入力して、ディレクトリとサブディレクトリ内のすべての **core** ファイルを見つけて削除します。

```
# find . -name core -exec rm {} \;
```

例 — core ファイルを見つけて削除する

次の例は、どのように find コマンドを使用して jones のユーザーアカウントから core ファイルを見つけて削除するかを示します。

```
# cd /home/jones# find . -name core -exec rm {} \;
```

▼ クラッシュダンプファイルを削除する方法

クラッシュダンプファイルは非常に大きくなる可能性があります。したがって、必要以上に長期間保存しないでください。

1. スーパーユーザーになります。

2. 次のように入力して、クラッシュダンプファイルが格納されているディレクトリに移動します。

```
# cd /var/crash/system
```

system は、クラッシュダンプファイルを作成したシステムのことです。



注意 – 手順 3 を実行する前に、正しいディレクトリにいることを確認してください。手順 3 はカレントディレクトリ内のすべてのファイルを削除します。

3. クラッシュダンプファイルを削除します。

```
# rm *
```

4. クラッシュダンプファイルが削除されていることを確認します。

```
# ls
```

例 — クラッシュダンプファイルを削除する

次の例は、システム *venus* からどのようにクラッシュダンプファイルを削除するかと、クラッシュダンプファイルが削除されているかを確認する方法を示します。

```
# cd /var/crash/venus
```

```
# rm *
```

```
# ls
```


第 17 章

ディスク割り当て制限の管理 (手順)

この章では、ディスクの割り当て制限を設定し管理する方法を示します。この章で説明する手順は次のとおりです。

- 257 ページの「割り当て制限用にファイルシステムを構成する方法」
- 258 ページの「1 ユーザーに割り当て制限を設定する方法」
- 259 ページの「複数ユーザーに割り当て制限を設定する方法」
- 260 ページの「割り当て制限の整合性を確認する方法」
- 261 ページの「割り当て制限を有効にする方法」
- 262 ページの「割り当て制限を超過したかどうかを確認する方法」
- 263 ページの「ファイルシステムの割り当て制限を確認する方法」
- 264 ページの「弱い期間制限値のデフォルトを変更する方法」
- 265 ページの「1 ユーザーの割り当て制限を変更する方法」
- 266 ページの「1 ユーザーの割り当て制限を無効にする方法」
- 267 ページの「割り当て制限を無効にする方法」

ディスクの割り当て制限

ディスクの割り当て機能を使用することにより、システム管理者は、各ユーザーが使用できるディスク容量と i ノード数 (おおよそのファイル数に該当) を制限して UFS ファイルシステムを制御できます。このため、ディスク割り当て制限は、特に、ユーザーのホームディレクトリが存在するファイルシステムで便利です。公開ファイルシステムと /tmp ファイルシステムについてはこのディスク割り当て機能による利点はあまりありません。

ディスク割り当て制限の使用

一度割り当て制限を設定しても、それを変更して、ユーザーが使用できるディスク容量とiノード数を調整できます。また、システムに変更が必要な場合は、それに合わせて割り当て制限を追加または削除できます。ディスク割り当て制限、またはディスク割り当て制限を超えることができる時間を変更する手順、個々のディスク割り当て制限を無効にする手順、あるいはディスク割り当て制限をファイルシステムから削除する手順については、264 ページの「割り当て制限の変更と削除」を参照してください。

ディスク割り当て制限を監視できます。割り当て制限コマンドを使用することによりシステム管理者は、ファイルシステムでの割り当て制限を表示したり、割り当て制限を超えて使用しているユーザーを検索したりできます。これらのコマンドの使用方法については、261 ページの「割り当て制限のチェック」を参照してください。

ディスク割り当て制限の弱い制限値と強い制限値の設定

弱い制限値と強い制限値の両方を設定できます。システムは、ユーザーが自分の強い制限値を超えることを許可しません。しかし、システム管理者は、ユーザーが一時的に超えることができる、弱い制限値を設定できます。弱い制限値は、強い制限値より小さくなければなりません。

いったんユーザーが弱い制限値を超えると、タイマーが起動します。タイマーが動いている間、ユーザーは弱い制限値を超えて操作できます。しかし、強い制限値は超えることができません。再びユーザーが弱い制限値を下回ると、タイマーはリセットされます。しかし、タイマーが期限切れになったときに、まだユーザーの使用率が弱い制限値を超えていた場合、弱い制限値は、強い制限値として実施されます。デフォルトでは、弱い制限値のタイマーは7日です。

`repquota` コマンドと `quota` コマンドの `timeleft` フィールドは、タイマーの値を示します。

たとえば、あるユーザーの弱い制限値が 10,000 ブロックで、強い制限値が 12,000 ブロックであると仮定します。そのユーザーのブロック使用率が 10,000 ブロックを超えて、7日間のタイマーも期限切れになった場合、そのユーザーは自分の使用率が弱い制限値を下回るまで、それ以上のディスクブロックをそのファイルシステム上に割り当ててはできません。

ディスクブロックとファイル制限の相違

ファイルシステムは、ユーザーに2つの資源を提供します。データ用のブロックと、ファイル用のiノードです。各ファイルは、1つのiノードを使用します。ファイルデータは、データブロック内に格納されます(通常は、1Kバイトブロックで構成される)。

ディレクトリがなくても、ユーザーは空のファイルを作成することによって(ブロックを使用することなく)、自分のiノードディスク割り当て制限を超過することができます。また、ユーザーは、ユーザーの割り当て制限のデータブロックをすべて消費するのに十分な大きさのファイルを1つ作成して、自分のブロックのディスク割り当て制限を超過したiノードを1つ使用することができます。

ディスク割り当て制限の設定

割り当て制限を設定する一般的な手順は次のとおりです。

1. 割り当て制限オプションを/etc/vfstabファイルエントリに追加し、quotasファイルをファイルシステムの最上位ディレクトリに作成して、システムをリブートするたびにディスクの割り当て制限が適用されるようにします。
2. まず1人のユーザー用にディスク割り当て制限を作成したら、それをプロトタイプとして他のユーザー用にコピーできます。
3. ディスク割り当て制限を有効にする前に、その割り当て制限と現在のディスク使用状況との整合性をチェックして、矛盾しないことを確認します。
4. 最後に、1つ以上のファイルシステムのディスク割り当て制限を有効にします。

上記手順の具体的な情報については、257ページの「割り当て制限の設定(作業マップ)」を参照してください。

次の表で、ディスク割り当て制限を設定するコマンドを説明します。

表 17-1 割り当て制限を行うコマンド

コマンド	作業	マニュアルページ
edquota	各ユーザーのiノード数とディスク容量に強い制限と弱い制限を設定する	edquota(1M)

表 17-1 割り当て制限を行うコマンド (続き)

コマンド	作業	マニュアルページ
quotacheck	マウントされている各 UFS ファイルシステムを調べ、ファイルシステムのディスク割り当て制限ファイルに格納されている情報とファイルシステムの現在のディスク使用状況を比較し、矛盾を明らかにする	quotacheck (1M)
quotaon	指定したファイルシステムの割り当て制限を有効にする	quotaon (1M)
quota	マウントされているファイルシステムのユーザーの割り当て制限を表示し、割り当て制限が正しく設定されていることを確認する	quota (1M)

割り当て制限設定のガイドライン

ユーザーの割り当て制限を設定する前に、各ユーザーに割り当てるディスク容量の大きさとファイル数を決定する必要があります。ファイルシステムの合計領域サイズを超えないようにする場合は、ファイルシステムの合計サイズをユーザー数に等分すればよいでしょう。たとえば、3人のユーザーが 100M バイトのスライスを共有し、それぞれが同じディスク容量のサイズを必要とする場合は、各ユーザーに 33M バイトずつ割り当てます。

すべてのユーザーがそれぞれに割り当て制限を押し上げることがないような環境では、割り当て制限の合計がファイルシステムの合計サイズを超えるように個々の割り当て制限を設定することも可能です。たとえば、3人のユーザーが 100M バイトのスライスを共有する場合は、各ユーザーに 40M バイトを割り当ててもよいということです。

あるユーザーについて `edquota` コマンドを使用して割り当て制限を決定すると、同じファイルシステム上の他のユーザーにも同じ割り当て制限プロトタイプとして利用できます。

割り当て制限を有効にする前に、まず UFS ファイルシステムの割り当て制限を構成し、各ユーザーの割り当て制限を設定して、`quotacheck` コマンドを実行することによって、現在のディスク使用状況と割り当て制限ファイル間の整合性をチェックします。また、システムのレポートの頻度が低い場合は、`quotacheck` コマンドを定期的に実行します。

`edquota` により設定した割り当て制限は、`quotaon` コマンドを使用して有効にしなければ強制的に設定されません。割り当て制限ファイルを正しく構成したら、システムがリブートし、そのファイルシステムがマウントされるたびに、割り当て制限は自動的に有効になります。

割り当て制限の設定 (作業マップ)

作業	説明	参照先
1. ファイルシステムの割り当て制限の構成	/etc/vfstab を編集して、ファイルシステムがマウントされるたびに割り当て制限が有効になるようにする。また、quotas ファイルを作成する	257 ページの「割り当て制限用にファイルシステムを構成する方法」
2. 1 ユーザー用の割り当て制限の設定	edquota を使用して 1 ユーザーアカウント用にディスクと i ノードの割り当て制限を行う	258 ページの「1 ユーザーに割り当て制限を設定する方法」
3. (省略可能) 複数ユーザーの割り当て制限の設定	edquota コマンドを使用して、その他のユーザーアカウント用にプロトタイプの割り当て制限を適用する	259 ページの「複数ユーザーに割り当て制限を設定する方法」
4. 整合性のチェック	quotacheck を使用して、1 つまたは複数のファイルシステムの整合性について、現在の使用状況とディスクの割り当て制限を比較する	260 ページの「割り当て制限の整合性を確認する方法」
5. 割り当て制限を有効にする	quotaon を使用して、1 つまたは複数のファイルシステムの割り当て制限を有効にする	261 ページの「割り当て制限を有効にする方法」

▼ 割り当て制限用にファイルシステムを構成する方法

1. スーパーユーザーになります。
2. /etc/vfstab ファイルを編集します。割り当て制限を設定しようとする各 UFS ファイルシステムの「mount options」フィールドに **rq** を追加します。
3. 割り当て制限を格納しようとするファイルシステムのルートディレクトリに移動します。
4. 次のように入力して、**quotas** というファイルを作成します。

```
# touch quotas
```
5. **root** にのみ、読み取り権／書き込み権を与えます。

```
# chmod 600 quotas
```

次に進む手順

次の表を見て、次に進む手順を決めてください。

作業	参照箇所
1 ユーザーまたは複数ユーザー用の割り当て制限を設定する	258 ページの「1 ユーザーに割り当て制限を設定する方法」 259 ページの「複数ユーザーに割り当て制限を設定する方法」
既存ファイルを持つファイルシステム上に割り当て制限を設定する場合は、quotacheck コマンドを実行する	260 ページの「割り当て制限の整合性を確認する方法」
ファイルシステムの割り当て制限を有効にします。	261 ページの「割り当て制限を有効にする方法」

例 — 割り当て制限用にファイルシステムを構成する

次の /etc/vfstab の例は、システム pluto の /export/home ディレクトリが、ローカルシステムの NFS ファイルシステムとしてマウントされていることを示しています。割り当て制限が使用可能であるかは、mount options 列の下の rq エントリによってわかります。

```
#device          device  mount      FS   fsck  mount  mount
#to mount        to fsck  point      type pass  at boot options
#
pluto:/export/home -      /export/home nfs   -     yes   rq
```

次の例は /etc/vfstab の内容で、割り当て制限が有効 (mount options 列の rq エントリで示される) なローカル UFS ファイルシステムが /work ディレクトリにマウントされていることを示しています。

```
#device          device          mount  FS   fsck  mount  mount
#to mount        to fsck          point  type pass  at boot options
#
/dev/dsk/c0t4d0s0 /dev/rdisk/c0t4d0s0 /work ufs  3     yes   rq
```

▼ 1 ユーザーに割り当て制限を設定する方法

1. スーパーユーザーになります。
2. 割り当て制限エディタを使用して、**quotas** ファイルがファイルシステムのルートディレクトリにある各マウント済み **UFS** ファイルシステムに対して、1 行の割り当て制限情報を含む一時ファイルを作成します。

```
# edquota username
```

username は、割り当て制限を設定するユーザーです。

3. **1K** バイトディスクブロック数の弱い制限値と強い制限値、および **i** ノード数の弱い制限値と強い制限値を、それぞれ **0** (デフォルト) から各ファイルシステム用に指定されている割り当て値に変更します。

4. ユーザーの割り当て制限を確認します。

```
# quota -v username
```

```
-v                ディスク割り当て制限がある、マウント済みのファイルシステム上の、ユーザーのディスク割り当て制限情報を表示する  
  
username         ディスク割り当て制限を表示するユーザー名を指定する
```

例 — 1 ユーザーに割り当て制限を設定する

次の例は、`edquota` コマンドで開いた一時ファイルの内容を示しています。このシステムでは、ルートディレクトリに `quotas` ファイルが含まれているマウント済みファイルシステムは、`/files` だけです。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

次の例は、割り当て制限設定後の一時ファイルの上と同じ行を示しています。

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

▼ 複数ユーザーに割り当て制限を設定する方法

1. スーパーユーザーになります。
2. 次のように割り当て制限エディタを使用して、すでにプロトタイプユーザー用に設定した割り当て制限を指定するその他のユーザーに適用します。

```
# edquota -p prototype-user username ...
```

```
prototype-user   すでに割り当て制限を設定してあるアカウントのユーザー名  
  
username ..      1人以上の追加アカウントのユーザー名を指定する
```

例 — 複数ユーザーにプロトタイプ割り当て制限を設定する

次の例は、ユーザー `bob` に設定された割り当て制限をユーザー `mary` と `john` に適用する方法を示しています。

```
# edquota -p bob mary john
```

▼ 割り当て制限の整合性を確認する方法

システムをリブートすると、quotacheck コマンドが自動的に実行されます。通常、割り当て制限を設定したファイルシステムが空の場合、quotacheck コマンドを実行する必要はありません。ただし、既存ファイルを持つファイルシステム上に割り当て制限を設定した場合は、quotacheck コマンドを実行して、既存ファイルや既存 i ノードと割り当て制限データベースとを同期させる必要があります。

また、quotacheck コマンドを大規模ファイルシステムで実行するには、時間がかかることを考慮してください。

注 - ディスクのデータの正確さを保つには、quotacheck コマンドを実行するとき、チェックするファイルシステムが他のユーザーによって使用できないようにしてください。

1. スーパーユーザーになります。
2. 次のように UFS ファイルシステム上の整合性チェックを実行します。

```
# quotacheck [-va] filesystem
```

-v	(省略可能) 特定のファイルシステム上の各ユーザーのディスク割り当て制限を示す
-a	/etc/vfstab ファイルに rq エントリがある全ファイルシステムをチェックする
filesystem	チェックするファイルシステムを指定する

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

例 — 割り当て制限の整合性を確認する

次の例は、スライス /dev/rdisk/c0t0d0s7 上の /export/home ファイルシステムのディスク割り当て制限をチェックする方法を示しています。/export/home ファイルシステムは、/etc/vfstab ファイルに rq エントリを持つ、唯一のファイルシステムです。

```
# quotacheck -va
*** Checking quotas for /dev/rdisk/c0t0d0s7 (/export/home)
```

▼ 割り当て制限を有効にする方法

1. スーパーユーザーになります。
2. ファイルシステムの割り当て制限を有効にします。

```
# quotaon [-v] -a filesystem ...
```

-v	割り当て制限が無効にされた場合、各ファイルシステムからメッセージを表示する
-a	/etc/vfstab ファイル内に <code>rq</code> エントリがある全ファイルシステムの割り当て制限を有効にする
filesystem ...	指定する 1 つ以上のファイルシステムの割り当て制限を有効にする スペースでファイル名を区切って、複数のファイルシステムを指定する

例 — 割り当て制限を有効にする

次の例は、スライス `/dev/dsk/c0t4d0s7` と `/dev/dsk/c0t3d0s7` 上のファイルシステムのディスク割り当て制限を有効にする方法を示しています。

```
# quotaon -v /dev/dsk/c0t4d0s7 /dev/dsk/c0t3d0s7
/dev/dsk/c0t4d0s7: quotas turned on
/dev/dsk/c0t3d0s7: quotas turned on
```

割り当て制限のチェック

ディスクと i ノードの割り当て制限を設定して有効にしたら、それらの割り当て制限を超過して使用しているユーザーをチェックできます。また、ファイルシステム全体の割り当て制限情報をチェックすることもできます。

次の表で、ディスク割り当て制限をチェックするコマンドを説明します。

表 17-2 割り当て制限をチェックするコマンド

コマンド	作業
<code>quota</code>	ユーザー割り当て制限と現在のディスク使用量を表示する。ユーザーの割り当て制限超過使用量も表示可能
<code>repquota</code>	指定されたファイルシステムの割り当て制限、ファイル、および所有しているディスク容量を表示する

▼ 割り当て制限を超過したかどうかを確認する方法

quota コマンドを使用して、割り当て制限が適用されているファイルシステム上の個々のユーザーの割り当て制限とディスク使用量を表示できます。

1. スーパーユーザーになります。
2. 次のように入力して、割り当て制限が有効にされているマウント済みファイルシステムのユーザー割り当て制限を表示します。

```
# quota [-v] username
```

-v	割り当て制限が設定されているマウント済みファイルシステムすべてについてユーザー割り当て制限を表示する
username	ユーザーアカウントのユーザー名またはユーザー ID (UID)

例 — 割り当て制限を超過したかどうかを確認する

次の例は、UID 301 によって識別されるユーザーアカウントに 1K バイトの割り当て制限が設定されているが、ディスク容量をまったく使用していないことを示しています。

```
# quota -v 301
Disk quotas for bob (uid 301):
Filesystem  usage  quota limit  timeleft  files  quota  limit  timeleft
/export/home  0      1      2          0      2      3
```

Filesystem	ファイルシステムのマウントポイント
usage	現在のブロック使用数
quota	弱いブロック制限値
limit	強いブロック制限値
timeleft	ディスク割り当て制限タイマーの残り時間 (日単位)
files	現在の i ノード使用数
quota	弱い i ノード制限値
limit	強い i ノード制限値
timeleft	ディスク割り当て制限タイマーの残り時間 (日単位)

▼ ファイルシステムの割り当て制限を確認する方法

repquota コマンドを使用して1つ以上のファイルシステム上のすべてのユーザーの割り当て制限とディスク使用量を表示します。

1. スーパーユーザーになります。
2. ディスクがまったく使用されていないなくても、1つまたは複数のファイルシステムのすべての割り当て制限を表示します。

```
# repquota [-v] -a filesystem
```

-v	資源を消費していないユーザーも含めて、すべてのユーザーのディスク割り当て制限を報告する
-a	すべてのファイルシステムについて報告する
filesystem	指定したファイルシステムについて報告する

例 — ファイルシステムの割り当て制限を確認する

次の例は、割り当て制限が1つのファイルシステム (/export/home) だけに対して有効なシステムでの repquota コマンドからの出力を示しています。

```
# repquota -va
/dev/dsk/c0t3d0s7 (/export/home):
      Block limits                File limits
User      used  soft  hard  timeleft  used  soft  hard  timeleft
#301  --      0    1    2.0 days    0    2    3
#341  --    57   50   60   7.0 days  2    90  100
```

Block limits

used	現在のブロック使用数
soft	弱いブロック制限値
hard	強いブロック制限値
timeleft	ディスク割り当て制限タイマーの残り時間(日単位)

File limits

used	現在のiノード使用数
soft	弱いiノード制限値
hard	強いiノード制限値

timeleft

ディスク割り当て制限タイマーの残り時間 (日単位)

割り当て制限の変更と削除

割り当て制限を変更して、ユーザーが使用するディスク容量と i ノード数を調整できます。または、必要に応じて各ユーザーから、あるいはファイルシステム全体から割り当て制限を削除できます。

次の表で、割り当て制限を変更または削除するのに使用するコマンドを示します。

表 17-3 割り当て制限を変更または削除するコマンド

コマンド	マニュアルページ	説明
edquota	edquota	各ユーザーについて i ノード数とディスク容量の強い制限と弱い制限を変更する。また、任意のユーザーが弱い制限値を超えることが許される期間の長さを変更する
quotaoff	quotaoff (1M)	指定したファイルシステムの割り当て制限を無効にする

▼ 弱い期間制限値のデフォルトを変更する方法

デフォルトでは、ユーザーはある週の割り当て制限に対する弱い制限値を 1 週間超えることができます。弱い制限値を 1 週間よりも長く超えると、システムはそのユーザーに対し、i ノードとディスクブロックの使用を禁止します。

edquota コマンドを使用すると、この割り当て制限の期間制限を変更できます。

1. スーパーユーザーになります。
2. 次のように割り当て制限エディタを使用して、弱い期間制限値を含む一時ファイルを作成します。

```
# edquota -t
-t オプションは、各ファイルシステムの弱い期間制限値を指定します。
```

3. 期間制限を、0 (デフォルト) から数値とキーワード **month**、**week**、**day**、**hour**、**min**、または **sec** を使用して指定する値に変更します。

注 - この手順は、現在のディスク割り当て制限違反者には影響しません。

例 — 期間の弱い制限値のデフォルトを変更する

次の例は、`/export/home` がただ 1 つの `quotas` を持つマウント済みファイルシステムであるシステムで `edquota` コマンドによって開かれた一時ファイルの内容を示しています。デフォルト値 0 は、デフォルトで 1 週間の期間制限値が使用されることを意味します。

```
fs /export/home blocks time limit = 0 (default), files time limit = 0 (default)
```

次の例は、ブロック割り当て制限の超過に対する期間制限値が 2 週間に変更され、ファイル数の超過に対する期間制限値が 16 日に変更された後の、上の例と同じ一時ファイルの内容を示しています。

```
fs /export/home blocks time limit = 2 weeks, files time limit = 16 days
```

▼ 1 ユーザーの割り当て制限を変更する方法

1. スーパーユーザーになります。
2. 次のように割り当て制限エディタを使用して、`quotas` ファイルがそれぞれのファイルシステムのルートディレクトリにある各マウント済みファイルシステムに対して 1 行ずつエントリが入っている一時ファイルを開きます。

```
# edquota username
```

`username` は、割り当て制限を変更したいユーザー名を指定します。



注意 - `edquota` コマンドの引数として複数のユーザーを指定できますが、表示される情報にはどのユーザーのものなのか示されないで、混乱を招く恐れがあります。

3. **1K** バイトディスクブロック数の弱い制限値と強い制限値、および **i** ノード数の弱い制限値と強い制限値を入力します。
4. ユーザーの割り当て制限が正しく変更されたことを確認します。

```
# quota -v username
```

`-v` ディスク割り当て制限が有効にされている、すべてのマウント済みのファイルシステムについて、ユーザーのディスク割り当て制限情報を表示する

`username` 割り当て制限をチェックしたいユーザー名を指定する

例 — 1 ユーザーの割り当て制限を変更する

次の例は、`edquota` コマンドで開いた一時ファイルの内容を示しています。このシステムでは、ルートディレクトリに `quotas` ファイルが含まれているマウント済みファイルシステムは、`/files` だけです。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

次の例は、上と同じ一時ファイルの割り当て制限変更後の内容を示しています。

```
fs /files blocks (soft = 0, hard = 500) inodes (soft = 0, hard = 100)
```

次の例は、ユーザー `smith` の強い制限値の変更結果を確認する方法と、1K バイトブロック数と i ノード数の強い制限値がそれぞれ 500 と 100 に変更されていることを示しています。

```
# quota -v smith
Disk quotas for smith (uid 12):
Filesystem  usage  quota  limit  timeleft  files  quota  limit  timeleft
/files      1      0      500      1      0      100
```

▼ 1 ユーザーの割り当て制限を無効にする方法

1. スーパーユーザーになります。
2. 次のように割り当て制限エディタを使用して、`quotas` ファイルがその最上位ディレクトリにある各マウント済みファイルシステムに対して 1 行の割り当て制限情報を含む一時ファイルを作成します。

```
# edquota username
```

`username` は、割り当て制限を無効にしたいユーザー名を指定します。



注意 `edquota` コマンドの引数として複数のユーザーを指定できますが、表示される情報にはどのユーザーのものなのか示されないため、混乱を招く恐れがあります。

3. 1K バイトディスクブロック数の弱い制限値と強い制限値、および i ノード数の弱い制限値と強い制限値を 0 (ゼロ) に変更します。

注 - 必ずこれらの値を 0 (ゼロ) に変更してください。テキストファイルから行を削除してはいけません。

4. ユーザーの割り当て制限を無効にしたことを確認します。

```
# quota -v username
```

<code>-v</code>	ディスク割り当て制限が有効にされている、すべてのマウント済みのファイルシステムについて、ユーザーのディスク割り当て制限情報を表示する
<code>username</code>	割り当て制限を確認したいユーザー名 (UID) を指定する

例 — 1 ユーザーの割り当て制限を無効にする

次の例は、`edquota` コマンドで開いた一時ファイルの内容を示しています。このシステムでは、ルートディレクトリに `quotas` ファイルが含まれているマウント済みファイルシステムは、`/files` だけです。

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

次の例は、割り当て制限を無効にした後の上と同じ一時ファイルの内容を示しています。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

▼ 割り当て制限を無効にする方法

1. スーパーユーザーになります。
2. ファイルシステムの割り当て制限を無効にします。

```
# quotaoff [-v] -a filesystem ...
```

<code>-v</code>	割り当て制限が無効にされた場合、各ファイルシステムからメッセージを表示する
<code>-a</code>	全ファイルシステムの割り当て制限を無効にする
<code>filesystem</code>	指定する 1 つ以上のファイルシステムの割り当て制限を無効にする スペースでファイル名を区切って、複数のファイルシステムを指定する

例—割り当て制限を無効にする

次の例は、`/export/home` ファイルシステムの割り当て制限を無効にする方法を示しています。

```
# quotaoff -v /export/home  
/export/home: quotas turned off
```

第 18 章

システムタスクのスケジュール設定 (手順)

この章では、`crontab` コマンドおよび `at` コマンドを使用して、定型作業や 1 度限りのシステムタスクをスケジュールリングする方法を説明します。さらに、`cron.deny`、`cron.allow`、`at.deny` の各ファイルを使用して、これらのコマンドの使用を制御する方法も説明します。

この章で説明する手順は次のとおりです。

- 274 ページの「`crontab` ファイルを作成または編集する方法」
- 276 ページの「`crontab` ファイルを表示する方法」
- 277 ページの「`crontab` ファイルを削除する方法」
- 279 ページの「`crontab` の使用を拒否する方法」
- 279 ページの「`crontab` の使用を特定のユーザーに限定する方法」
- 282 ページの「`at` ジョブを作成する方法」
- 283 ページの「`at` 待ち行列を表示する方法」
- 283 ページの「`at` ジョブを表示する方法」
- 284 ページの「`at` ジョブを削除する方法」
- 285 ページの「`at` コマンドの使用を拒否する方法」

システムタスクを自動的に実行する方法

たくさんのシステムタスクを設定して、自動的に実行することができます。これらのタスクの中には、定期的な実行が必要になる作業があります。また、夜間や週末などの就業時間外に 1 回だけ実行する作業もあります。

この節では、`crontab` と `at` という 2 つのコマンドについて概説します。これらのコマンドを使用すると、定型作業をスケジュールして、自動的に実行することができます。`crontab` は、繰り返し実行するコマンドをスケジュールします。`at` コマンドは、1 回だけ実行するタスクをスケジュールします。

次の表は、crontab コマンドと at コマンド、これらのコマンドの使用を制御できるファイルをまとめたものです。

表 18-1 コマンドの要約: システムタスクのスケジューリング

コマンド	スケジューリングの対象	ファイルの格納場所	制御ファイル
crontab	一定間隔で実行する複数のシステムタスク	/var/spool/cron/crontabs	/etc/cron.d/cron.allow /etc/cron.d/cron.deny
at	1つのシステムタスク	/var/spool/cron/atjobs	/etc/cron.d/at.deny

Solaris 管理コンソールのスケジュールされたジョブツールを使用して、定型作業をスケジュールすることもできます。Solaris 管理コンソールの使用と起動の方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールの操作 (手順)」を参照してください。

繰り返されるジョブのスケジューリング (crontab)

定型的なシステム管理タスクは、crontab コマンドを使用して、毎日、毎週、または毎月それぞれ 1 回ずつ実行するようにスケジュールできます。

毎日 1 回の crontab によるシステム管理作業には、次のようなものがあります。

- 作成後、数日以上経過したファイルを一時ディレクトリから削除する
- アカウンティング要約コマンドを実行する
- df および ps コマンドを使用してシステムのスナップショットを取る
- 日常のセキュリティ監視を実行する
- システムのバックアップを実行する

毎週 1 回の crontab システム管理作業には、次のようなものがあります。

- man -k コマンドで処理する catman データベースを再構築する
- fsck -n を実行して存在するディスク問題のリストを表示する

毎月 1 回の crontab システム管理作業には、次のようなものがあります。

- 指定月に使用されなかったファイルをリストする
- 月次アカウンティングレポートを生成する

上記に加えて、連絡事項の通知の転送やバックアップファイルの削除や、さらに他の定型的システム作業を実行するように crontab コマンドをスケジュールすることもできます。

crontab ジョブをスケジューリングする手順については、274 ページの「[crontab ファイルを作成または編集する方法](#)」を参照してください。

1 つのジョブのスケジューリング (at)

特定の 1 つのジョブを後で実行するように at コマンドを使用してスケジュールできます。ジョブは 1 つのコマンドやスクリプトで構成されます。

crontab コマンドと同様に、at コマンドは定型作業の自動実行をスケジュールします。しかし、crontab ファイルとは異なり、at ファイルはそれぞれのタスクを 1 回実行して、その後はディレクトリから削除されてしまいます。したがって、at はそれぞれ単純なコマンドまたはスクリプトを実行して、後で調べることができるようにそれらの出力を別々のファイルに送るように使用するのが最も効果的です。

at ジョブの実行を指定するには、単にコマンド構文に従って at オプションで実行時刻を指定してください。at ジョブの実行を指定する方法についての詳細は、281 ページの「[at コマンドの説明](#)」を参照してください。

at コマンドは、入力されたコマンドまたはスクリプトを、現在の環境変数のコピーと一緒に /var/spool/cron/atjobs ディレクトリに格納します。作成された at ジョブには、ファイル名として、at 待ち行列内での位置を指定する長い数値と .a 拡張子からなる、たとえば 793962000.a のような文字列が与えられます。

cron デーモンは、通常 15 分間隔で定期的に atrun プログラムを実行します。atrun プログラムは、スケジュールされた時間に at ジョブを実行します。cron デーモンが at ジョブを実行すると、at ジョブのファイルが atjobs ディレクトリから削除されます。

at ジョブをスケジューリングする手順については、282 ページの「[at ジョブを作成する方法](#)」を参照してください。

繰り返されるシステムタスクのスケジューリング (cron)

以降の各項で、crontab ファイルをどのように作成、編集、表示、削除するか、さらに、それらのファイルの使用をどのように制御するかを説明します。

crontab ファイルの内容

cron デーモンは、各 crontab ファイル内にあるコマンドに従ってシステムタスクをスケジュールします。crontab ファイルには、それぞれ一定間隔で実行されるコマンドが 1 行に 1 つずつ入っています。各行の先頭は cron デーモンが各コマンドを実行する日時情報です。

たとえば、SunOS ソフトウェアのインストール時に root という名前の crontab ファイルが提供されますが、このファイルの内容は次のとおりです。

```
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
```

最初の行は、毎日午前 3 時 10 分に logadm コマンドを実行します。2 行目は、毎週日曜日の午前 3 時 15 分に nfsfind スクリプトを実行します。3 行目は、毎日午前 2 時 10 分に夏時間をチェック (して必要に応じて修正) するスクリプトを実行します。RTC タイムゾーンも /etc/rtc_config ファイルもない場合、このエントリは何もしません。4 行目は、毎日午前 3 時 30 分に Generic Security Service テーブル /etc/gss/gsscred_db の重複エントリをチェック (重複エントリがある場合は削除) します。

crontab ファイル内のコマンド行の構文の詳細は、[273 ページの「crontab ファイルエントリの構文」](#)を参照してください。

crontab ファイルは、/var/spool/cron/crontabs ディレクトリに保存されます。SunOS ソフトウェアのインストール時には、root 以外にもいくつかの crontab ファイルが提供されます ([表 18-2](#) を参照してください)。

表 18-2 デフォルトの crontab ファイル

crontab ファイル	機能
adm	アカウントिंग
lp	印刷
root	一般的なシステム機能とファイルシステムの整理
sys	パフォーマンス情報の収集
uucp	一般的な uucp の整理

デフォルトの crontab ファイルの他に、ユーザーは crontab ファイルを作成してユーザー自身のシステムタスクをスケジュールできます。その他の crontab ファイルは、それらの中に作成されるユーザーのアカウントに基づいて、bob、mary、smith、jones などのように命名されます。

root またはほかのユーザーが所有する crontab ファイルを使用するには、スーパーユーザーの特権が必要です。

crontab ファイルの作成、編集、表示、削除の手順については、以降の節で説明します。

cron デーモンのスケジューリング

cron デーモンは crontab コマンドの自動スケジューリングを管理します。cron デーモンは、通常、15 分ごとに /var/spool/cron/crontab ディレクトリに crontab ファイルがあるかどうかをチェックします。cron デーモンは、新しい crontab ファイルがないか、既存の crontab が変更されていないかを確認し、いずれかがあった場合は、ファイル内のリストから実行時刻を読み取り、正しい時刻にコマンドを実行します。

ほとんど同様に、cron デーモンは /var/spool/cron/atjobs ディレクトリ内の at ファイルのスケジューリングを制御します。

crontab ファイルエントリの構文

crontab ファイルは、1 行に 1 つのコマンドが入った構成になっています。これらのコマンド行の最初の 5 つのフィールドには、コマンドが実行される時刻を指定し、それぞれスペースで区切ります。これら 5 つのフィールドを、次の表で説明します。

表 18-3 crontab 時刻フィールドの値

時刻フィールド	値
分	0-59
時	0-23
日	1-31
月	1-12
曜日	0-6 (0 は日曜日)

次に、crontab 時刻フィールドで特殊文字を使用する際のガイドラインを示します。

- 各フィールドはスペースで区切る
- 複数の値の間はコンマで区切る
- 値の範囲はハイフンを使用して指定する
- 取り得るすべての値を含むには、ワイルドカードとしてアスタリスクを使用する
- コメントまたは空白行を示すには、行の先頭にコメント記号 (#) を使用する

たとえば、次の crontab コマンドエントリは、毎月 1 日と 15 日の午後 4 時に、ユーザーのコンソールウィンドウに注意を促すメッセージを表示します。

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab ファイル内の各コマンドは、長くても 1 行内に入れなければなりません。これは、crontab が余分なキャリッジリターンを認識しないからです。crontab のエントリとコマンドオプションの詳細は、crontab(1) のマニュアルページを参照してください。

crontab ファイルの作成と編集

crontab ファイルを作成するもっとも簡単な方法は、crontab -e コマンドを使用することです。このコマンドは、使用するシステム環境で使用できる環境変数 EDITOR で指定されたテキストエディタを起動します。この環境変数が設定されていない場合は、crontab はデフォルトのエディタ ed を使用します。あらかじめ、使い慣れたエディタを選択しておく必要があります。

次の例は、エディタが定義されたかどうかを確認する方法と、vi をデフォルトのエディタとして設定する方法を示しています。

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

crontab ファイルを作成すると、自動的に /var/spool/cron/crontabs ディレクトリ内に格納され、作成者のユーザー名で命名されます。スーパーユーザー特権があれば、他のユーザーや root の crontab ファイルを作成または編集できます。

▼ crontab ファイルを作成または編集する方法

1. (省略可能) root または他のユーザーが所有する crontab ファイルを作成または編集する場合は、スーパーユーザーになります。
2. 次のように入力して、新しい crontab ファイルを作成するか、既存の crontab ファイルを編集します。

```
$ crontab -e [username]
```

username は、crontab ファイルを作成または編集するユーザーのアカウント名を指定します。他のユーザーの crontab ファイルを作成または編集するには、スーパーユーザーの権限が必要です。



注意 – 誤ってオプションを指定しないで `crontab` コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに `crontab` コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の `crontab` ファイルが空のファイルで上書きされます。

3. コマンド行をファイルに追加します。

273 ページの「[crontab ファイルエントリの構文](#)」に記載されている構文に従ってください。 `crontab` ファイルは `/var/spool/cron/crontabs` に格納されます。

4. `crontab` ファイルの変更箇所を確認します。

```
# crontab -l [username]
```

例 — `crontab` ファイルを作成または編集する

次の例は、他のユーザーのための `crontab` ファイルをどのように作成するかを示します。

```
# crontab -e jones
```

次の新しい `crontab` ファイルに追加されたコマンドは、毎週日曜日の午前 1 時にユーザーのホームディレクトリからすべてのログファイルを自動的に削除します。このコマンドエントリは出力先を変更しないので、出力先変更文字がコマンド行の `*.log` の後に追加されて、そのコマンドが正しく実行されるようにしています。

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

`crontab` ファイルを確認する方法

特定のユーザーの `crontab` ファイルがあるかどうかを確認するには、`/var/spool/cron/crontabs` ディレクトリで `ls -l` コマンドを使用します。たとえば、次の表示はユーザー `smith` と `jones` の `crontab` ファイルがあることを示しています。

```
$ ls -l /var/spool/cron/crontabs
-rw-r--r-- 1 root sys 190 Feb 26 16:23 adm
-rw----- 1 root staff 225 Mar 1 9:19 jones
-rw-r--r-- 1 root root 1063 Feb 26 16:23 lp
-rw-r--r-- 1 root sys 441 Feb 26 16:25 root
-rw----- 1 root staff 60 Mar 1 9:15 smith
-rw-r--r-- 1 root sys 308 Feb 26 16:23 sys
```

ユーザーの `crontab` ファイルの内容を確認するには、`crontab -l` コマンドを使用します。276 ページの「[crontab ファイルを表示する方法](#)」を参照してください。

crontab ファイルの表示

crontab -l コマンドは、cat コマンドが他のファイルタイプの内容を表示するのとまったく同様に、使用しているユーザーの crontab ファイルの内容を表示します。このコマンドを使用するために、ディレクトリを (crontab ファイルが入っている) /var/spool/cron/crontabs に変更する必要はありません。

デフォルトでは、crontab -l コマンドは自分自身の crontab ファイルを表示します。他のユーザーの crontab ファイルは、スーパーユーザーでなければ表示できません。

▼ crontab ファイルを表示する方法

1. (省略可能) **root** または他のユーザーの **crontab** ファイルを表示する場合は、スーパーユーザーになります。
2. 次のように入力して、**crontab** ファイルを表示します。

```
$ crontab -l [username]
```

username は、表示する crontab ファイルのユーザーのアカウント名を指定します。他のユーザーの crontab ファイルを表示するには、スーパーユーザーの権限が必要です。



注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに crontab コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の crontab ファイルが空のファイルで上書きされます。

例 — crontab ファイルを表示する

次の例で、どのように crontab -l コマンドを使用してユーザのデフォルトの crontab ファイル、root のデフォルトの crontab ファイル、他のユーザーの crontab ファイルを表示するかを示します。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
$ suPassword:
# crontab -l
#ident "@(#)root 1.19 98/07/06 SMI" /* SVr4.0 1.1.3.1 */
#
# The root crontab should be used to perform accounting data collection.
```

```
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

crontab ファイルの削除

デフォルトでは、`rm` コマンドを使用して誤って `crontab` ファイルを削除してしまうことがないように、`crontab` ファイルは保護されています。`crontab` ファイルを削除する場合は、`rm` コマンドではなく `crontab -r` コマンドを使用してください。

デフォルトでは、`crontab -r` は、このコマンドを実行したユーザーの `crontab` を削除します。`root` またはその他のユーザーの `crontab` を削除するには、スーパーユーザーにならなければなりません。

このコマンドを使用するには、ディレクトリを (`crontab` ファイルが入っている) `/var/spool/cron/crontabs` に変更する必要はありません。

▼ crontab ファイルを削除する方法

1. (省略可能) `root` または他のユーザーの `crontab` ファイルを削除するには、スーパーユーザーになります。
2. 次のように入力して、`crontab` ファイルを削除します。

```
$ crontab -r [username]
```

`username` は、削除する `crontab` ファイルのユーザーのアカウント名を指定します。他のユーザーの `crontab` ファイルを削除するには、スーパーユーザーの権限が必要です。



注意 - 誤ってオプションを指定しないで `crontab` コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに `crontab` コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の `crontab` ファイルが空のファイルで上書きされます。

3. `crontab` ファイルが削除されたことを確認します。

```
$ ls /var/spool/cron/crontabs
```

例 — crontab ファイルを削除する

次の例では、ユーザー smith が `crontab -r` コマンドを使用して自分の crontab ファイルを削除します。

```
$ ls /var/spool/cron/crontabs
adm    jones    lp      root    smith   sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm    jones    lp      root    sys     uucp
```

crontab コマンドの使用制御

`cron.deny` と `cron.allow` という `/etc/cron.d` ディレクトリ内の 2 つのファイルを使用して、`crontab` の使用を制御できます。これらのファイルによって、指定したユーザーだけが、それぞれ自分の `crontab` ファイルの作成、編集、表示、または削除などの `crontab` コマンドのタスクを実行できるようにします。

`cron.deny` および `cron.allow` ファイルは、それぞれ 1 行に 1 ユーザー名が入ったリストからなります。これらの使用制御用ファイルは、次のように連携して機能を果たします。

- `cron.allow` が存在する場合は、このファイルにリストされているユーザーだけが `crontab` ファイルを作成、編集、表示、または削除できます。
- `cron.allow` が存在しない場合は、`cron.deny` にリストされているユーザーを除くすべてのユーザーが `crontab` ファイルの実行を依頼できます。
- `cron.allow` も `cron.deny` も存在しない場合は、`root` 以外は `crontab` コマンドを実行できません。

`cron.deny` と `cron.allow` ファイルを編集または作成するには、スーパーユーザーの権限が必要です。

Solaris ソフトウェアのインストール時に作成される `cron.deny` ファイルには、次のユーザー名が含まれます。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

デフォルトの `cron.deny` ファイル内のユーザー名は、いずれも `crontab` コマンドを使用できません。このファイルを編集して、`crontab` コマンドの使用を拒否したいユーザー名を追加できます。

デフォルトでは、`cron.allow` ファイルは提供されません。つまり、Solaris ソフトウェアをインストールした直後は、すべてのユーザー (デフォルトの `cron.deny` ファイルにリストされているユーザーを除く) が `crontab` コマンドを使用できます。`cron.allow` ファイルを作成した場合、そのユーザーだけが `crontab` コマンドを使用できます。

▼ `crontab` の使用を拒否する方法

1. スーパーユーザーになります。
2. `/etc/cron.d/cron.deny` ファイルを編集し、`crontab` コマンドを使用させないユーザー名を 1 行に 1 つずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3. `/etc/cron.d/cron.deny` ファイルを確認します。

```
# cat /etc/cron.d/cron.deny
```

▼ `crontab` の使用を特定のユーザーに限定する方法

1. スーパーユーザーになります。
2. `/etc/cron.d/cron.allow` ファイルを作成します。
3. `root` ユーザー名を `cron.allow` ファイルに入力します。
このファイルに `root` を追加しないと、スーパーユーザーの `crontab` コマンドの使用が拒否されます。
4. 次のように、`crontab` コマンドの使用を許可するユーザー名を 1 行に 1 つずつ入力します。

```
root
username1
```

```
username2
username3
.
.
.
```

例 — crontab の使用を特定のユーザーに限定する

次は、ユーザー visitor、jones、temp に crontab を使用させない cron.deny ファイルの例です。

```
$ cat /etc/cron.d/cron.denydaemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

次は cron.allow ファイルの例です。ユーザー smith、jones、lp および root だけが、crontab コマンドを使用できます。

```
$ cat /etc/cron.d/cron.allow
root
jones
lp
smith
```

crontab の使用制限を確認する方法

特定のユーザーが crontab を使用できるかどうかを確認するには、そのユーザーのアカウントでログインして crontab -l コマンドを使用します。

```
$ crontab -l
```

そのユーザーが crontab を使用できて、すでに crontab ファイルを作成している場合は、その crontab ファイルが表示されます。そのユーザーが crontab を使用できるが、crontab ファイルがない場合は、次のようなメッセージが表示されます。

```
crontab:crontab ファイルを開けません
```

このユーザーは、cron.allow (が存在する場合) に含まれているか、cron.deny に含まれていません。

ユーザーが crontab コマンドを使用できない場合は、上記の crontab ファイルの有無に関わらず、次のメッセージが表示されます。

```
crontab: cron を使用許可されていません
```


このメッセージは、ユーザーが `cron.allow` (が存在する場合) に含まれていないか、`cron.deny` ファイルに含まれていることを意味します。

1つのシステムタスクのスケジューリング (at)

以降の節では、`at` コマンドを使用して特定の時刻にジョブ (コマンドとスクリプト) をスケジューリングする方法、これらのジョブを表示して削除する方法、`at` コマンドの使用を制御する方法について説明します。

デフォルトでは、ユーザーはそれぞれ自分の `at` ジョブファイルを作成、表示、または削除できます。 `root` または他のユーザーの `at` ファイルにアクセスするには、スーパーユーザーの権限が必要です。

`at` ジョブの実行を指定すると、`at` ジョブにジョブ識別番号と `.a` 拡張子が与えられ、それがジョブのファイル名になります。

at コマンドの説明

`at` ジョブの実行を指定するには、次の手順に従います。

1. コマンド実行時刻を指定して `at` ユーティリティを起動します。
2. 後で実行させるコマンドまたはスクリプトを入力します。

注 - このコマンドまたはスクリプトからの出力が重要な場合は、後で調べることができるように、出力内容を必ずファイルに書き込むようにしてください。

たとえば、次の `at` ジョブは、7月31日の真夜中に `smith` のユーザーアカウントから `core` ファイルを削除します。

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2003
```

at コマンドの使用制御

特定のユーザーだけがそれぞれの at ジョブに関する待ち行列情報を作成、削除、または表示できるように、at コマンドの使用を制御するファイルを設定できます。at コマンドの使用を制御するファイルは /etc/cron.d/at.deny です。ここにはユーザー名が列挙 (1 行に 1 人) されています。このファイルに列挙されているユーザーは、at コマンドを使用できません。

Solaris ソフトウェアのインストール時に作成される at.deny ファイルには、次のユーザー名が含まれます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

スーパーユーザーの特権があれば、at.deny ファイルを編集して、at の使用を制限したい他のユーザー名を追加できます。

▼ at ジョブを作成する方法

1. **at** ユーティリティを起動して、ジョブを実行したい時間を指定します。

```
$ at [-m] time [date]
```

<i>-m</i>	ジョブ終了後にメールを送る
<i>time</i>	ジョブをスケジュールしたい時刻の時単位の値。24 時間制を使用しない場合は、am または pm を追加する。使用できるキーワードは、midnight、noon、now。分単位の値の指定は省略可能
<i>date</i>	月または曜日の名前の最初の 3 英字以上、またはキーワード today または tomorrow

2. **at** プロンプトに、実行したいコマンドまたはスクリプトを 1 行に 1 つずつ入力します。
各行の終わりで Return キーを押すことにより、複数のコマンドを入力できます。
3. **at** ユーティリティを終了し、**Control-D** キーを押して **at** ジョブを保存します。
作成できた at ジョブは待ち行列番号を割り当てられ、それがそのジョブのファイル名にもなります。この番号は at ユーティリティの終了時に表示されます。

例 — at ジョブを作成する

次の例は、ユーザー `jones` がバックアップファイルを午後 7 時 30 分に削除するように作成した `at` ジョブを示しています。ジョブの終了後にメールメッセージを受け取れるように、`-m` オプションが使用されています。

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul 12 19:30:00 2003
```

`jones` は次のメールメッセージを受け取りました。このメッセージは `at` ジョブが終了したことを確認しています。

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

次の例は、`jones` が土曜の午前 4 時の大規模な `at` ジョブをどのようにスケジュールしたのかを示しています。ジョブの出力先は `big.file` です。

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

at 待ち行列を表示する方法

`at` 待ち行列で実行を待っているジョブを確認するには、次に示すように `atq` コマンドを使用します。このコマンドは、その使用者が作成した `at` ジョブに関する状態情報を表示します。

```
$ atq
```

at ジョブを確認する方法

`at` ジョブが作成できたかどうかを確認するには、`atq` コマンドを使用します。次の `atq` コマンドは、`jones` に属する `at` ジョブが待ち行列に入っていることを確認しています。

```
$ atq
Rank      Execution Date      Owner      Job          Queue  Job Name
  1st     Jul 12, 2003 19:30   jones     897355800.a    a      stdin
  2nd     Jul 14, 2003 23:45   jones     897543900.a    a      stdin
  3rd     Jul 17, 2003 04:00   jones     897732000.a    a      stdin
```

at ジョブを表示する方法

自分の `at` ジョブの実行時刻に関する情報を表示するには、次のように `at -l` コマンドを使用します。

```
$ at -l [job-id]
```

-l *job-id* オプションで、状態を表示したいジョブの識別番号を指定します。

例 — at ジョブを表示する

次の例は、at -l コマンドからの出力を示しています。このコマンドは、特定のユーザーが依頼したすべてのジョブに関する状態を表示します。

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
```

次の例は、at -l コマンドに1つのジョブを指定して表示された出力を示しています。

```
$ at -l 897732000.a
897732000.a    Tue Jul 17 04:00:00 2003
```

▼ at ジョブを削除する方法

1. (省略可能) **root** または他のユーザーの **at** ジョブを削除する場合は、スーパーユーザーになります。

2. 次のように入力して、**at** ジョブが実行される前に待ち行列から削除します。

```
$ at -r [job-id]
-r job-id オプションで、削除したいジョブの識別番号を指定します。
```

3. **at -l** (または **atq**) コマンドを使用して、**at** ジョブが削除されていることを確認します。

at -l コマンドは、at 待ち行列に残っているジョブを表示します。識別番号を指定したジョブは、このリストに表示されないはずですが。

```
$ at -l [job-id]
```

例 — at ジョブを削除する

次の例では、ユーザーが7月17日の午前4時に実行されるようにスケジュールしたat ジョブを削除しようとしています。まず、このユーザーはat 待ち行列を表示してそのジョブの識別番号を探します。次に、そのジョブをat 待ち行列から削除します。最後に、at 待ち行列をもう一度表示して上記のジョブが削除されていることを確認します。

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
```

```
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

▼ at コマンドの使用を拒否する方法

1. スーパーユーザーになります。
2. `/etc/cron.d/at.deny` ファイルを開きます。at コマンドを使用させないようにするユーザー名を1行に1つずつ追加または削除します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

例 — at の使用を拒否する

次は、ユーザー smith と jones が at コマンドを使用できないように編集された `at.deny` ファイルの例です。

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

at の使用の拒否を確認する方法

特定のユーザー名が正しく `/etc/cron.d/at.deny` に追加されているかどうかを確認するには、そのユーザー名でログインして、`at -l` コマンドを使用します。そのユーザーが at コマンドを使用できない場合は、次のメッセージが表示されます。

```
# su smith
Password:
$ at -l
```

```
at: you are not authorized to use at. Sorry.
```

同様に、そのユーザーが at ジョブの実行を依頼しようとした場合は、次のメッセージが表示されます。

```
$ at 2:30pm
```

```
at: you are not authorized to use at. Sorry.
```

このメッセージによって、そのユーザーが at.deny ファイルに含まれていることが確認されます。

at コマンドを使用できる場合、at -l コマンドは何も返しません。

第 19 章

システムアカウンティングの設定と管理 (手順)

この章では、システムアカウンティングを設定して維持する方法について説明します。

この章で説明する手順は次のとおりです。

- 293 ページの「システムアカウンティングを設定する方法」
- 295 ページの「ユーザーに課金する方法」
- 296 ページの「壊れた wtmpx ファイルの修復方法」
- 297 ページの「tacct エラーを修復する方法」
- 298 ページの「runacct スクリプトを再起動する方法」
- 293 ページの「システムアカウンティングを設定する方法」
- 299 ページの「システムアカウンティングを永続的に無効にする方法」

この章の内容は以下のとおりです。

- 287 ページの「システムアカウンティング」
- 292 ページの「システムアカウンティングの設定」

拡張アカウンティング機能の使い方については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「拡張アカウンティング」を参照してください。

システムアカウントのさまざまなレポートについては、第 20 章を参照してください。

システムアカウンティング

SunOS 5.9 のシステムアカウンティングソフトウェアは、ユーザー接続時間、プロセスに使用された CPU 時間、およびディスク使用量についてのデータを収集および記録できるプログラム群です。一度このデータを収集すると、レポートを生成したり、システム使用に対して課金したりすることができます。

アカウントリングプログラムを使用すると、次の事項を行うことができます。

- システムの使用状況の監視
- パフォーマンス上の問題の追跡と解決
- システムセキュリティの維持

システムアカウントリングプログラムは、設定が済むと、ほとんどの場合自動的に実行されます。

システムアカウントリングの動作

自動アカウントリングは、まずアカウントリング起動スクリプトをルートの `crontab` ファイルに配置することによって設定します。すると、アカウントリング起動スクリプトが、`cron` コマンドによって自動的に起動されます。

次の概要は、システムアカウントリングのプロセスを示したものです。

1. システムを起動してからシャットダウンするまでの間に、システムの利用に関する (ユーザーログイン、実行されたプロセス、データの格納などの) raw データがアカウントリングファイルに収集されます。
2. 定期的に (通常 1 日に 1 回)、`/usr/lib/acct/runacct` スクリプトが各種のアカウントリングファイルを処理して、累積要約ファイルと日次アカウントリングレポートを生成します。次に、`/usr/lib/acct/prdaily` スクリプトが日次レポートを印刷します。
`runacct` スクリプトについては、301 ページの「`runacct` スクリプト」を参照してください。
3. 毎月、`monacct` スクリプトを実行することによって、`runacct` 累積要約ファイルを処理して印刷します。`monacct` スクリプトによって生成される要約レポートは、月次またはその他の会計期間ベースのユーザーに対する効率的な課金手段になります。

システムアカウントリングのコンポーネント

アカウントリングユーティリティは、データから要約ファイルとレポートを生成する C 言語プログラムとシェルスクリプトを提供します。これらのプログラムは `/usr/lib/acct` ディレクトリにあります。アカウントリングレポートは、`/var/adm/acct` ディレクトリにあります。

日次アカウントリングによって、次の 4 種類の監査を行うことができます。

- 接続
- プロセス
- ディスク
- 料金計算

接続アカウントティング

接続アカウントティングでは、次のデータを調べることができます。

- 特定のユーザーがログインしていた時間
- tty 回線の利用状況
- システムのリポート回数
- アカウントティングソフトウェアが有効または無効に設定された頻度

この情報を提供するために、システムは次のデータを格納します。

- 時間調節の記録
- ブート時刻
- アカウントティングソフトウェアが有効または無効に設定された時刻
- 実行レベルの変更
- ユーザープロセスの作成 (login プロセスと init プロセス)、およびプロセスの終了

これらのレコード (date、init、login、ttypmon、acctwtmp などのシステムプログラム出力によって生成されます) は /var/adm/wtmpx ファイルに格納されます。wtmpx ファイルのエントリには、次の情報を入れることができます。

- ログイン名
- 装置名
- プロセス ID
- エントリタイプ
- エントリがいつ作成されたのかを示すタイムスタンプ

プロセスアカウントティング

プロセスアカウントティングでは、システムで実行される各プロセスに関する次のようなデータを追跡できます。

- プロセスを使用するユーザーおよびグループのそれぞれのユーザー ID とグループ ID
- プロセスの開始時刻と経過時間
- プロセスの CPU 時間 (ユーザー時間とシステム時間)
- 使用されるメモリーの量
- 実行されるコマンド
- プロセスを制御する tty

プロセスが終了するたびに、exit プログラムは上記のデータを収集し、/var/adm/pacct ファイルに書き込みます。

ディスクアカウントティング

ディスクアカウントティングでは、各ユーザーがディスク上にもっているファイルについて次のデータを収集し、それらを整形できます。

- ユーザーのユーザー名とユーザー ID

■ ユーザーのファイルが使用しているブロック数

これらのデータはシェルスクリプト `/usr/lib/acct/dodisk` によって収集されますが、収集周期は `/var/spool/cron/crontabs/root` ファイルに追加する cron コマンドによって決定されます。一方、`dodisk` スクリプトは、`acctdisk` および `acctdusg` コマンドを起動して、ログインによるディスク使用量を収集させます。



注意 - `dodisk` スクリプトを実行して収集された情報は、`/var/adm/acct/nite/disktacct` ファイルに格納されます。これらの情報は、次に `dodisk` スクリプトを実行したときに上書きされます。したがって、`dodisk` スクリプトは同じ日に 2 回以上実行しないでください。

`acctdusg` コマンドは、ランダムに書き込まれたため穴があいたファイルに対して過剰に課金します。このような問題が起こるのは、`acctdusg` コマンドが、ファイルサイズを決めるときに、ファイルの間接ブロックを読み取らないからです。`acctdusg` コマンドは、i ノードの `di_size` の値をチェックして、ファイルのサイズを決めます。

ユーザー料金の計算

`chargefee` ユーティリティは、ユーザーに提供した特別なサービス (ファイルの復元など) に対する課金を `/var/adm/fee` ファイルに格納します。このファイルの各エントリは、ユーザーのログイン名、ユーザー ID、および料金から構成されています。このファイルは、`runacct` スクリプトによって毎日チェックされて、新しいエントリが全体のアカウントングレコードにマージされます。`chargefee` スクリプトを実行してユーザーに課金する命令については、295 ページの「ユーザーに課金する方法」を参照してください。

日次アカウントング機能の動作

次に、SunOS の日次アカウントング機能がどのように動作するかを要約して示します。

1. システムをマルチユーザーモードに切り替えると、`/usr/lib/acct/startup` プログラムが実行されます。この `startup` プログラムは、それぞれ日次アカウントング機能を呼び出す他のプログラムを実行します。
2. `acctwtmp` プログラムは `/var/adm/wtmpx` ファイルに「ブート」レコードを追加します。このレコードには、システム名が `wtmpx` レコード内のユーザー名として示されます。次の表に、`raw` アカウントングデータがどのように収集され、どこに格納されるかをまとめて示します。

表 19-1 raw アカウンティングデータ

<code>/var/adm</code> 内のファイル	格納される情報	データを書き込むプログラム	表記形式
wtmpx	接続セッション数	login、init	バイナリ
	日付変更	date	
	リポート	acctwtmp	
	シャットダウン	shutacct	
pacctn	プロセス	カーネル(プロセス終了時)	バイナリ
		turnacct switch (古いファイルの内容が 500 ブロックに達すると、新しいファイルが作成される)	
fee	特別料金	chargefee	ASCII
acct/nite/diskacct	使用ディスク領域	dodisk	バイナリ

- turnacct スクリプトが `-on` オプションで起動されて、プロセスアカウンティングを開始します。turnacct スクリプトは、特に `/var/adm/pacct` 引数を使用して `accton` プログラムを実行します。
- remove シェルスクリプトが、runacct によって `sum` ディレクトリに保存されている `pacct` および `wtmpx` ファイルを「整理」します。
- login および init プログラムが、`/var/adm/wtmpx` ファイルにレコードを書き込み、接続セッションを記録します。すべての日付変更(引数を指定して `date` を使用)も `/var/adm/wtmpx` ファイルに書き込まれます。リポート回数とシャットダウン回数も、`acctwtmp` コマンドを使用して、`/var/adm/wtmpx` ファイルに記録されます。
- プロセスが終了すると、カーネルが `/var/adm/pacct` ファイルにプロセスごとに 1 レコードを `acct.h` 形式で書き込みます。
cron コマンドは、1 時間ごとに `ckpacct` スクリプトを実行して、`/var/adm/pacct` ファイルのサイズをチェックします。このファイルが 500 ブロック(デフォルト)よりも大きくなった場合は、`turnacct switch` コマンドが実行されます。(このプログラムは `pacct` ファイルを `pacctn` ファイルに移動して、新しいファイルを作成します)。pacct ファイルを小さく分けることの利点は、それらのレコードを処理するときに障害が発生して、runacct スクリプトを再起動しようとしたときに明らかになります。
- runacct スクリプトは、cron コマンドによって毎晩実行されます。runacct スクリプトは、`/var/adm/pacctn`、`/var/adm/wtmpx`、`/var/adm/fee` および `/var/adm/acct/nite/diskacct` のアカウンティングファイルを処理し、ユーザー別のコマンド要約と利用状況要約を生成します。

8. /usr/lib/acct/prdaily スクリプトは、runacct スクリプトによって1日1回実行され、/var/adm/acct/sum/rprt MMDD ファイルに日次アカウントリング情報を書き込みます。
9. monacct スクリプトが月に1回(または毎会計期の終わりなど、ユーザーが決めた周期で)実行されます。monacct スクリプトは、sum ディレクトリに格納されているデータに基づいてレポートを作成します。これらのデータは runacct スクリプトによって毎日更新されています。このレポートを作成後、monacct スクリプトは sum ディレクトリを「整理」して、新しい runacct データを格納するためのファイルを準備します。

システムがシャットダウンしたときの動作

shutdown コマンドを使用してシステムをシャットダウンした場合は、shutacct スクリプトが自動的に実行されます。shutacct スクリプトは /var/adm/wtmpx ファイルに理由レコードを書き、アカウントリングプロセスを無効に設定します。

システムアカウントリングの設定

システムアカウントリングは、システムがマルチユーザーモード (システム状態 2) のときに実行されるように設定できます。システムアカウントリングには、次の内容が含まれます。

1. /etc/rc0.d/K22acct および /etc/rc2.d/S22acct 起動スクリプトの作成
2. crontab ファイル /var/spool/cron/crontabs/adm および /var/spool/cron/crontabs/root の変更

表 19-2 に、デフォルトのアカウントリング管理スクリプトを説明します。

表 19-2 デフォルトのアカウントリング管理スクリプト

アカウントリングスクリプト	マニュアルページ	種類	実行頻度
ckpacct	ckpacct (1M)	/usr/adm/pacct ログファイルのサイズをチェックし、ファイルが大きくなりすぎていないことを確認する	定期的

表 19-2 デフォルトのアカウント管理スクリプト (続き)

アカウント管理スクリプト	マニュアルページ	種類	実行頻度
runacct	runacct (1M)	接続、ディスク、および料金のアカウント管理情報を処理する。このスクリプトから、処理したくないアカウント管理機能のコマンドを削除できる	日次
monacct	monacct (1M)	会計アカウント管理要約レポートを月に1回のペースで生成する。このスクリプトの実行頻度を定めることができる。このスクリプトから、生成したくないアカウント管理機能のコマンドを削除できる	会計期間に基づく

デフォルトでどのアカウント管理スクリプトを実行するかを選択できます。これらのエントリを `crontab` ファイルに追加すると、アカウント管理は自動的に実行されるようになります。

▼ システムアカウント管理を設定する方法

1. スーパーユーザーになります。
2. 必要な場合は、`pkgadd` コマンドを使用して、システムに `SUNWacct` と `SUNWaccu` パッケージをインストールします。
3. `/etc/init.d/acct` を実行レベル 2 の起動スクリプトとしてインストールします。

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
```

4. `/etc/init.d/acct` を実行レベル 0 の停止スクリプトとしてインストールします。

```
# ln /etc/init.d/acct /etc/rc0.d/K22acct
```

5. `ckpacct`、`runacct`、および `monacct` スクリプトが自動的に起動するように、`adm` ユーザーの `crontab` ファイルに次の行を追加します。

```
# EDITOR=vi; export EDITOR
# crontab -e adm
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

6. **dodisk** スクリプトが自動的に起動するように、**root** の **crontab** ファイルに次の行を追加します。

```
# crontab -e
30 22 * * 4 /usr/lib/acct/dodisk
```

7. **/etc/acct/holidays** を編集して祭日と休日を含めます。
詳細は、**holidays(4)** のマニュアルページおよび次の例を参照してください。

8. システムをリブートするか、手入力でアカウントिंगを起動します。

```
# /etc/init.d/acct start
```

例 — アカウンティングを設定する

この編集された **adm** の **crontab** には、**ckpacct**、**runacct** および **monacct** スクリプトが含まれています。

```
#ident "@(#)adm          1.5      92/07/14 SMI"    /* SVr4.0 1.2  */
#
# The adm crontab file should contain startup of performance
# collection if the profiling and performance feature has been
# installed.
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * * /usr/lib/acct/monacct
```

この編集された **root** の **crontab** には、**dodisk** プログラムのエントリが含まれています。

```
#ident "@(#)root          1.19     98/07/06 SMI"    /* SVr4.0 1.1.3.1  */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 22 * * 4 /usr/lib/acct/dodisk
```

次に **/etc/acct/holidays** ファイルの例を示します。

```
* @(#)holidays      January 1, 2002
*
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr      Prime      Non-Prime
* Year      Start      Start
*
2002      0800      1800
```

```

*
* only the first column (month/day) is significant.
*
* month/day    Company
*             Holiday
*
1/1           New Years Day
7/4           Indep. Day
12/25        Christmas

```

ユーザーへの課金

ファイルの復元やリモート印刷など、ユーザーに特別なサービスを提供する場合は、`chargefee` ユーティリティを実行して、ユーザーに課金したいこともあります。`chargefee` ユーティリティは、`/var/adm/fee` ファイルに料金を記録します。次に、`runacct` ユーティリティが実行されるたびに、新しいエントリが拾い出されて、全体のアカウントングレコードにマージされます。

詳細は、`acctsh(1M)` のマニュアルページを参照してください。

▼ ユーザーに課金する方法

1. スーパーユーザーになります。
2. ユーザーに提供されたサービスに対して課金します。

```
# /usr/lib/acct/chargefee username amount
```

username 課金したいユーザーアカウント

amount ユーザーに対する課金の単位数。これは、ファイルの印刷や復元のような作業に基づいて、ユーザーに課金するために設定する任意の単位。`chargefee` を実行し、特定の作業に対してユーザーに課金するスクリプトを書く必要がある

例 — ユーザーに課金する

次の例は、ユーザーがアカウント `print_customer` にログインするたびに 10 単位を課金します。

```
# /usr/lib/acct/chargefee print_customer 10
```

アカウント情報の管理

この節では、壊れたアカウントファイルを修復し、runacct スクリプトを再起動する方法を説明します。

壊れたファイルと wtmpx エラーを修復する

アカウントシステムは絶対に障害を発生しないというものではありません。ファイルが壊れたり、失われることがあります。ファイルによっては、単に無視してよいものや、バックアップから復元できるものがあります。ただし、特定のファイルは、アカウントシステムの完全性を維持するために修復しなければなりません。

wtmpx ファイルは、アカウントシステムを日常的に運用する上で発生する問題の大部分の原因になっています。日付を手動で変更した時に、システムがマルチユーザーモードの場合は、日付変更レコードが/var/adm/wtmpx ファイルに書き込まれます。wtmpfix ユーティリティは、日付変更が行われたときに、wtmp レコードのタイムスタンプを調整するように設計されています。ただし、日付変更とレポートの組み合わせによっては、wtmpfix の処理から漏れて、acctcon プログラムの処理が失敗することがあります。

▼ 壊れた wtmpx ファイルの修復方法

1. スーパーユーザーになります。
2. /var/adm ディレクトリに移動します。
3. wtmpx ファイルの形式をバイナリから ASCII に変換します。

```
# /usr/lib/acct/fwtmp < wtmpx > wtmpx.ascii
```

4. wtmpx.ascii を編集して、壊れたレコードを削除します。
5. wtmpx.ascii ファイルを再びバイナリファイルに変換します。

```
# /usr/lib/acct/fwtmp -ic < wtmpx.ascii > wtmpx
```

詳細は、fwtmp(1M) のマニュアルページを参照してください。

tacct エラーを修復する

/var/adm/acct/sum/tacct ファイルの完全性は、システム資源に対してユーザーに課金している場合は重要です。不正な tacct レコードが、負の数値、重複ユーザー ID、65535 のユーザー ID で表示されることがあります。このような場合はま

ず、prtacct スクリプトを使用して、/var/adm/acct/sum/tacctprev ファイルを印刷してチェックします。内容が正しい場合は、最新の/var/adm/acct/sum/tacctMMDD ファイルを使用して/var/adm/acct/sum/tacct ファイルを作成し直してください。次の手順は、簡単な修復手順の概要を説明しています。

▼ tacct エラーを修復する方法

1. スーパーユーザーになります。
2. ディレクトリ /var/adm/acct/sum に移動します。
3. tacctMMDD ファイルの形式をバイナリから **ASCII** に変換します。

```
# /usr/lib/acct/acctmerg -v < tacctMMDD > xtacct
```

MMDD は、月と日をあらわす 2 桁の数値を組み合わせたものです。
4. xtacct ファイルを編集して、壊れたレコードを削除し、重複レコードを別のファイルに書き込みます。
5. xtacct ファイルを **ASCII** 形式からバイナリに変換します。

```
# /usr/lib/acct/acctmerg -i < xtacct > tacctMMDD
```

MMDD は、月と日をあらわす 2 桁の数値を組み合わせたものです。
6. ファイル tacctprv と tacct.MMDD をマージしてファイル tacct を生成します。

```
# /usr/lib/acct/acctmerg < tacctprev tacctMMDD > tacct
```

runacct スクリプトを再起動する

runacct スクリプトは、さまざまな理由で失敗することがあります。最も一般的な理由には、システムクラッシュ、/var ディレクトリの容量不足での実行、wtmptx ファイルが壊れたなどがあります。activeMMDD ファイルが存在する場合は、まずエラーメッセージをチェックします。active ファイルおよび lock ファイルが存在する場合は、異常なメッセージがないかどうか fd2log ファイルを調べます。

runacct スクリプトは、引数を指定しないで実行すると、その実行がその日の最初の実行であるとみなします。runacct スクリプトを再起動し、runacct スクリプトがアカウントを返す月と日を指定する場合は、引数 MMDD が必要になります。処理のエントリーポイントは statefile ファイルの内容に基づきます。statefile ファイルを無効にするには、処理を開始したい状態をコマンド行に指定します。利用可能な状態の説明については、runacct (1M) のマニュアルページを参照してください。



注意 - `runacct` プログラムを手動で実行するときは、ユーザー `adm` として実行していることを確認してください。

▼ `runacct` スクリプトを再起動する方法

1. `lastdate` ファイルと `lock*` ファイル (もしあれば) を削除します。

```
$ cd /var/adm/acct/nite  
$ rm lastdate lock*
```

`lastdate` ファイルには、`runacct` プログラムが最後に実行された日付が含まれています。次の手順で `runacct` を再起動することによって、このファイルが再び作成されます。

2. `runacct` スクリプトを再起動します。

```
$ /usr/lib/acct/runacct MMDD [state] 2> /var/adm/acct/nite/fd2log &
```

`MMDD`

月日を2桁の数値で指定する

`state`

`runacct` 処理を開始させたい状態または開始点を指定する。

システムアカウンティングの停止と無効

システムアカウンティングは、一時的に停止することも、永続的に無効にすることもできます。

▼ 一時的にシステムアカウンティングを停止する方法

1. スーパーユーザーになります。
2. 適切な行をコメントアウトすることによって、プログラム `ckpacct`、`runacct`、および `monacct` の実行が停止するように、ユーザー `adm` の `crontab` ファイルを編集します。

```
# EDITOR=vi; export EDITOR  
# crontab -e adm  
#0 * * * * /usr/lib/acct/ckpacct
```

```
#30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
#30 7 1 * * * /usr/lib/acct/monacct
```

3. 適切な行をコメントアウトすることによって、プログラム **dodisk** の実行が停止するように、ユーザー **root** の **crontab** ファイルを編集します。

```
# crontab -e
#30 22 * * 4 /usr/lib/acct/dodisk
```

4. 次のように入力して、アカウントプログラムを停止します。

```
# /etc/init.d/acct stop
```

5. (省略可能)新規に追加したコメント記号を **crontab** ファイルから削除し、アカウントプログラムを再起動して、システムアカウントプログラムを再び利用できるようにします。

```
# /etc/init.d/acct start
```

▼ システムアカウントプログラムを永続的に無効にする方法

1. スーパーユーザーになります。
2. ユーザー **adm** の **crontab** ファイルを編集して、プログラム **ckpacct**、**runacct**、および **monacct** 用のエントリを削除します。

```
# EDITOR=vi; export EDITOR
# crontab -e adm
```

3. ユーザー **root** の **crontab** ファイルを編集して、プログラム **dodisk** 用のエントリを削除します。

```
# crontab -e
```

4. 実行レベル 2 用の起動スクリプトのリンクを取り外します。

```
# unlink /etc/rc2.d/S22acct
```

5. 実行レベル 0 用の停止スクリプトのリンクを取り外します。

```
# unlink /etc/rc0.d/K22acct
```

6. 次のように入力して、アカウントプログラムを停止します。

```
# /etc/init.d/acct stop
```


第 20 章

システムアカウントティング (リファレンス)

この章では、システムアカウントの参照情報について説明します。

この章の内容は、次のとおりです。

- 303 ページの「日次アカウントングレポート」
- 301 ページの「runacct スクリプト」
- 311 ページの「システムアカウントングファイル」

runacct スクリプト

主となる日次アカウントングスクリプトである runacct は、通常はプライムタイムつまり最も忙しい時間帯を避けて cron コマンドにより起動されます。この runacct スクリプトは、接続、料金、ディスク、プロセス用の各アカウントングファイル処理します。さらに、このスクリプトは、課金目的で prdaily と monacct スクリプトが使用する日次および累積要約ファイルも準備します。

runacct スクリプトは、エラーが発生した場合、ファイルを壊さないよう注意します。一連の保護機構により、エラーを認識し、インテリジェント診断を提供し、最小のユーザー介入で runacct スクリプトを起動し直せるように処理を完了しようとしています。runacct は、説明メッセージを active というファイルに書き込み、進捗状況を記録します。runacct スクリプトが使用するファイルは、特記しない限り、/var/adm/acct/nite ディレクトリにあります。runacct スクリプト実行中の診断内容の出力は、fd2log ファイルに書き込まれます。

runacct スクリプトが実行されると、lock および lock1 ファイルが作成されます。これらのファイルは、runacct が同時に実行されるのを防ぎます。runacct プログラムは、起動時にこれらのファイルが存在すれば、エラーメッセージを出力します。lastdate ファイルには runacct スクリプトが最後に起動されたときの月日が入っており、このファイルは runacct スクリプトを 1 日に 1 回しか実行できないようにするために使用されます。runacct スクリプトがエラーを検出した場合は、エ

ラメッセージがコンソールに出力され、メールが root と adm に送られ、ロックが解除され、診断ファイルが保存され、実行が終了されます。runacct スクリプトを再び起動する手順については、298 ページの「runacct スクリプトを再起動する方法」を参照してください。

runacct スクリプトを再起動可能にするために、処理は再入可能な別々の状態に分割されます。statefile というファイルを使用して、最後に終了した状態が追跡されます。各状態が終了するたびに、statefile は更新されて次の状態に反映されます。1つの状態の処理が終了すると、statefile ファイルが読み取られて次の状態が処理されます。runacct スクリプトは、CLEANUP 状態に達すると、ロックを解除して実行を終了します。各状態は次の表のように実行されます。

表 20-1 runacct 状態

状態	説明
SETUP	turnacct switch コマンドが実行されて新しい pacct ファイルが作成される。/var/adm/pacctn プロセスアカウンティングファイル (pacct ファイルを除く) は、/var/adm/Spacctn.MMDD ファイルに移される。/var/adm/wtmpx ファイルは /var/adm/acct/nite/wtmp.MMDD ファイルに移動され (現在の時間のレコードは終わりに追加される)、新しい /var/adm/wtmp ファイルが作成される。closewtmp および utmp2wtmp プログラムは、現在ログインしているユーザーに課金するために wtmp.MMDD ファイルと新しい wtmpx ファイルにレコードを追加する。
WTMPFIX	wtmpfix プログラムが nite ディレクトリ内の wtmp.MMDD ファイルを調べて誤りがないか確認する。日付の変更により、acctcon プログラムの失敗が引き起こされることがあるため、日付のレコードが変更されると、wtmpfix プログラムは wtmpx ファイルのタイムスタンプを調整しようとする。さらに、wtmpx ファイルからのエントリが壊れていた場合、壊れたエントリをすべて削除する。修復されたバージョンの wtmp.MMDD ファイルは tmpwtmp ファイルに書き込まれる。
CONNECT	acctcon プログラムが使用されて、ctacct.MMDD ファイルに接続アカウンティングレコードが記録される。これらのレコードは tacct.h 形式になっている。acctcon は、さらに lineuse および reboots ファイルを作成する。reboots ファイルは、wtmpx ファイルで見つかったすべてのブートレコードを記録する
PROCESS	acctprc プログラムが使用されて、/var/adm/Spacctn.MMDD プロセス課金ファイルが ptacctn.MMDD ファイルのアカウンティングレコードの合計に変換する。runacct スクリプトが失敗した場合に、Spacct ファイルが処理されないように、Spacct ファイルと ptacct ファイルは番号で照合される
MERGE	acctmerg プログラムが、プロセスアカウンティングレコードを接続アカウンティングレコードとマージして daytacct ファイルを作成する

表 20-1 runacct 状態 (続き)

状態	説明
FEES	acctmerg プログラムが、fee ファイルからの ASCII tacct レコードを daytacct ファイルにマージする
DISK	dodisk スクリプトの手続きが実行されて disktacct ファイルが作成されている場合は、そのファイルを DISK プログラムは daytacct ファイルにマージし、disktacct ファイルを /tmp/disktacct.MMDD ファイルに移動させる
MERGETACCT	acctmerg プログラムが、daytacct を累積合計課金ファイルである sum/tacct ファイルとマージする。daytacct ファイルは毎日 sum/tacct.MMDD ファイルに保存されるため、sum/tacct ファイルは壊れたり失われたりしても、作成し直すことができる
CMS	acctcms プログラムが数回実行される。まず、このプログラムが実行され、spacctn ファイルを使用してコマンド要約を生成し、sum/daycms ファイルに日付を書き込む。次に、acctcms プログラムは、sum/daycms ファイルと sum/cms 累積コマンド要約ファイルをマージする。最後に、acctcms プログラムは、nite/daycms および nite/cms という ASCII コマンド要約ファイルを、それぞれ sum/daycms および sum/cms ファイルから作成する。lastlogin プログラムを使用してログファイル /var/adm/acct/sum/loginlog が作成される。これは、各ユーザーが最後にログインした時刻を示すレポートである。runacct スクリプトが真夜中を過ぎてから実行された場合は、いずれかのユーザーが最後にログインした時刻を示す日付が 1 日分狂うことになる
USEREXIT	インストールに依存する任意の (ローカル) 課金プログラムをこの時点で取り入れることができる。runacct スクリプトは、そのプログラムを /usr/lib/acct/runacct.local プログラムと呼ばれるものと想定する
CLEANUP	一時ファイルを整理し、prdaily スクリプトを実行し、その出力を sum/rpt.MMDD ファイルに保存し、ロックを解除し、終了する



注意 - runacct スクリプトを CLEANUP 状態で起動し直すときは、最後の ptacct ファイルが不完全であるため、このファイルを削除してください。

日次アカウントングレポート

runacct シェルスクリプトは、呼び出されるたびに基本的な 5 種類のレポートを生成します。5 つの基本レポートは次のとおりです。

表 20-2 日次アカウンティングレポート

レポートの種類	説明
304 ページの「日次レポート」	tty 番号別の端末回線の利用状況を示す
305 ページの「日次利用状況レポート」	ユーザー別のシステム資源の利用状況を示す。ユーザー ID 順に表示される
306 ページの「日次コマンド要約レポート」	コマンド別のシステム資源の利用状況を示す。メモリー使用量が大きい順に表示される。つまり、メモリーを最も多く使用したコマンドから先に表示される。これと同じ情報が月次コマンド概要で該当月について報告される
308 ページの「月次コマンド要約」	monacct プログラムの最新の実行から累積した日付を反映した累積概要
308 ページの「最終ログインレポート」	各ユーザーが最後にログインした日付を示す。日付順に表示される

日次レポート

このレポートは、使用された各端末回線に関する情報を示します。次に日次レポートの例を示します。

```
Oct 16 02:30 2002 DAILY REPORT FOR venus Page 1
```

```
from Mon Oct 15 02:30:02 2002
to Tue Oct 16 02:30:01 2002
1 runacct
1 acctcon
```

```
TOTAL DURATION IS 1440 MINUTES
LINE      MINUTES  PERCENT  # SESS  # ON  # OFF
console   868      60       1      1    2
TOTALS    868      --       1      1    2
```

from および to の行は、レポートに反映される時間帯を指定します。この時間帯とは、直前の日次レポートが生成された時間から現在の日次レポートが生成されるまでの時間のことです。次に、システムリブートのログ、シャットダウン、電源異常からの回復、acctwtmp プログラムによって /var/adm/wtmpx ファイルにダンプされたその他のレコードが続きます。詳細は、acct(1M) のマニュアルページを参照してください。

このレポートの第 2 部は回線利用状況の内訳です。TOTAL DURATION は、システムがどれだけの時間マルチユーザーモード (端末回線を通してアクセス可能です) であったかを示します。次の表は、日次レポートのデータを説明したものです。

表 20-3 日次レポート

列	説明
LINE	端末回線またはアクセスポート
MINUTES	アカウント期間を通じてこの回線が使用中であった合計分
PERCENT	TOTAL DURATION を MINUTES の合計値で割った数値
# SESS	この回線またはポートがログインセッション用にアクセスされた回数
# ON	SESS に同じ。(このカラムにはそれ以上の意味はない。回線またはポートがユーザーのログインに使用された回数を表示する)
# OFF	このカラムは、この回線でユーザーがログアウトした回数と発生した割り込みを表す。割り込みは一般にシステムがマルチユーザーモードにされてから ttymon が初めて起動されたときに発生する。# OFF が大きな割合で # SESS を上回る場合は、マルチプレクサ、モデム、ケーブルに障害があるか、どこかに接続の問題がある可能性がある。一番考えられる原因は、マルチプレクサからのケーブルの接続が外れたままになっていることである

マシンの稼動中は、`/var/adm/wtmpx` ファイルが接続アカウントの元になるため、このファイルを監視する必要があります。wtmpx ファイルが急速に大きくなる場合は、次のコマンドを実行して、どの tty 回線が最も使用頻度が高いかを調べてください。

```
# /usr/lib/acct/acctcon -l file < /var/adm/wtmpx
```

割り込みが頻繁に発生する場合は、全般的なシステムのパフォーマンスが影響を受けることになります。さらに、wtmp ファイルが壊れることもあります。この問題を解決するには、296 ページの「壊れた wtmpx ファイルの修復方法」を参照してください。

日次利用状況レポート

このレポートは、システム資源の利用状況のユーザー別の内訳を示します。このレポートの例は、次のとおりです。

```
Oct 16 02:30 2002 DAILY USAGE REPORT FOR skisun Page 1
```

UID	LOGIN NAME	CPU PRIME	(MINS) NPRIME	KCORE- PRIME	MINS NPRIME	CONNECT PRIME	(MINS) NPRIME	DISK BLOCKS	# OF PROCS	# OF SESS	# OF DISK SAMPLES	FEE
0	TOTAL	72	148	11006173	51168	26230634	57792	539	330	0	2150	1
0	root	32	76	11006164	33664	26230616	22784	0	0	0	127	0
4	adm	0	0	22	51	0	0	0	420	0	0	0
101	rimmer	39	72	894385	1766020	539	330	0	1603	1	0	0

次の表は、日次利用状況レポートのデータを説明したものです。

表 20-4 日次利用状況レポート

列	説明
UID	ユーザー ID 番号
LOGIN NAME	ユーザーのログイン (またはユーザー) 名。複数のログイン名をもつユーザーを識別する
CPU (MINS)	ユーザーのプロセスが CPU を使用した時間を表す。このカテゴリの情報は、PRIME (プライムタイム時間帯) と NPRIME (プライムタイム時間帯外) に分けられる。アカウントシステムこれらのデータのバージョンは、 <code>/etc/acct/holidays</code> ファイルに格納されている
KCORE-MINS	プロセスが実行中に使用する累積メモリー量を表す。毎分あたりに使用される K バイトメモリーセグメント数を表す。この計量値も PRIME と NPRIME に分けられる
CONNECT (MINS)	ユーザーがシステムにログインしていた時間を分単位で表す。「実時間」とも呼ぶ。PRIME と NPRIME の利用状況に分けられる。たとえば、この時間の値が大きく # OF PROCS の数値が小さい場合は、ログインの所有者がまず朝にログインし、その後はその日の終わりまで端末にほとんど触れていないと考えられる
DISK BLOCKS	<code>acctdusg</code> プログラムからの出力であり、ディスクアカウントングプログラムを実行し、アカウントングレコード (<code>daytacct</code>) をマージする。アカウントングの目的では、ブロックは 512 バイト
# OF PROCS	ユーザーが起動したプロセス数を表す。数値が大きい場合は、ユーザーのシェルプロセスが制御できなくなった可能性がある
# OF SESS	ユーザーがシステムにログインした回数
# DISK SAMPLES	平均ディスクブロック数 (DISK BLOCKS) を得るためにディスクアカウントングが何回実行されたかを示す
FEE	<code>chargefee</code> スクリプトによってユーザーに課金された累積合計額を表す。使用されない場合が多い

日次コマンド要約レポート

このレポートはコマンド別のシステム資源の利用状況を示します。このレポートでは、最も使用率の高いコマンドがわかり、それらコマンドがどのようにシステム資源を利用しているかに基づいて、どのようにしたらシステムの最適チューニングが可能かを知ることができます。

これらのレポートは TOTAL KCOREMIN によってソートされます。TOTAL KCOREMIN は任意の基準ですが、システムでのドレーンの計算にはすぐれた指標です。

日次コマンド要約の例を、次に示します。

TOTAL COMMAND SUMMARY									
COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	2150	1334999.75	219.59	724258.50	6079.48	0.10	0.00	397338982	419448
netscape	43	2456898.50	92.03	54503.12	26695.51	2.14	0.00	947774912	225568
adeptedi	7	88328.22	4.03	404.12	21914.95	0.58	0.01	93155160	8774
dtmail	1	54919.17	5.33	17716.57	10308.94	5.33	0.00	213843968	40192
acroread	8	31218.02	2.67	17744.57	11682.66	0.33	0.00	331454464	11260
dtwm	1	16252.93	2.53	17716.57	6416.05	2.53	0.00	158662656	12848
dtterm	5	4762.71	1.30	76300.29	3658.93	0.26	0.00	33828352	11604
dtaction	23	1389.72	0.33	0.60	4196.43	0.01	0.55	18653184	539
dtssessio	1	1174.87	0.24	17716.57	4932.97	0.24	0.00	23535616	5421
dtcm	1	866.30	0.18	17716.57	4826.21	0.18	0.00	3012096	6490

次の表は、日次コマンド要約のデータを説明したものです。

表 20-5 日次コマンド要約レポート

列	説明
COMMAND NAME	コマンド名。プロセスアカウンティングシステムではオブジェクトモジュールしか報告されないで、シェルプロセスはすべて sh という名前でも取り扱われる。a.out または core と呼ばれるプログラム、またはその他の、適切とは思われない名前のプログラムの使用頻度を監視すると良い。acctcom プログラムを使用して、名前に疑問があるコマンドを誰が実行したか、スーパーユーザー特権が使用されたかどうかを知ることができる
NUMBER CMDS	プライムタイム時間帯に、このコマンドが呼び出された回数
TOTAL KCOREMIN	実行時の毎分あたりにプロセスが使用した K バイトメモリーセグメント数という計量値の累積合計
TOTAL CPU-MIN	このプログラムのプライムタイム時間帯の累積合計処理時間
TOTAL REAL-MIN	このプログラムのプライムタイム時間帯の累積合計実時間 (壁掛け時計)。分単位
MEAN SIZE-K	NUMBER CMDS で表される呼び出し回数に対する TOTAL KCOREMIN の平均
MEAN CPU-MIN	NUMBER CMDS に対する TOTAL CPU-MIN の平均
HOG FACTOR	合計 CPU 時間を経過時間で割った値。システム利用可能時間とシステム使用時間との比であり、プロセスがその実行中に消費する合計利用可能 CPU 時間の相対値を示す
CHARS TRNSFD	読み取りおよび書き込みシステムコールによってプッシュされた文字の合計数。オーバーフローのために負の値になることがある

表 20-5 日次コマンド要約レポート (続き)

列	説明
BLOCKS READ	プロセスが実行した物理ブロックの読み取りおよび書き込みの合計数

月次コマンド要約

日次コマンド要約と月次コマンド要約のレポート形式は、実際は同じものです。ただし、日次コマンド要約は現在のアカウント期間だけでレポートするのに対し、月次コマンド要約は会計期間の当初から現在の日付までをレポートします。つまり、月次レポートは、monacct プログラムが最後に実行されたときからの累積データの累積要約を表します。

レポートの例を次に示します。

Oct 16 02:30 2002 MONTHLY TOTAL COMMAND SUMMARY Page 1

TOTAL COMMAND SUMMARY									
COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	42718	4398793.50	361.92	956039.00	12154.09	0.01	0.00	16100942848	825171
netscape	789	3110437.25	121.03	79101.12	25699.58	0.15	0.00	3930527232	302486
adeptedi	84	1214419.00	50.20	4174.65	24193.62	0.60	0.01	890216640	107237
acroread	145	165297.78	7.01	18180.74	23566.84	0.05	0.00	1900504064	26053
dtmail	2	64208.90	6.35	20557.14	10112.43	3.17	0.00	250445824	43280
dtaction	800	47602.28	11.26	15.37	4226.93	0.01	0.73	640057536	8095
soffice.	13	35506.79	0.97	9.23	36510.84	0.07	0.11	134754320	5712
dtwm	2	20350.98	3.17	20557.14	6419.87	1.59	0.00	190636032	14049

月次コマンド要約で提供されるデータの説明については、[306 ページの「日次コマンド要約レポート」](#)を参照してください。

最終ログインレポート

このレポートは、特定のログインが最後に使用された日付を示します。この情報を使用して、使用されていないログインやログインディレクトリを見つけることができます。それらのログインやログインディレクトリは保存して削除できます。次に例を示します。

Oct 16 02:30 2002 LAST LOGIN Page 1

01-06-12	kryten	01-09-08	protoA	01-10-14	ripley
01-07-14	lister	01-09-08	protoB	01-10-15	scutter1
01-08-16	pmorph	01-10-12	rimmer	01-10-16	scutter2

acctcom による pacct ファイルの確認

/var/adm/pacctn ファイル、または acct.h 形式の任意のファイルの内容は、acctcom プログラムを使用していつでも調べることができます。このコマンドを実行するときに、ファイルも標準入力も指定しなければ、acctcom コマンドは pacct ファイルを読み取ります。acctcom コマンドで読み取られる各レコードは、終了したプロセスの情報を表します。アクティブなプロセスは、ps コマンドを実行して調べます。acctcom コマンドのデフォルト出力は次に示す情報を示します。

acctcom 出力ファイルを、次に示します。

```
# acctcom
COMMAND
NAME      USER      TTYNAME      START      END          REAL        CPU        MEAN
          TIME      TIME          (SECS)     (SECS)     SIZE(K)
#accton   root      ?            02:30:01  02:30:01    0.03        0.01      304.00
turnacct  adm       ?            02:30:01  02:30:01    0.42        0.01      320.00
mv        adm       ?            02:30:01  02:30:01    0.07        0.01      504.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.03        0.01      712.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.01        0.01      824.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.01        0.01      912.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.01        0.01      920.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.01        0.01     1136.00
utmp_upd  adm       ?            02:30:01  02:30:01    0.01        0.01      576.00
closewtm  adm       ?            02:30:01  02:30:01    0.10        0.01      664.00
```

- コマンド名 (# 記号はスーパーユーザー特権を使用して実行された場合)
- ユーザー名
- tty 名 (不明の場合は ?)
- コマンド開始時刻
- コマンド終了時刻
- 実時間 (秒単位)
- CPU 時間 (秒単位)
- 平均サイズ (K バイト単位)

acctcom オプションを使用すると、次の情報を得ることができます。

- fork/exec フラグの状態 (exec を使用しない fork の場合は 1)
- システム終了状態
- hog 係数
- 合計 kcore 分
- CPU 係数
- 転送文字数
- 読み取りブロック数

表 20-6 で acctcom のオプションを説明します。

表 20-6 acctcom のオプション

オプション	説明
-a	選択したプロセスに関する特定の平均統計を表示する。統計は出力が記録された後に表示される
-b	ファイルを逆読みし、最後のコマンドから先に表示する。標準入力の読み込みには関係しない
-f	fork/exec フラグおよびシステム終了状態カラムを出力する。出力は 8 進数
-h	平均メモリーサイズの代わりに hog 係数を表示する。これは経過時間とプロセスが実行中に消費した合計 CPU 利用可能時間との比。hog 係数 = 合計 CPU 利用時間/経過時間
-i	入出力カウントを含むカラムを出力する
-k	メモリーサイズの代わりに、キロバイト/分ごとのコアサイズの合計値を表示する
-m	平均コアサイズを表示する。これがデフォルト
-q	平均統計だけを出力する。出力レコードは出力しない
-r	CPU 係数 (システム使用時間 / (システム使用時間 + ユーザー使用時間)) を表示する
-t	システムおよびユーザー CPU 時間を表示する
-v	出力からカラム見出しを除外する
-C sec	合計 (システム + ユーザー) CPU 時間が sec 秒を超えたプロセスだけを表示する
-e time	time 以前に存在したプロセスを hr[:min[:sec]] の書式で表示する
-E time	time 以前に開始されたプロセスを hr[:min[:sec]] の書式で表示する。同じ time を -s と -E の両方に使用すれば、そのときに存在していたプロセスを表示する
-g group	group に属しているプロセスだけを表示する
-H factor	factor を超えるプロセスだけを表示する。ただし、factor は「hog 係数」(-h オプションを参照)
-I chars	chars によって指定されるカットオフ数を超える文字数を転送したプロセスだけを表示する
-l line	端末 /dev/line に属しているプロセスだけを表示する
-n pattern	pattern 「+」が 1 回以上現れることを意味する以外は、一般的な正規表現に一致するコマンドだけを表示する
-o ofile	レコードを出力しないで、レコードを acct.h 形式で ofile にコピーする

表 20-6 acctcom のオプション (続き)

オプション	説明
-O <i>sec</i>	CPU システム時間が <i>sec</i> 秒を超えるプロセスだけを表示する
-s <i>time</i>	<i>time</i> 以後に存在したプロセスを <i>hr[:min[: sec]]</i> の書式で表示する
-S <i>time</i>	<i>time</i> 以後に開始されたプロセスを <i>hr[:min[: sec]]</i> の書式で表示する
-u <i>user</i>	<i>user</i> に属しているプロセスだけを表示する

システムアカウントングファイル

/var/adm ディレクトリには、使用中のデータ収集ファイルが含まれています。次の表は、このディレクトリにあるアカウントングファイルを説明しています。

表 20-7 /var/adm ディレクトリ内のファイル

ファイル	説明
dtmp	acctdusg プログラムからの出力
fee	chargefee プログラムからの出力である ASCII の tacct レコード
pacct	現在使用中のプロセスアカウントングファイル
pacct <i>n</i>	turnacct スクリプトの実行によって切り替えられたプロセスアカウントングファイル
Spacct <i>n</i> .MMDD	runacct スクリプトの実行中に生成された MMDD 日付のプロセスアカウントングファイル

/var/adm/acct ディレクトリには、nite、sum、fiscal の各ディレクトリが設けられ、それぞれに実際のデータ収集ファイルが格納されます。たとえば、nite ディレクトリは runacct スクリプトが毎日繰り返して使用するファイルを格納しています。表 20-8 で、/var/adm/acct/nite ディレクトリ内の各ファイルを簡単に説明します。

表 20-8 /var/adm/acct/nite ディレクトリ内のファイル

ファイル	説明
active	runacct スクリプトが進捗状況の記録用、警告メッセージ、エラーメッセージの出力用として使用する
active.MMDD	runacct スクリプトがエラーを検出した後は、active ファイルに同じ

表 20-8 /var/adm/acct/nite ディレクトリ内のファイル (続き)

ファイル	説明
cms	prdaily が使用する ASCII の合計コマンド要約
ctacct.MMDD	tacct.h 形式の接続アカウントレコード
ctmp	acctcon1 プログラムの出力。ctmp.h 形式の接続セッションレコード (acctcon1 と acctcon2 は互換性を保証するために用意)
daycms	prdaily スクリプトが使用する ASCII 日次コマンド要約
daytacct	tacct.h 形式の 1 日分の合計アカウントレコード
disktacct	tacct.h 形式のディスクアカウントレコード。dodisk スクリプトが作成する
fd2log	runacct スクリプトの実行中の診断出力
lastdate	runacct スクリプトが最後に実行された日 (date +%m%d 書式)
lock	runacct スクリプトの逐次使用の制御に使用する
lineuse	prdaily スクリプトが使用する tty 回線利用状況レポート
log	acctcon プログラムからの診断出力
log.MMDD	runacct スクリプトがエラーを検出した後は、log ファイルと同じ
owtmpx	前日の wtmpx ファイル
reboots	wtmpx ファイルからの開始および終了日付とリブートのリスト
statefile	runacct スクリプトの実行中の現在状態の記録用に使用
tmpwtmp	wtmpfix プログラムが修復した wtmpx ファイル
wtmperror	wtmpfix エラーメッセージが格納される場所
wtmperror.MMDD	runacct スクリプトがエラーを検出した後は、wtmperror ファイルと同じ
wtmp.MMDD	runacct スクリプトが使用する wtmpx ファイルのコピー

sum ディレクトリは、runacct スクリプトが更新し、monacct スクリプトが使用する累積要約ファイルを格納します。次の表で、/var/adm/acct/sum ディレクトリの中にあるファイルを説明します。

表 20-9 /var/adm/acct/sum ディレクトリ内のファイル

ファイル	説明
cms	バイナリ形式の、会計期の合計コマンド要約ファイル
cmsprev	最新の更新がなされていないコマンド要約ファイル

表 20-9 /var/adm/acct/sum ディレクトリ内のファイル (続き)

ファイル	説明
daycms	内部要約書式の、当日の利用状況を表すコマンド要約ファイル
loginlog	各ユーザーが最後にログインした日付のレコード。lastlogin スクリプトによって作成され、prdaily スクリプトによって使用される。
rprt.MMDD	prdaily スクリプトが保存し出力
tacct	会計期の累積合計アカウントングファイル
tacctprev	最新の更新がない点を除いて tacct と同じ
tacct.MMDD	MMDD 日付分の合計アカウントングファイル

fiscal ディレクトリは monacct スクリプトが作成する定期的要約ファイルを格納します。次の表で、/var/adm/acct/fiscal ディレクトリの中にあるファイルを説明します。

表 20-10 /var/adm/acct/fiscal ディレクトリ内のファイル

ファイル	説明
cmsn	内部要約書式の、会計期 <i>n</i> の合計コマンド要約ファイル
fiscrptn	会計期 <i>n</i> の rprtn と同じレポート
tacctn	会計期 <i>n</i> の合計アカウントングファイル

runacct スクリプトが生成するファイル

次の表で、/var/adm/acct ディレクトリ内の runacct スクリプトによって生成された最も役に立つファイルを説明します。

表 20-11 runacct が生成するファイル

ファイル	説明
nite/lineuse	runacct スクリプトは acctcon プログラムを呼び出し、/var/adm/acct/nite/tmpwtmp ファイルから端末の回線利用状況に関するデータを収集し、そのデータを /var/adm/acct/nite/lineuse ファイルに書き込む。prdaily スクリプトはこれらのデータを使用して回線利用状況を報告する。このレポートは特に不良回線の検出に有効となる。ログアウトとログインの回数の比率が 3 対 1 を上回る場合は、回線に障害が起こっている可能性が高い。
nite/daytacct	tacct.h 形式の当日の合計課金ファイル

表 20-11 runacct が生成するファイル (続き)

ファイル	説明
sum/tacct	毎日の nite/daytacct データの累積が含まれており、課金の目的で使用される。monacct スクリプトは、各月または会計期ごとにデータの累積を新たに開始する。
sum/daycms	runacct スクリプトは acctcms プログラムを呼び出し、当日中に使用されたコマンドを処理し、日次コマンド要約レポートを作成して、/var/adm/acct/sum/daycms ファイルにデータを保存する。ASCII バージョンは /var/adm/acct/nite/daycms ファイル
sum/cms	毎日のコマンド要約の累積。monacct スクリプトの実行によって新たに累積を開始する。ASCII バージョンは nite/cms ファイル
sum/loginlog	runacct スクリプトは lastlogin スクリプトを呼び出し、/var/adm/acct/sum/loginlog ファイルのログインの中で最新のログイン日付を更新する。lastlogin コマンドは、すでに有効ではないログインをこのファイルから削除する
sum/rprt.MMDD	runacct スクリプトが実行されるたびに、prdaily スクリプトによって印刷された日次レポートのコピーが保存される

第 21 章

システムパフォーマンスの管理 (トピック)

以下の各章で、システムパフォーマンスの管理について説明します。

第 22 章	システムパフォーマンスについての概要を説明します。
第 23 章	プロセスコマンドを使用してシステムパフォーマンスを向上する手順を説明します。
第 24 章	vmstat、sar、およびディスク利用状態をチェックするコマンドを使用して、システムパフォーマンスを監視する手順を説明します。

第 22 章

システムパフォーマンスの管理 (概要)

コンピュータやネットワークのパフォーマンスを十分に引き出すことは、システム管理における重要な作業です。この章では、コンピュータシステムのパフォーマンスの維持と管理に影響する要素について簡単に説明します。

この章の内容は以下のとおりです。

- 320 ページの「システムパフォーマンスに関する情報の参照箇所」
- 321 ページの「システムパフォーマンスおよびシステム資源」
- 321 ページの「プロセスとシステムのパフォーマンス」
- 323 ページの「パフォーマンスの監視」

システムパフォーマンスの管理に関する新機能

ここでは、システムパフォーマンスの管理に使用する新機能について説明します。

DNLC 機能の向上

ディレクトリ名検索キャッシュ (DNLC) が拡張され、1000 以上のファイルを含む大容量のディレクトリにあるファイルにアクセスする際のパフォーマンスが向上しました。

DNLC は一般的なファイルシステムサービスであり、最近参照されたディレクトリ名とそれに関連する v ノードをキャッシュします。UFS ディレクトリのエントリは、ディスク上に直線的に保存されています。よって、エントリを格納するには、その名

前で各エントリを検索する必要があります。新しいエントリを追加する際は、ディレクトリ全体を検索して、その名前が存在していないことを確認する必要があります。このパフォーマンスに関する問題を解決するために、DNLC を使ってディレクトリ全体をメモリーにキャッシュします。

このリリースにおける DNLC のもう 1 つの新機能は、検索したが存在しなかったファイルオブジェクトをキャッシュすることです。これは「ネガティブキャッシュ」と呼びます。ファイルが存在するかどうかを繰り返し調べるアプリケーションに便利です。

詳細については、『Solaris カーネルのチューンアップ・リファレンスマニュアル』を参照してください。

pargs コマンドと preap コマンド

新しい 2 つのコマンド `pargs` と `preap` により、プロセスのデバッグが改善されています。`pargs` コマンドを使用すると、動作中のプロセスまたはコアファイルに関連付けられた引数と環境変数を表示できます。`preap` コマンドを使用すると、終了した (ゾンビ) プロセスを削除できます。

また、`pargs` コマンドを使用することによって、プロセスに渡された引数を `ps` コマンドで一部しか表示できないという問題がようやく解決されました。次のように、`pargs` コマンドを `pgrep` コマンドと併用して、プロセスに渡された引数を表示できます。

```
# pargs `pgrep ttymon`
579: /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548: /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

`pargs -e` コマンドを使用して、プロセスに関連付けられた環境変数を次の例のように表示できます。

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

`pargs` コマンドと `preap` コマンドを使用して、ユーザーの検査権限に含まれるすべてのプロセスを検査できます。スーパーユーザーは、すべてのプロセスを検査できます。

`preap` コマンドを使用して、終了したプロセスを削除できます。終了したプロセスは、ゾンビプロセスとも呼ばれます。ゾンビプロセスとは、その終了状態がまだ親によって取得または要求されていないプロセスをいいます。これらのプロセスは一般的に支障はありませんが、プロセスの数が多い場合には、システム資源を消費することになります。

`preap` コマンドの使用方法については、`preap(1)` のマニュアルページを参照してください。`pargs` コマンドの使用方法については、`proc(1)` のマニュアルページを参照してください。

パフォーマンスツール

Solaris 管理コンソールのパフォーマンスツールを使って、システムパフォーマンスおよびシステム資源の使用を監視できます。

Solaris 資源管理によるシステム資源の割り当て、監視、および制御については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「Solaris 9 リソースマネージャの紹介」を参照してください。

Solaris 管理コンソールの使用と起動の方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールの操作 (手順)」を参照してください。

新しい FSS スケジューラと FX スケジューラ

FSS スケジューラ (フェアシェアスケジューラ) および FX スケジューラ (固定優先順位スケジューラ) は、新しいユーザープロセスのスケジューラです。両方のスケジューラでは、タイムシェアリング (TS) スケジューリングクラスおよび対話 (IA) スケジューリングクラスと同じ範囲の優先順位 (0 から 59) を使用します。

FX は、プロセスを確実な優先順位で実行する、固定優先順位スケジューラです。FX はデフォルトのスケジューラではありません。TS では負荷を分散します。FX では負荷を分散しません。同じシステムで FX と TS を使用できますが、FX の優先度を高く設定しすぎると、TS で実行するプロセスが停止状態になります。

FSS を使用すると、CPU 資源の配分を明示的にプロジェクトに割り当てることによって、アプリケーションのパフォーマンスが保証されます。その他の FSS の利点は、次のとおりです。

- システムでの CPU 資源の細かい配分を可能にする。
- 作業負荷に対して重要度に基づいて有効な CPU 資源を割り当てるように制御する。作業負荷の重要度は、各作業負荷に割り当てる CPU 資源の配分数によって表されます。

プロセッサセットを使用するのでなければ、同じシステムで FSS を TS または FX と併用しないでください。プロセッサセットを使用する場合、同じ CPU で競合しないように、各プロセッサセットで実行するすべてのプロセスが同じスケジューリングクラスにある限りは、同じシステムで FSS を TS、IA、FX と併用できます。特に、プロセッサセットを使用して FX クラスのアプリケーションで FSS クラスのアプリケーションの停止状態を招く優先順位を使用しないようにするのでなければ、FX スケジューラは新しい FSS スケジューリングクラスと併用しないでください。

次の表に、上記のスケジューラや、プロジェクト、タスクなどの他の Solaris 資源管理に関する機能、システム資源をより良く管理するための情報を示します。

トピック	参照箇所
FSS スケジューラおよび FX スケジューラの使用法	『Solaris のシステム管理 (資源管理とネットワークサービス)』の「フェアシェアスケジューラ」
FSS	FSS (7)
FX	priocntl(1) と dispadmin(1M) のマニュアルページ
プロジェクト	『Solaris のシステム管理 (資源管理とネットワークサービス)』の「プロジェクトとタスク」

システムパフォーマンスに関する情報の参照箇所

作業	参照箇所
プロセスの管理	第 23 章
システムのパフォーマンスの監視	第 24 章
Solaris 調整可能パラメータの変更	『Solaris カーネルのチューンアップ・リファレンスマニュアル』

システムパフォーマンスおよびシステム資源

コンピュータシステムのパフォーマンスは、システムがその資源をどのように使用して割り当てるかによって左右されます。したがって、通常の条件下でどのように動作するかを知るために、システムパフォーマンスを定期的に監視する必要があります。期待できるパフォーマンスについてよく把握し、問題が発生したときに分析できなければなりません。

パフォーマンスに影響を及ぼすシステム資源は次のとおりです。

システム資源	説明
中央処理装置 (CPU)	CPU は、命令をメモリーからフェッチして実行します。
入出力 (I/O) デバイス	I/O デバイスは、コンピュータとの間で情報をやりとりします。この種のデバイスには、端末とキーボード、ディスクドライブ、プリンタなどがあります。
メモリー	物理 (またはメイン) メモリーは、システム上のメモリー (RAM) の容量を示します。

コンピュータシステムの動作とパフォーマンスに関する統計情報を表示するツールについては、[第 24 章](#)を参照してください。

プロセスとシステムのパフォーマンス

次の表は、プロセスに関連する用語を示します。

表 22-1 プロセスに関連する用語

用語	説明
プロセス	システムの動作またはジョブ。システムをブートしてコマンドを実行するか、アプリケーションを起動するたびに、システムは 1 つ以上のプロセスをアクティブにする

表 22-1 プロセスに関連する用語 (続き)

用語	説明
軽量プロセス (LWP)	仮想 CPU または実行資源。LWP は、利用できる CPU 資源をスケジューラクラスと優先順位に基づいて使用するよう、カーネルによってスケジューラされる。LWP には、カーネルスレッドと LWP が含まれる。カーネルスレッドには、メモリーに常駐する情報が入っている。また、LWP には、スワップ可能な情報が入っている
アプリケーションスレッド	ユーザーのアドレス空間内で独立して実行できる別個のスタックを持った一連の命令。アプリケーションスレッドは LWP の最上部で多重化できる

1 つのプロセスは、複数の軽量プロセスと複数のアプリケーションスレッドで構成できます。カーネルはカーネルスレッド構造をスケジューラします。この構造は、SunOS 環境内をスケジューラする実体です。表 22-2 に各種プロセス構造体を示します。

表 22-2 プロセス構造体

構造体	説明
proc	プロセス全体に関連し、メインメモリーに常駐しなければならない情報が入っている
kthread	1 つの LWP に関連し、メインメモリーに常駐しなければならない情報が入っている
user	スワップ可能な、プロセス単位の情報が入っている
klwp	スワップ可能な、LWP プロセス単位の情報が入っている

次の図に、これらのプロセス構造体の関係を示します。

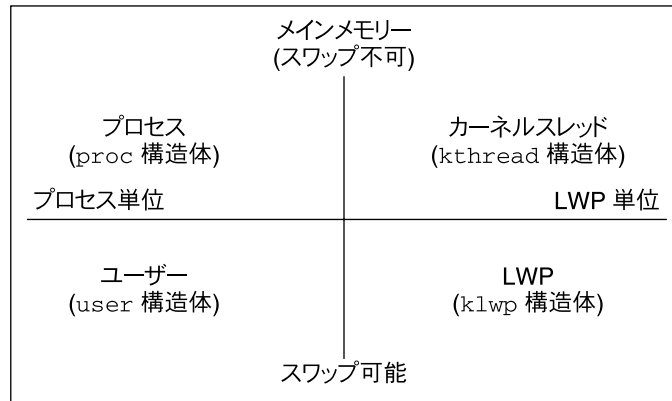


図 22-1 プロセス構造体

プロセス内のすべてのスレッドは、ほとんどのプロセス資源にアクセスできます。ほとんどすべてのプロセスの仮想メモリーが共有されます。あるスレッドが共有データを変更すると、その変更結果をプロセス内の他のスレッドが利用できます。

パフォーマンスの監視

コンピュータの稼働中は、各種のシステム動作を追跡するためにオペレーティングシステムのカウンタが増分されます。追跡されるシステム動作は次のとおりです。

- 中央処理装置 (CPU) の使用状況
- バッファの使用状況
- ディスクとテープの入出力 (I/O) 動作
- 端末デバイスの動作
- システムコールの動作
- コンテキスト切替え
- ファイルアクセス
- 待ち行列の動作
- カーネルテーブル
- プロセス間通信
- ページング
- 空きメモリーとスワップ空間
- カーネルメモリー割り当て (KMA)

監視ツール

Solaris ソフトウェアには、システムパフォーマンスを追跡できるように複数のツールが提供されています。次のような監視ツールがあります。

表 22-3 パフォーマンス監視ツール

コマンド	説明	参照箇所
sar コマンドと sadc コマンド	システム動作データを収集および報告する	第 24 章
ps コマンドと prstat コマンド	活動中のコマンドについての情報を表示する	第 23 章
vmstat コマンドと iostat コマンド	システム動作データの要約。仮想メモリーの統計、ディスクの使用率、CPU の動作など	第 24 章
swap コマンド	ユーザーのシステムで利用可能なスワップ領域についての情報を表示する	『Solaris のシステム管理 (デバイスとファイルシステム)』の「追加スワップ空間の構成 (手順)」
netstat コマンドと nfsstat コマンド	ネットワークパフォーマンスについての情報を表示する	netstat (1M) と nfsstat (1M) のマニュアルページ
Sun Enterprise SyMON	Sun Enterprise レベルのシステム上で、システム動作データを収集する	『Sun Enterprise SyMON 2.0.1 Software User's Guide』

第 23 章

システムプロセスの管理 (手順)

この章では、システムプロセスを管理する手順について説明します。この章で説明する手順は次のとおりです。

- 327 ページの「プロセスを表示する方法」
- 330 ページの「プロセスに関する情報を表示する方法」
- 331 ページの「プロセスを制御する方法」
- 332 ページの「プロセスを終了させる方法」
- 333 ページの「プロセスクラスに関する基本情報を表示する方法 (priocntl)」
- 334 ページの「プロセスのグローバル優先順位を表示する方法」
- 334 ページの「プロセスの優先順位を指定する方法 (priocntl)」
- 335 ページの「タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)」
- 336 ページの「プロセスのクラスを変更する方法 (priocntl)」
- 338 ページの「プロセスの優先順位を変更する方法 (nice)」

システムプロセスの管理に使用するコマンド (概要)

次の表では、プロセス情報を管理するために使用されるコマンドについて説明します。

表 23-1 プロセスを管理するためのコマンド

コマンド	マニュアルページ	説明
ps, pgrep, prstat	ps(1)、pgrep(1)、prstat(1M)	システム上のアクティブなプロセスの状態をチェックする。また、プロセスについての詳細な情報を表示する

表 23-1 プロセスを管理するためのコマンド (続き)

コマンド	マニュアルページ	説明
dispadmin	dispadmin(1M)	デフォルトのスケジューリングポリシーをリストする
priocntl	priocntl(1)	プロセスに優先順位クラスを割り当て、プロセスの優先度を管理する
nice	nice(1)	タイムシェアリングプロセスの優先度を変更する
psrset	psrset(1M)	特定のプロセスグループを、1つのプロセッサではなく、プロセッサのグループに結合する

Solaris 管理コンソールのプロセスツールを使用すると、ユーザーフレンドリなインタフェースによるプロセスを管理できます。Solaris 管理コンソールの使用と起動の方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールの操作 (手順)」を参照してください。

ps コマンド

ps コマンドを使用すると、システム上で活動中のプロセスの状態をチェックできます。また、プロセスについての技術的な情報も表示できます。このデータは、プロセスの優先順位をどのように設定するか判断するなどの管理作業に利用できます。

ps コマンドを使用すると、システム上で活動中のプロセスの状態をチェックできます。使用するオプションに応じて、次の情報が表示されます。

- プロセスの現在の状態
- プロセス ID
- 親プロセス ID
- ユーザー ID
- スケジューリングクラス
- 優先順位
- プロセスのアドレス
- 使用したメモリー
- 使用した CPU 時間

次の表では、ps コマンドを使用して報告されるいくつかのフィールドについて説明します。どのフィールドが表示されるかは、選択するオプションによって異なります。使用可能なすべてのオプションについては、ps(1) のマニュアルページを参照してください。

表 23-2 ps により出力されるフィールド

フィールド	説明
UID	プロセス所有者の実効ユーザー ID
PID	プロセス ID
PPID	親プロセス ID
C	スケジューリングのためのプロセッサ使用率。このフィールドは <code>-c</code> オプションを使用すると表示されない
CLS	プロセスが所属するスケジューリングクラス。リアルタイム、システム、またはタイムシェアリングのいずれか。このフィールドは、 <code>-c</code> オプションを指定した場合にのみ表示される
PRI	カーネルスレッドのスケジューリング優先順位。番号が大きいほど優先順位が高い
NI	プロセスの nice 値。これは、スケジューリング優先順位に影響する。プロセスの nice 値を大きくすると、その優先順位が下がる
ADDR	proc 構造体のアドレス
SZ	プロセスの仮想アドレスサイズ
WCHAN	プロセスが休眠中のイベントまたはロックのアドレス
STIME	プロセスの起動時刻 (時、分、秒)
TTY	プロセス (またはその親プロセス) が起動された端末。疑問符は、制御端末がないことを示す
TIME	プロセスの起動以降に使用した合計 CPU 時間
CMD	プロセスを生成したコマンド

プロセスを表示する方法

ps コマンドを使用して、システム上で実行中のすべてのプロセスを表示することができます。

```
$ ps [-efc]
```

ps	ログインセッションに関連するプロセスのみを表示する
-ef	システム上で実行中のすべてのプロセスに関する詳細情報を表示する
-c	プロセススケジューラ情報を表示する

例 — プロセスを表示する

次の例は、オプションを指定しないときの `ps` コマンドからの出力を示します。

```
$ ps
  PID TTY          TIME CMD
 1664 pts/4        0:06 csh
 2081 pts/4        0:00 ps
```

次の例は、`ps -ef` の出力を示します。この出力例は、システムのブート時に最初に実行されたプロセスが `sched` (スワップ) であり、それに続いて `init` プロセス、`pageout` の順に実行されたことを示しています。

```
$ ps -ef
  UID  PID  PPID  C   STIME TTY          TIME CMD
  root    0    0    0   Dec 20 ?          0:17 sched
  root    1    0    0   Dec 20 ?          0:00 /etc/init -
  root    2    0    0   Dec 20 ?          0:00 pageout
  root    3    0    0   Dec 20 ?          4:20 fsflush
  root   374   367    0   Dec 20 ?          0:00 /usr/lib/saf/ttymon
  root   367    1    0   Dec 20 ?          0:00 /usr/lib/saf/sac -t 300
  root   126    1    0   Dec 20 ?          0:00 /usr/sbin/rpcbind
  root    54    1    0   Dec 20 ?          0:00 /usr/lib/sysevent/syseventd
  root    59    1    0   Dec 20 ?          0:00 /usr/lib/picl/picld
  root   178    1    0   Dec 20 ?          0:03 /usr/lib/autofs/automountd
  root   129    1    0   Dec 20 ?          0:00 /usr/sbin/keyserv
  root   213    1    0   Dec 20 ?          0:00 /usr/lib/lpsched
  root   154    1    0   Dec 20 ?          0:00 /usr/sbin/inetd -s
  root   139    1    0   Dec 20 ?          0:00 /usr/lib/netsvc/yp/ypbind ...
  root   191    1    0   Dec 20 ?          0:00 /usr/sbin/syslogd
  root   208    1    0   Dec 20 ?          0:02 /usr/sbin/nscd
  root   193    1    0   Dec 20 ?          0:00 /usr/sbin/cron
  root   174    1    0   Dec 20 ?          0:00 /usr/lib/nfs/lockd
daemon  175    1    0   Dec 20 ?          0:00 /usr/lib/nfs/statd
  root   376    1    0   Dec 20 ?          0:00 /usr/lib/ssh/sshd
  root   226    1    0   Dec 20 ?          0:00 /usr/lib/power/powerd
  root   315    1    0   Dec 20 ?          0:00 /usr/lib/nfs/mountd
  root   237    1    0   Dec 20 ?          0:00 /usr/lib/utmpd
  .
  .
  .
```

/proc ファイルシステムとコマンド

プロセスコマンドを使用すると、`/proc` ディレクトリにあるプロセスに関する詳細情報を表示できます。`/proc` ディレクトリはプロセスファイルシステム (PROCFS) とも呼ばれます。アクティブなプロセスのイメージは、そのプロセス ID 番号を使って `/proc` に格納されます。

プロセスツールは `ps` コマンドの一部のオプションに似ていますが、このツールから提供される出力の方が詳細です。プロセスコマンドには次の機能があります。

- `fstat` や `fcntl` 情報、作業ディレクトリ、親プロセスと子プロセスからなるツリーなど、プロセスに関する詳細情報を表示します。
- プロセスへの制御を提供して、ユーザーがプロセスを停止または再開できるようにします。

/proc プロセスコマンドを使用したプロセスの管理

いくつかのプロセスコマンドを使用すると、アクティブなプロセスに関する詳細な技術情報を表示したり、そのプロセスを制御したりできます。表 23-3 に、いくつかの `/proc` コマンドを示します。

プロセスが無限ループ内でトラップされた場合や、実行時間が長すぎる場合は、プロセスを終了 (`kill`) できます。 `pkill` コマンドを使用してプロセスを終了する方法については、第 23 章を参照してください。

`/proc` ファイルシステムは、状態情報と制御機能のためのサブディレクトリを含むディレクトリ階層です。

`/proc` ファイルシステムは、ウォッチポイント機能も提供します。この機能は、プロセスのアドレス領域の個々のページの読み取り権または書き込み権を再マップするために使用されます。この機能は制限がなく、MT-safe です。

デバッグ用ツールは、`/proc` の新しいウォッチポイント機能を使用するように変更されています。つまり、ウォッチポイントプロセス全体がより高速になったためです。

`dbx` デバッグ用ツールを使用してウォッチポイントを設定するときの次の制限は取り除かれました。

- SPARC レジスタウィンドウのため、スタック上のローカル変数にウォッチポイントを設定する。
- マルチスレッド化されたプロセスにウォッチポイントを設定する。

詳細については、`proc(4)`、`core(4)`、および `mdb(1)` のマニュアルページを参照してください。

表 23-3 /proc プロセスコマンド

プロセスコマンド	説明
<code>pcred</code>	プロセスの資格情報を表示する
<code>pfiles</code>	プロセス内で開いているファイルに関する <code>fstat</code> 情報と <code>fcntl</code> 情報を表示する
<code>pflags</code>	<code>/proc</code> 追跡フラグ、保留状態のシグナルと保持状態のシグナル、他の状態情報を出力する

表 23-3 /proc プロセスコマンド (続き)

プロセスコマンド	説明
<code>pldd</code>	プロセスにリンクされている動的ライブラリを表示する
<code>pmap</code>	各プロセスのアドレス空間マップを表示する
<code>psig</code>	各プロセスのシグナルの動作とハンドラを表示する
<code>prun</code>	各プロセスを開始する
<code>pstack</code>	各プロセス内の LWP の 16 進とシンボルのスタックトレースを出力する
<code>pstop</code>	各プロセスを停止する
<code>ptime</code>	microstate アカウントを使用してプロセスの時間を測定する
<code>ptree</code>	プロセスを含むプロセスツリーを表示する
<code>pwait</code>	プロセス終了後の状態情報を表示する
<code>pwdx</code>	プロセスの現在の作業ディレクトリを表示する

詳細については、`pgrep(1)` のマニュアルページを参照してください。

▼ プロセスに関する情報を表示する方法

1. 詳細を表示したいプロセスのプロセス ID を表示します。

```
# pgrep process
```

`process` は、詳細を表示したいプロセスの名前です。

プロセス ID は、出力の第 1 列目に表示されます。

2. 表示したいプロセス情報を表示します。

```
# /usr/bin/pcommand pid
```

`pcommand` 実行したいプロセスツールコマンド。表 23-3 を参照

`pid` プロセス ID

例 — プロセスに関する情報を表示する

次の例は、プロセスツールコマンドを使用して `lpsched` プロセスに関する詳細情報を表示する方法を示しています。

```
# pgrep lpsched 1  
213
```

```

# pwdx 213 2
213: /
# ptree 213 3
213 /usr/lib/lpsched
# pfiles 213 4
213: /usr/lib/lpsched
Current rlimit: 4096 file descriptors
0: S_IFIFO mode:0000 dev:270,0 ino:67 uid:0 gid:0 size:0
   O_RDWR
1: S_IFIFO mode:0000 dev:270,0 ino:67 uid:0 gid:0 size:0
   O_RDWR
3: S_IFCHR mode:0666 dev:136,0 ino:35882 uid:0 gid:3 rdev:21,0
   O_WRONLY FD_CLOEXEC
4: S_IFDOOR mode:0444 dev:275,0 ino:18526 uid:0 gid:0 size:0
   O_RDONLY|O_LARGEFILE FD_CLOEXEC door to nscd[208]
5: S_IFREG mode:0664 dev:136,0 ino:64648 uid:71 gid:8 size:0
   O_WRONLY

```

1. lpsched のプロセス ID を表示します。
2. lpsched の現在の作業ディレクトリを表示します。
3. lpsched が入っているプロセスツリーを表示します。
4. fstat と fcntl の情報を表示します。

▼ プロセスを制御する方法

1. 制御するプロセスのプロセス ID を表示します。

```
# pgrep process
```

process は、制御するプロセスの名前です。

プロセスの ID は、出力の第 1 列目に表示されます。

2. 適切なプロセスコマンドを使用してプロセスを制御します。

```
# /usr/bin/pcommand pid
```

pcommand

実行したいプロセスコマンド。表 23-3 を参照

pid

プロセス ID

3. プロセス状態を確認します。

```
# ps -ef | grep PID
```

例 — プロセスを制御する

次の例は、プロセスツールを使用して dtpad を停止し再開する方法を示しています。

```
# pgrep dtpad 1
2921
```

```
# pstop 2921 2  
# prun 2921 3
```

1. dtpad のプロセス ID を表示する
2. dtpad プロセスを停止する
3. dtpad プロセスを再開する

プロセスの終了 (pkill)

プロセスを強制的に終了 (kill) しなければならない場合があります。プロセスが無限ループに入っていたり、大きいジョブを開始したが完了する前に停止したい場合があります。所有しているプロセスであれば、どれでも終了できます。また、スーパーユーザーはプロセス ID が 0、1、2、3、4 のものを除き、システム上のどんなプロセスでも終了できます。プロセス ID が 0、1、2、3、4 のプロセスを終了させると、システムがクラッシュする可能性があります。

詳細については、proc (1) のマニュアルページを参照してください。

▼ プロセスを終了させる方法

1. (省略可能) 別のユーザーが所有するプロセスを終了するには、スーパーユーザーになります。
2. 停止するプロセスのプロセス ID を表示します。

```
$ pgrep process
```

process は、詳細を表示したいプロセスの名前です。
プロセスの ID は、出力の第 1 列目に表示されます。

3. プロセスを停止します。

```
$ pkill [-9] pid
```

-9

プロセスを確実に終了させる

pid

停止するプロセス ID

4. プロセスが停止したことを確認します。

```
$ pgrep process
```

プロセスクラス情報の管理

次のリストは、システム上で構成されるクラスと、タイムシェアリングクラスの利用者優先順位の範囲です。クラスの種類は次のとおりです。

- フェアシェア (FSS)
- 固定優先順位 (FX)
- システム (SYS)
- 対話型 (IA)
- リアルタイム (RT)
- タイムシェアリング (TS)
 - ユーザーが指定する -60 から +60 までの優先順位の範囲
 - プロセスの優先順位は、親プロセスから継承されます。これを「ユーザーモード」の優先順位と呼びます。
 - システムは、ユーザーモードの優先順位をタイムシェアリングディスパッチパラメータテーブル内で検索し、nice または `priocntl` (ユーザー提供) 優先順位に追加し、0 から 59 までの範囲を確保して「グローバル」優先順位を作成します。

プロセスのスケジュール優先順位を変更する (`priocntl`)

プロセスのスケジュール優先順位とは、プロセススケジューラによって割り当てられる優先順位のことです。これらの優先順位は、スケジューラのスケジュールポリシーに従って割り当てられます。 `dispadm` コマンドを使用すると、デフォルトのスケジュールポリシーを表示できます。

`priocntl` コマンドを使用して、プロセスを優先順位クラスに割り当てて、プロセスの優先順位を管理できます。 `priocntl` コマンドを使用してプロセスを管理する手順については、334 ページの「プロセスの優先順位を指定する方法 (`priocntl`)」を参照してください。

プロセスクラスに関する基本情報を表示する方法 (`priocntl`)

`priocntl -l` コマンドを使用すると、プロセスのスケジューリングクラスと優先順位の範囲を表示できます。

```
$ priocntl -l
```

例—プロセスクラスに関する基本情報を表示する (priocntl)

次の例に `priocntl -l` コマンドからの出力を示します。

```
# priocntl -l
CONFIGURED CLASSES
=====

SYS (System Class)

TS (Time Sharing)
    Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
    Configured FX User Priority Range: 0 through 60

IA (Interactive)
    Configured IA User Priority Range: -60 through 60
```

プロセスのグローバル優先順位を表示する方法

`ps` コマンドを使用して、プロセスのグローバル優先順位を表示できます。

```
$ ps -ecl
```

グローバル優先順位は、PRI カラムの下に表示されます。

例 — プロセスのグローバル優先順位を表示する

次の例は、`ps -ecl` コマンドの出力を示します。PRI カラム内のデータは、`pageout` プロセスが最上位の優先順位を持ち、`sh` が最下位の優先順位であることを示しています。

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY      TIME  COMD
19 T 0   0    0   SYS 96   f00d05a8  0   ?         ?         0:03  sched
 8 S 0   1    0   TS 50   ff0f4678 185  ff0f4848 ?         36:51  init
19 S 0   2    0   SYS 98   ff0f4018  0   f00c645c ?         0:01  pageout
19 S 0   3    0   SYS 60   ff0f5998  0   f00d0c68 ?         241:01 fsflush
 8 S 0 269  1   TS 58   ff0f5338 303  ff49837e ?         0:07   sac
 8 S 0 204  1   TS 43   ff2f6008  50   ff2f606e console 0:02   sh
```

▼ プロセスの優先順位を指定する方法 (priocntl)

1. スーパーユーザーになります。
2. 指定した優先順位でプロセスを起動します。

```
# priocntl -e -c class -m userlimit -p pri command-name
```

<code>-e</code>	コマンドを実行する
<code>-c class</code>	プロセスを実行する範囲のクラスを指定する。有効なクラスは TS (タイムシェアリング)、RT (リアルタイム)、IA (対話型)、FSS (フェアシェア)、または FX (固定優先順位)
<code>-m userlimit</code>	<code>-p</code> オプションを使用するときに、優先順位を上下できる最大範囲を指定する
<code>-p pri command-name</code>	リアルタイムスレッド用に RT クラス内で相対優先順位を指定できるようにする。タイムシェアリングプロセスの場合は、 <code>-p</code> オプションを使用すると -60 から +60 までのユーザー指定の優先順位を指定できる

3. プロセス状態を確認します。

```
# ps -ecl | grep command-name
```

例—プロセスの優先順位を指定する (prionctl)

次の例では、ユーザーが指定する最上位の優先順位を使用して `find` コマンドを開始します。

```
# prionctl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
```

▼ タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (prionctl)

1. スーパーユーザーになります。
2. 実行中のタイムシェアリングプロセスのスケジューリングパラメータを変更します。

```
# prionctl -s -m userlimit [-p userpriority] -i idtype idlist
```

<code>-s</code>	ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する
<code>-m userlimit</code>	<code>-p</code> オプションを使用するときに、優先順位を上下できる最大範囲を指定する
<code>-p userpriority</code>	優先順位を指定する

`-i idtype idlist` *idtype* と *idlist* の組み合わせを使用してプロセスを識別する。
idtype では *PID* や *UID* など、ID のタイプを指定する。*idlist*
を使用して、*pid* または *UID* のリストを識別する

3. プロセス状態を確認します。

```
# ps -ecl | grep idlist
```

例—タイムシェアリングプロセスのスケジューリングパラメータを変更する (`priocntl`)

次の例では、500 ミリ秒のタイムスライス、クラス RT 内の優先順位 20、グローバル優先順位 120 を指定して、コマンドを実行します。

```
# priocntl -e -c RT -t 500 -p 20 myprog  
# ps -ecl | grep myprog
```

▼ プロセスのクラスを変更する方法 (`priocntl`)

1. (省略可能) スーパーユーザーになります。

注—プロセスをリアルタイムプロセスに変更したり、リアルタイムプロセスから変更したりするには、ユーザーはスーパーユーザーであるか、リアルタイムシェル内で作業中でなければなりません。

2. プロセスのクラスを変更します。

```
# priocntl -s -c class -i idtype idlist
```

`-s` ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する

`-c class` クラス TS またはクラス RT を指定して、プロセスのクラスを変更する

`-i idtype idlist` *idtype* と *idlist* の組み合わせを使用してプロセスを識別する。
idtype では *PID* や *UID* など、ID のタイプを指定する。*idlist*
を使用して、*pid* または *UID* のリストを識別する

3. プロセス状態を確認します。

```
# ps -ecl | grep idlist
```


例—プロセスのクラスを変更する (priosctl)

次の例では、ユーザー 15249 が所有するすべてのプロセスをリアルタイムプロセスに変更します。

```
# priosctl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

注 - スーパーユーザーとしてユーザープロセスをリアルタイムクラスに変更すると、そのユーザーは `priosctl -s` を使用して、リアルタイムのスケジューリングパラメータを変更できません。

タイムシェアリングプロセスの優先順位を変更する (nice)

`nice` コマンドは、SunOS の旧バージョンとの下位互換性を保つためのみサポートされます。 `priosctl` コマンドを使用する方がプロセスを柔軟に管理できます。

プロセスの優先順位は、そのスケジュールクラスポリシーと `nice` 値によって決定されます。各タイムシェアリングプロセスは、グローバル優先順位を持っています。グローバル優先順位は、ユーザーが指定した優先順位 (`nice` コマンドまたは `priosctl` コマンドの影響を受ける) とシステムで計算された優先順位を加算して算出されます。

プロセスの実行優先順位番号は、オペレーティングシステムによって割り当てられ、スケジュールクラス、使用される CPU 時間、 `nice` 値 (タイムシェアリングプロセスの場合) などの複数の要素によって決定されます。

各タイムシェアリングプロセスは、親プロセスから継承したデフォルトの `nice` 値で起動します。 `nice` 値は、 `ps` レポートの `NI` カラムに表示されます。

ユーザーは、自分が与える `nice` 値優先順位を大きくしてプロセスの優先順位を下げる事ができます。ただし、 `nice` 値を小さくしてプロセスの優先順位を上げることができるのは、スーパーユーザー (または `root`) だけです。これは、ユーザーが各自のプロセスの優先順位を大きくして CPU の独占比率を高めるのを防ぐためです。

`nice` 値の範囲は 0 から +39 までで、0 が最上位の優先順位となります。各タイムシェアリングプロセスのデフォルトの `nice` 値は 20 です。 `nice` コマンドには利用できるバージョンが 2 つあり、1 つは標準バージョンの `/usr/bin/nice`、もう 1 つは C シェルの組み込みコマンドです。

▼ プロセスの優先順位を変更する方法 (nice)

注 - この節では /usr/bin/nice コマンドの構文についてだけ説明し、C シェル nice 組み込みコマンドについての説明は行いません。C シェル nice コマンドについては、`cs(1)` のマニュアルページを参照してください。

1. まず、ユーザーとしてコマンドの優先順位を下げるのか、あるいはスーパーユーザーとしてコマンドの優先順位を上げるのか下げるのかを決定してから、次のうちの 1 つを選択します。

- 手順 2 の例に従って、ユーザーとしてコマンドの優先順位を下げる。
- 手順 3 に従って、スーパーユーザーとしてコマンドの優先順位を上げるか、または下げる。

2. ユーザーとして、**nice** 値を大きくすることでコマンドの優先順位を下げます。

次の nice コマンドは、nice 値を 5 単位分大きくすることで、*command-name* を実行する優先順位を下げます。

```
$ /usr/bin/nice -5 command-name
```

上記のコマンドでは、マイナス記号は次にくるものがオプションであることを表します。このコマンドは、次のように指定することもできます。

```
% /usr/bin/nice -n 5 command-name
```

次の nice コマンドは、nice 値をデフォルトの 10 単位分大きくすることで、*command-name* の優先順位を下げます。ただし、最大値の 39 を超えさせることはできません。

```
% /usr/bin/nice command-name
```

3. スーパーユーザーとして、**nice** 値を変更することでコマンドの優先順位を上げるか、または下げます。

次の nice コマンドは、nice 値を 10 単位分小さくすることで、*command-name* の優先順位を上げます。ただし、最低値の 0 未満にすることはできません。

```
# /usr/bin/nice --10 command-name
```

上記のコマンドでは、最初のマイナス記号は次にくるものがオプションであることを表します。2 番目のマイナス記号は負の数を表します。

次の nice コマンドは、nice 値を 5 単位分大きくすることで、*command-name* の優先順位を下げます。ただし、最高値の 39 を超えさせることはできません。

```
# /usr/bin/nice -5 command-name
```

詳細については、`nice(1)` のマニュアルページを参照してください。

システムのプロセスに関する問題解決方法

すでに判明している問題の解決方法のヒントを次に示します。

- 同じユーザーが所有する複数の同じジョブがないかどうかを調べます。ジョブが終了するまで待たずに多数のバックグラウンドジョブを起動するスクリプトを実行した場合に、この問題が発生することがあります。
- CPU 時間が大量に増えているプロセスがないかどうかを調べます。この問題を調べるには、`ps` 出力の `TIME` フィールドを確認します。そのプロセスが無限ループに入っている可能性があります。
- 実行中のプロセスの優先順位が高すぎないかどうかを調べます。`ps -c` コマンドを使用して `CLS` フィールドを調べると、各プロセスのスケジューラクラスが表示されます。リアルタイム (RT) プロセスとして実行中のプロセスが CPU を独占している可能性があります。また、`nice` 値の大きいタイムシェアリング (TS) プロセスがないかどうかを調べます。スーパーユーザー特権を持つユーザーが、このプロセスの優先順位を上げすぎた可能性があります。システム管理者は、`nice` コマンドを使用して優先順位を下げるすることができます。
- 制御がきかなくなったプロセスを調べます。このようなプロセスは、CPU 時間の使用が継続的に増加していきます。プロセスが開始 (STIME) されたときに調べるか、しばらくの間 CPU 時間 (TIME) が累計されるのを観察していると、この問題が発生しているかどうかを判断できます。

第 24 章

システムパフォーマンスの監視 (手順)

この章では、`vmstat`、`iostat`、`df`、または `sar` コマンドを使用してシステムパフォーマンスを監視する手順について説明します。この章で説明する手順は次のとおりです。

- 343 ページの「仮想メモリーの統計情報を表示する方法 (`vmstat`)」
- 344 ページの「システムイベント情報を表示する方法 (`vmstat -s`)」
- 345 ページの「スワップの統計情報を表示する方法 (`vmstat -S`)」
- 345 ページの「キャッシュフラッシュの統計情報を表示する方法 (`vmstat -c`)」
- 346 ページの「各デバイス当りの割り込み数を表示する方法 (`vmstat -i`)」
- 346 ページの「ディスクの使用状況を表示する方法 (`iostat`)」
- 348 ページの「拡張ディスク統計情報を表示する方法 (`iostat -xtc`)」
- 349 ページの「ディスク容量情報を表示する方法 (`df`)」
- 350 ページの「ファイルアクセスをチェックする方法 (`sar -a`)」
- 351 ページの「バッファ動作をチェックする方法 (`sar -b`)」
- 353 ページの「システムコールの統計情報をチェックする方法 (`sar -c`)」
- 354 ページの「ディスク動作をチェックする方法 (`sar -d`)」
- 355 ページの「ページアウトとメモリーをチェックする方法 (`sar -g`)」
- 356 ページの「カーネルメモリーの割り当てをチェックする方法 (`sar -k`)」
- 358 ページの「プロセス間通信をチェックする方法 (`sar -m`)」
- 359 ページの「ページイン動作をチェックする方法 (`sar -p`)」
- 360 ページの「待ち行列動作をチェックする方法 (`sar -q`)」
- 361 ページの「未使用のメモリーをチェックする方法 (`sar -r`)」
- 362 ページの「CPU の使用状況をチェックする方法 (`sar -u`)」
- 364 ページの「システムテーブルの状態をチェックする方法 (`sar -v`)」
- 365 ページの「スワップ動作をチェックする方法 (`sar -w`)」
- 366 ページの「端末動作をチェックする方法 (`sar -y`)」
- 368 ページの「システム全体のパフォーマンスをチェックする方法 (`sar -A`)」
- 371 ページの「自動データ収集を設定する方法」

システムパフォーマンスの監視に関する 新機能

vmstat コマンドには、`faults-in` 列によるすべてのシステム割り込みが追加されています。

従来の Solaris リリースでは、この列でクロックデバイスの割り込みが報告されませんでした。

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

仮想メモリーの統計情報の表示 (vmstat)

vmstat コマンドを使用すると、仮想メモリーの統計情報と、CPU の負荷、ページング、コンテキスト切替え数、デバイス割り込み、システムコールなどのシステムイベントに関する情報を表示できます。また、vmstat コマンドを使用すると、スワップ、キャッシュフラッシュ、および割り込みに関する統計情報も表示できます。

次の表に vmstat の出力内のフィールドを示します。

表 24-1 vmstat コマンドからの出力

カテゴリ	フィールド名	説明
procs		次の状態を報告する
	r	ディスクパッチ待ち行列内のカーネルスレッド数
	b	資源を待機中のブロックされたカーネルスレッド数
	w	資源処理の完了を待機中のスワップアウトされた軽量プロセス数
memory		実メモリーと仮想メモリーの使用状況を表示する
	swap	使用可能なスワップ空間
	free	空きリストのサイズ
page		ページフォルトとページング動作を 1 秒当りの単位数として表示する

表 24-1 vmstat コマンドからの出力 (続き)

カテゴリ	フィールド名	説明
	re	回収されたページ数
	mf	軽度のフォルトと重大なフォルト
	pi	ページインされたキロバイト数
	po	ページアウトされたキロバイト数
	fr	解放されたキロバイト数
	de	最後にスワップインされたプロセスに必要だと予想されるメモリー
	sr	ページデーモンによって走査され、現在は使用されていないページ数。sr が 0 以外の値であれば、ページデーモンは実行されている
disk		最高 4 台のディスク上のデータを示す、1 秒当りのディスク処理数を表示する
faults		トラップ/割り込み率 (1 秒当り) を表示する
	in	1 秒当りの割り込み数
	sy	1 秒当りのシステムコール数
	cs	CPU のコンテキスト切替え率
cpu		CPU 時間の使用状況を表示する
	us	ユーザー時間
	sy	システム時間
	id	アイドル時間

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

仮想メモリーの統計情報を表示する方法 (vmstat)

秒単位で時間間隔を指定して vmstat コマンドを使用すると、仮想メモリーの統計情報が収集されます。

```
$ vmstat n
```

n は、報告間隔を秒単位で表した時間間隔です。

例 — 仮想メモリの統計情報を表示する

次の例に、5 秒間隔で収集された統計情報に関する `vmstat` の表示を示します。

```
$ vmstat 5
procs      memory                page                    disk                    faults              cpu
r  b  w  swap free re  mf  pi  po  fr de sr f0 s3 -- --  in  sy  cs us sy id
0  0  8 28312 668 0   9   2   0   1  0  0  0  1  0  0  10  61  82  1  2  97
0  0  3 31940 248 0  10  20   0  26  0  27  0  4  0  0  53 189 191  6  6  88
0  0  3 32080 288 3  19  49   6  26  0  15  0  9  0  0  75 415 277  6 15  79
0  0  3 32080 256 0  26  20   6  21  0  12  1  6  0  0 163 110 138  1  3  96
0  1  3 32060 256 3  45  52  28  61  0  27  5 12  0  0 195 191 223  7 11  82
0  0  3 32056 260 0   1   0   0   0  0  0  0  0  0  0  4  52  84  0  1  99
```

システムイベント情報を表示する方法 (`vmstat -s`)

`vmstat -s` コマンドを実行すると、システムを前回ブートした後に発生した各種システムイベントの合計が表示されます。

```
$ vmstat -s
      0 swap ins
      0 swap outs
      0 pages swapped in
      0 pages swapped out
392182 total address trans. faults taken
20419 page ins
  923 page outs
30072 pages paged in
 9194 pages paged out
65167 total reclaims
65157 reclaims from free list
      0 micro (hat) faults
392182 minor (as) faults
19383 major faults
85775 copy-on-write faults
66637 zero fill page faults
46309 pages examined by the clock daemon
      6 revolutions of the clock hand
15578 pages freed by the clock daemon
4398 forks
  352 vforks
4267 execs
12926285 cpu context switches
109029866 device interrupts
499296 traps
22461261 system calls
778068 total name lookups (cache hits 97%)
 18739 user   cpu
 34662 system cpu
52051435 idle  cpu
 25252 wait   cpu
```


スワップの統計情報を表示する方法 (vmstat -S)

vmstat -S を実行すると、スワップの統計情報が表示されます。

```
$ vmstat -S
procs      memory          page          disk          faults        cpu
r  b  w  swap  free  si  so pi po fr de sr f0 s0 s6 --  in  sy  cs us sy id
0  0  0 200968 17936  0  0  0  0  0  0  0  0  0  0  109  43  24  0  0 100
```

スワッピング統計情報を次の表に示します。その他のフィールドの説明については、[表 24-1](#) を参照してください。

表 24-2 vmstat -S コマンドからの出力

フィールド名	説明
si	1 秒当りにスワップされた平均軽量プロセス数
so	スワップアウトされた全プロセス数

注 - vmstat コマンドは、これらの両フィールドを出力しません。スワップ統計情報の詳細情報を表示するには、sar コマンドを使用してください。

キャッシュフラッシュの統計情報を表示する方法 (vmstat -c)

vmstat -c を実行すると、仮想キャッシュのキャッシュフラッシュ統計情報が表示されます。

```
$ vmstat -c
usr      ctx      rgn      seg      pag      par
0      60714      5 134584 4486560 4718054
```

出力には、今回のブート後に発生したキャッシュフラッシュの合計数が表示されます。キャッシュタイプを次の表に示します。

表 24-3 vmstat -c コマンドからの出力

キャッシュ名	キャッシュタイプ
usr	ユーザ
ctx	コンテキスト
rgn	領域

表 24-3 vmstat -c コマンドからの出力 (続き)

キャッシュ名	キャッシュタイプ
seg	セグメント
pag	ページ
par	ページの一部

各デバイス当りの割り込み数を表示する方法 (vmstat -i)

vmstat -i コマンドを実行すると、デバイス当りの割り込み数が表示されます。

例 — 各デバイス当りの割り込み数を表示する

次の例は、vmstat -i コマンドからの出力を示します。

```
$ vmstat -i
interrupt          total          rate
-----
clock              52163269      100
esp0                2600077       4
zsc0                25341         0
zsc1                48917         0
cgsixc0             459           0
lec0                400882        0
fdc0                 14            0
bppc0                0             0
audiocs0            0             0
-----
Total              55238959      105
```

ディスク使用状況の表示 (iostat n)

iostat コマンドを使用すると、ディスクの入出力に関する統計情報を表示し、スループット、使用率、待ち行列の長さ、トランザクション率、サービス時間の計測結果を表示できます。このコマンドの詳細は、iostat(1M) のマニュアルページを参照してください。

ディスクの使用状況を表示する方法 (iostat)

秒単位で時間間隔を指定して iostat コマンドを使用すると、ディスクの使用状況が表示されます。

```
$ iostat 5
      tty          fd0          sd3          nfs1          nfs31          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0   1   0   0   410   3   0   29   0   0   9   3   0   47   4   2   0 94
```

出力の1行目は、今回のブート以降の統計情報を示します。2行目以降は、時間間隔ごとの統計情報を示します。デフォルトでは、端末 (tty)、ディスク (fd と sd)、CPU (cpu) の統計情報が表示されます。

次の表に iostat コマンド出力内のフィールドを示します。

表 24-4 iostat n コマンドからの出力

デバイスの種類	フィールド名	説明
Terminal		
	tin	端末の入力待ち行列内の文字数
	tout	端末の出力待ち行列内の文字数
ディスク		
	bps	1秒当りのブロック数
	tps	1秒当りのトランザクション数
	serv	ミリ秒単位で表した平均サービス時間
CPU		
	us	ユーザーモード
	sy	システムモード
	wt	入出力待機中
	id	アイドル状態

例 — ディスクの使用状況を表示する

次の例は、5秒間隔で収集されるディスク統計情報を示します。

```
$ iostat 5
      tty          sd0          sd6          nfs1          nfs49          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0   0   1   0   49   0   0   0   0   0   0   0   0   15   0   0   0 100
  0  47   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
  0  16  44   6  132   0   0   0   0   0   0   0   0   0   0   0   1   99
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100
```

```

0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100
0 16 3 1 23 0 0 0 0 0 0 0 0 0 0 0 99
0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100
0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100
0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100

```

拡張ディスク統計情報を表示する方法 (iostat -xtc)

iostat -xtc コマンドを実行すると、拡張ディスク統計情報が表示されます。

```

$ iostat -xtc
extended device statistics
device      r/s    w/s    kr/s    kw/s  wait  actv  svc_t  %w  %b  tty      tout  us  sy  wt  id
fd0         0.0    0.0    0.0    0.0  0.0  0.0   0.0    0  0   0        0    0  0  0 100
sd0         0.0    0.0    0.4    0.4  0.0  0.0  49.5    0  0
sd6         0.0    0.0    0.0    0.0  0.0  0.0   0.0    0  0
nfs1        0.0    0.0    0.0    0.0  0.0  0.0   0.0    0  0
nfs49       0.0    0.0    0.0    0.0  0.0  0.0  15.1    0  0
nfs53       0.0    0.0    0.4    0.0  0.0  0.0  24.5    0  0
nfs54       0.0    0.0    0.0    0.0  0.0  0.0   6.3    0  0
nfs55       0.0    0.0    0.0    0.0  0.0  0.0   4.9    0  0

```

iostat -xtc コマンドを使用すると、ディスクごとに 1 行ずつ出力が表示されます。出力フィールドを次の表に示します。

表 24-5 iostat -xtc コマンドからの出力

フィールド名	説明
r/s	1 秒当りの読み取り数
w/s	1 秒当りの書き込み数
kr/s	1 秒当りの読み取りキロバイト数
kw/s	1 秒当りの書き込みキロバイト数
wait	サービス (待ち行列の長さ) を待機中の平均トランザクション数
actv	サービス中の平均トランザクション数
svc_t	ミリ秒単位で表した平均サービス時間
%w	待ち行列が空でない時間の割合
%b	ディスクがビジーである時間の割合

ディスク容量統計情報の表示 (df)

df コマンドを使用すると、マウントされている各ディスク上の空きディスク容量が表示されます。レポート用の統計情報では使用可能容量の合計の内先頭に 10% の空き容量を残しておくので、df から報告される「使用可能」ディスク容量は全容量の 90% のみに相当します。この先頭の空き容量は、パフォーマンスを高めるために常に空になっています。

実際に df コマンドからレポートされるディスク容量の割合は、使用済み容量を使用可能容量で割った値です。

ファイルシステムの容量が 90% を超える場合は、cp コマンドを使用して空いているディスクにファイルを転送するか、tar または cpio コマンドを使用してテープに転送できます。または、ファイルを削除できます。

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

ディスク容量情報を表示する方法 (df)

df -k コマンドを使用すると、ディスク容量情報がキロバイト単位で表示されます。

```
$ df -k
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0   192807    40231  133296    24%      /
```

次の表に df -k の出力内のフィールドを示します。

表 24-6 df -k コマンドからの出力

フィールド名	説明
kbytes	ファイルシステム内の使用可能容量の合計
used	使用されている容量
avail	使用可能容量
capacity	使用されている容量が全容量に占める割合
mounted on	マウントポイント

例 — ファイルシステム情報を表示する

次の例は、df -k コマンドの出力を示します。

```
$ df -k
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0   384120    131596  214112    39%      /
```

/dev/dsk/c0t0d0s6	1388419	1050390	282493	79%	/usr
/proc	0	0	0	0%	/proc
mnttab	0	0	0	0%	/etc/mnttab
fd	0	0	0	0%	/dev/fd
swap	467152	40	467112	1%	/var/run
swap	467160	48	467112	1%	/tmp
/dev/dsk/c0t0d0s4	1784644	1525360	205745	89%	/export
venus:/usr/dist	20612581	13237316	6963015	66%	/usr/dist

システム動作の監視 (sar)

次のような目的には、sar コマンドを使用します。

- システム動作についてのデータを編成し表示する
- 特殊な要求に基づいて、システム動作データにアクセスする
- システムパフォーマンスを測定および監視するレポートを自動的に生成する。また、特定のパフォーマンス障害を正確に突き止めるための、特殊な要求レポートも生成する。これらのツールについては、[368 ページの「システム動作データの自動収集 \(sar\)」](#) を参照

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

ファイルアクセスをチェックする方法 (sar -a)

sar -a コマンドを使用すると、ファイルアクセス操作の統計情報が表示されます。

```
$ sar -a
SunOS touchstone 5.9 Generic sun4u      03/04/2003

00:00:00  iget/s namei/s dirbk/s
01:00:00      0      3      0
02:00:00      0      3      0
03:00:00      0      3      0
04:00:00      0      3      0
05:00:00      0      3      0
06:00:00      0      3      0
07:00:00      0      3      0
08:00:00      0      3      0
08:20:01      0      3      0
08:40:00      0      3      0
09:00:00      0      3      0
09:20:01      0     10      0
09:40:01      0      1      0
10:00:02      0      5      0
```

```
Average          0          4          0
```

次の表に、`sar -a` コマンドによってレポートされるオペレーティングシステムのルーチンを示します。

表 24-7 `sar -a` コマンドからの出力

フィールド名	説明
<code>iget/s</code>	ディレクトリ名検索キャッシュ (DNLC) に入っていない <code>i</code> ノードに対して出された要求数
<code>namei/s</code>	1 秒当りのファイルシステムパスの検索数。 <code>namei</code> で DNLC 内にディレクトリ名が見つからない場合は、 <code>iget</code> が呼び出され、ファイルまたはディレクトリの <code>i</code> ノードが取得される。したがって、ほとんどの <code>igets</code> は DNLC が欠落した結果である
<code>dirbk/s</code>	1 秒間に実行されたディレクトリブロックの読み取り回数

表示される値が大きいほど、カーネルはユーザーファイルへのアクセスに長い時間を費やしています。この時間には、プログラムとアプリケーションによるファイルシステムの使用量が反映されます。 `-a` オプションを使用すると、アプリケーションのディスク依存度を表示できるので便利です。

バッファー動作をチェックする方法 (`sar -b`)

`sar -b` コマンドを使用すると、バッファー動作の統計情報が表示されます。

バッファーは、`i` ノード、シリンダグループブロック、間接ブロックなどのメタデータをキャッシュするために使用されます。

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0       0    100      0       0     55      0       0
```

次の表は、`-b` オプションを指定したときに表示されるバッファー動作を示します。

表 24-8 `sar -b` コマンドからの出力

フィールド名	説明
<code>bread/s</code>	ディスクからバッファークッシュに投入された 1 秒当りの平均読み取り数
<code>lread/s</code>	バッファークッシュからの 1 秒当りの平均論理読み取り数
<code>%rcache</code>	バッファークッシュ内で見つかった論理読み込み数の小数部 (<code>lread/s</code> に対する <code>bread/s</code> の比を 100% から差し引いた値)

表 24-8 sar -b コマンドからの出力 (続き)

フィールド名	説明
bwrit/s	バッファークャッシュからディスクに書き込まれた 1 秒当りの平均物理ブロック数 (512 ブロック)
lwrit/s	バッファークャッシュへの 1 秒当りの平均論理書き込み数
%wcache	バッファークャッシュ内で見つかった論理書き込み数の小数部 (lwrit/s に対する bwrit/s の比を 100% から差し引いた値)
pread/s	キャラクタ型デバイスインタフェースを使用する 1 秒当りの平均物理読み取り数
pwrit/s	キャラクタ型デバイスインタフェースを使用する 1 秒当りの平均物理書き込み要求数

最も重要なエントリは、キャッシュヒット率 %rcache と %wcache です。この 2 つのエントリは、システムバッファリングの効率を測定します。%rcache が 90% 未満の場合や、%wcache が 65% 未満の場合は、バッファ領域を大きくすればパフォーマンスを改善できる可能性があります。

例 — バッファー動作をチェックする

次の sar -b 出力の例は、すべてのデータは許容範囲に収まっているので、%rcache バッファーと %wcache バッファーが処理速度低下の原因ではないことを示します。すべてのデータは許容範囲に収まっています。

```
$ sar -b
SunOS touchstone 5.9 Generic sun4u      03/04/2003

00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0      50      0      0
02:00:00      0      0    100      0      0      45      0      0
03:00:00      0      0    100      0      0      72      0      0
04:00:00      0      0    100      0      0      43      0      0
05:00:00      0      0    100      0      0      42      0      0
06:00:00      0      0    100      0      0      48      0      0
07:00:00      0      0    100      0      0      42      0      0
08:00:00      0      0    100      0      0      48      0      0
08:20:01      0      0    100      0      0      42      0      0
08:40:00      0      0    100      0      0      48      0      0
09:00:00      0      0    100      0      0      43      0      0
09:20:01      0      2     99      0      2      83      0      0
09:40:01      0      0     98      0      0      61      0      0
10:00:02      0      1     99      0      0      62      0      0

Average      0      0     99      0      0      74      0      0
```


システムコールの統計情報をチェックする方法 (sar -c)

sar -c コマンドを使用すると、システムコールの統計情報が表示されます。

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38      2      2    0.00    0.00   149    120
```

次の表は、-c オプションを指定したときに報告されるシステムコールのカテゴリを示します。一般に、読み取りと書き込みはシステムコール合計の約半分を占めますが、割合はシステムで実行中の動作によって大幅に変動します。

表 24-9 sar -c コマンドからの出力

フィールド名	説明
scall/s	1 秒当たりのすべてのタイプのシステムコール数 (通常は、4 ないし 6 ユーザーのシステム上で 1 秒当たり約 30)
sread/s	1 秒当たりの read システムコール数
swrit/s	1 秒当たりの write システムコール数
fork/s	1 秒当たりの fork システムコール数 (4 ないし 6 ユーザーのシステム上で毎秒約 0.5)。この数値は、シェルスクリプトの実行中は大きくなる
exec/s	1 秒当たりの exec システムコール数。exec/s を fork/s で割った値が 3 より大きい場合は、効率の悪い PATH 変数を調べる
rchar/s	read システムコールによって転送される 1 秒当たりの文字 (バイト) 数
wchar/s	write システムコールによって転送される 1 秒当たりの文字 (バイト) 数

例 — システムコールの統計情報をチェックする

次の例は、sar -c コマンドからの出力を示します。

```
$ sar -c
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      118      14      9    0.00    0.00   3360   2506
02:00:00      118      14      9    0.00    0.00   3360   2505
03:00:00      118      14      9    0.00    0.00   3361   2506
04:00:00      118      14      9    0.01    0.00   3363   2506
05:00:00      118      14      9    0.00    0.00   3360   2505
06:00:00      118      14      9    0.00    0.00   3360   2505
07:00:00      118      14      9    0.00    0.00   3360   2505
08:00:00      118      14      9    0.00    0.00   3360   2505
```

```

08:20:01      120       14        9      0.00      0.00      3394      2522
08:40:00      119       14        9      0.01      0.01      3357      2502
09:00:00      119       14        9      0.01      0.00      3356      2500
09:20:01      437       76       48      0.04      0.04     19606     6531
09:40:01      585      117       73      0.00      0.00     12962    10119
10:00:02      564       97       60      0.03      0.03     16354     6196

Average      159       22       14      0.01      0.01     4657     3017

```

ディスク動作をチェックする方法 (sar -d)

sar -d コマンドを使用すると、ディスク動作の統計情報が表示されます。

```

$ sar -d
00:00:00  device          %busy  avque  r+w/s  blks/s  await  avserv
01:00:00  fd0              0      0.0    0       0       0.0    0.0

```

次の表は、-d オプションを指定したときに報告されるディスクデバイスの動作を示します。

表 24-10 sar -d コマンドからの出力

フィールド名	説明
device	監視中のディスクデバイス名
%busy	デバイスが転送要求のサービスに費やす時間の割合
avque	平均待ち時間と平均サービス時間の合計
r+w/s	デバイスへの 1 秒当りの読み取り転送数と書き込み転送数
blks/s	デバイスに転送される 1 秒当たりの 512 バイトブロック数
await	待ち行列内の転送要求の平均アイドル時間 (ミリ秒単位)。これは、待ち行列が占有されている場合のみ計測される
avserv	デバイスが 1 転送要求の処理に要する平均時間 (ミリ秒単位)。ディスクの場合は、この値にシークタイム、回転待ち時間、データ転送時間が含まれる

待ち行列内に何かがあるときは、待ち行列の長さで待ち時間が計測されるので注意してください。%busy の値が小さい場合に、待ち行列とサービス時間が大きければ、変更されたブロックをディスクに随時書き込むために、システムが定期的に処理していることを示す場合があります。

例 — ディスク動作をチェックする

次の例は、sar -d コマンドからの一部省略した出力を示します。

```

$ sar -d
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00  device          %busy  avque  r+w/s  blks/s  await  avserv
01:00:00  dad0              0      0.0    0       0      18.1   14.9
           dad0,a          0      0.0    0       0       0.0    0.0
           dad0,b          0      0.0    0       0       0.0   15.1
           dad0,c          0      0.0    0       0       0.0    0.0
           dad0,d          0      0.0    0       0      20.1   14.9
           dad0,e          0      0.0    0       0       0.0    0.0
           dad0,g          0      0.0    0       0       0.0    0.0
           dad0,h          0      0.0    0       0       0.0    0.0
           fd0             0      0.0    0       0       0.0    0.0
           nfs1            0      0.0    0       0       0.0    0.0
           nfs2            0      0.0    0       0       2.9    4.4
           nfs3            0      0.0    0       0       0.0    0.0
           sd0             0      0.0    0       0       0.0    0.0

```

ページアウトとメモリーをチェックする方法 (sar -g)

sar -g コマンドを使用すると、ページアウト動作の平均値とメモリー解放動作の平均値とが表示されます。

```

$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00    0.00    0.00    0.00    0.00    0.00

```

sar -g コマンドで表示される出力は、より多くのメモリーが必要かどうかを判断するのに役立ちます。ps -elf コマンドを使用すると、ページデーモンに使用される CPU サイクル数が表示されます。サイクル数が大きく、pgfree/s と pgscan/s の値が大きければ、メモリー不足を示します。

また、sar -g を使用すると、i ノードの再利用間隔が短すぎるために、再利用可能なページが失われているかどうか也表示されます。

次の表に -g オプションからの出力について説明します。

表 24-11 sar -g コマンドからの出力

フィールド名	説明
pgout/s	1 秒間にページアウトされた要求数
ppgout/s	1 秒間に実際にページアウトされたページ数 (1 つのページアウト要求で複数のページがページアウトされることがある)
pgfree/s	空きリストに配置された 1 秒当りのページ数

表 24-11 sar -g コマンドからの出力 (続き)

フィールド名	説明
pgscan/s	ページデーモンによって走査された 1 秒当りのページ数。この値が大きい場合は、ページデーモンが空きメモリのチェックに大量の時間を費やしている。これは、メモリーを増やす必要があることを示す
%ufs_ipf	ufs がそれに関連付けられた再使用可能ページを持つ iget によって空きリストから取り出された割合。これらのページはフラッシュされ、プロセスが回収できなくなる。したがって、これはページフラッシュを伴う igets の割合である。値が大きければ、i ノードの空きリストがページ境界であり、ufs の i ノード数を増やす必要があることを示す

例 — ページアウトとメモリーをチェックする

次の例は、sar -g コマンドからの出力を示します。

```
$ sar -g
SunOS touchstone 5.9 Generic sun4u      03/04/2003

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00      0.00      0.00      0.00      0.00      0.00
02:00:00      0.01      0.01      0.01      0.00      0.00
03:00:00      0.00      0.00      0.00      0.00      0.00
04:00:00      0.00      0.00      0.00      0.00      0.00
05:00:00      0.00      0.00      0.00      0.00      0.00
06:00:00      0.00      0.00      0.00      0.00      0.00
07:00:00      0.00      0.00      0.00      0.00      0.00
08:00:00      0.00      0.00      0.00      0.00      0.00
08:20:01      0.00      0.00      0.00      0.00      0.00
08:40:00      0.00      0.00      0.00      0.00      0.00
09:00:00      0.00      0.00      0.00      0.00      0.00
09:20:01      0.05      0.52      1.62     10.16      0.00
09:40:01      0.03      0.44      1.47      4.77      0.00
10:00:02      0.13      2.00      4.38     12.28      0.00
10:20:03      0.37      4.68     12.26     33.80      0.00

Average      0.02      0.25      0.64      1.97      0.00
```

カーネルメモリーの割り当てをチェックする方法 (sar -k)

sar -k コマンドを使用すると、KMA (Kernel Memory Allocator) に関して次の動作が表示されます。

KMA を使用すると、カーネルサブシステムは必要に応じてメモリーを割り当て、解放できます。最大量のメモリーを静的に割り当てるのではなく、ピークを下回る負荷を要求するのが予想されるため、KMA はメモリー要求を「小」(256 バイト未満)、

「大」(512 バイト ~ 4K バイト)、「サイズ超過」(4K バイト超)という 3 つのカテゴリに分けます。また、2 つのメモリープールを管理して、「小」要求と「大」要求を満たします。「サイズ超過」要求は、システムページアロケータからメモリーを割り当てることで満たされます。

KMA 資源を使用するドライバや STREAMS の作成に使用中のシステムを調査する場合は、`sar -k` コマンドを使用すると便利です。それ以外の場合は、このコマンドで提供される情報は不要です。KMA 資源を使用するが、終了前には特に資源を返さないドライバやモジュールがあると、メモリーのリークが生じることがあります。メモリーリークが発生すると、KMA によって割り当てられるメモリーは時間が経つにつれて増大します。したがって、`sar -k` コマンドの `alloc` フィールドの値が時間が経つにつれ増える場合は、メモリーリークの可能性があります。メモリーリークのもう 1 つの兆候は、要求が失敗することです。この問題が発生した場合は、メモリーリークのために KMA がメモリーを予約したり割り当てたりできなくなっている可能性があります。

メモリーリークが発生した場合は、KMA からメモリーを要求したが返していないドライバや STREAMS がないかどうかをチェックする必要があります。

```
$ sar -k
00:00:00 sml_mem   alloc   fail   lg_mem   alloc   fail   ovsz_alloc   fail
01:00:00 2523136 1866512 0 18939904 14762364 0 360448 0
02:00:02 2523136 1861724 0 18939904 14778748 0 360448 0
```

次の表に `-k` オプションからの出力について説明します。

表 24-12 `sar -k` コマンドからの出力

フィールド名	説明
<code>sml_mem</code>	KMA が小メモリー要求プール内で使用できるメモリーのバイト数 (小要求は 256 バイト未満)
<code>alloc</code>	KMA が小メモリー要求プールから小メモリー要求に割り当てたメモリーのバイト数
<code>fail</code>	少量のメモリーで失敗した要求数
<code>lg_mem</code>	KMA が大メモリー要求プール内で使用できるメモリーのバイト数 (大要求は 512 バイトから 4K バイトまで)
<code>alloc</code>	KMA が大メモリー要求プールから大メモリー要求に割り当てたメモリーのバイト数
<code>fail</code>	大メモリーで失敗した要求数
<code>ovsz_alloc</code>	サイズ超過要求 (4K バイトを超える要求) に割り当てられたメモリーの容量。これらの要求はページアロケータによって満たされる。このため、プールはない
<code>fail</code>	サイズ超過メモリーで失敗した要求数

例 — カーネルメモリーの割り当てをチェックする (sar)

次の例は、sar -k 出力を示します。

```
$ sar -k
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:05 sml_mem  alloc  fail  lg_mem  alloc  fail  ovsz_alloc  fail
01:00:03 1761280 1209760    0 4833280 3955104    0    2293760    0
02:00:05 1761280 1209020    0 4833280 3953360    0    2293760    0
03:00:03 1761280 1208712    0 4833280 3972096    0    2293760    0
04:00:03 1761280 1209796    0 4833280 3976432    0    2293760    0
05:00:03 1761280 1207708    0 4833280 3976864    0    2293760    0
06:00:04 1761280 1208112    0 4833280 3977200    0    2293760    0
07:00:04 1761280 1209024    0 4833280 3977200    0    2293760    0
08:00:04 1761280 1208932    0 4833280 4035120    0    2293760    0
08:20:04 1761280 1210480    0 4833280 4035120    0    2293760    0
08:40:04 1761280 1210592    0 4833280 4035968    0    2293760    0
09:00:03 1761280 1212440    0 4833280 4039024    0    2293760    0
09:20:04 1761280 1214388    0 4833280 4043792    0    2293760    0
09:40:04 1761280 1210760    0 4833280 4041744    0    2293760    0
10:00:05 1761280 1208192    0 4833280 4040496    0    2293760    0
10:20:03 1761280 1213404    0 4833280 4046576    0    2293760    0
10:40:02 1761280 1212184    0 4833280 4046576    0    2293760    0

Average 1761280 1210219    0 4833280 4009542    0    2293760    0
```

プロセス間通信をチェックする方法 (sar -m)

sar -m コマンドを使用すると、プロセス間通信の動作が表示されます。

```
$ sar -m
00:00:00  msg/s  sema/s
01:00:00  0.00  0.00
```

通常、これらの数字は、メッセージやセマフォを使用するアプリケーションを実行していない限りゼロ (0.00) です。

次の表に -m オプションからの出力について説明します。

表 24-13 sar -m コマンドからの出力

フィールド名	説明
msg/s	1 秒当りのメッセージ処理 (送受信) 数
sema/s	1 秒当りのセマフォ処理数

例 — プロセス間通信をチェックする

次の例は、sar -m コマンドからの一部省略した出力を示します。

```

$ sar -m
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00  msg/s  sema/s
01:00:00  0.00   0.00
02:00:02  0.00   0.00
03:00:00  0.00   0.00
04:00:00  0.00   0.00
05:00:01  0.00   0.00
06:00:00  0.00   0.00

Average    0.00   0.00

```

ページイン動作をチェックする方法 (sar -p)

sar -p コマンドを使用すると、保護フォルトや変換フォルトを含むページイン動作が表示されます。

```

$ sar -p
00:00:00  atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00   0.07   0.00    0.00    0.21    0.39    0.00

```

次の表に -p オプションから報告される統計情報を示します。

表 24-14 sar -p コマンドからの出力

フィールド名	説明
atch/s	現在メモリーに入っているページを回収して満たされる 1 秒当りのページフォルト数 (1 秒当りの付加数)。この例には、空きリストから無効なページを回収し、別のプロセスに現在使用中のテキストページを共有する処理が含まれる (たとえば、複数のプロセスが同じプログラムテキストにアクセスしている場合など)
pgin/s	ファイルシステムがページイン要求を受信する 1 秒当りの回数
ppgin/s	ページインされる 1 秒当りのページ数。ソフトロック要求 (slock/s を参照) などの 1 つのページイン要求や、大型ブロックサイズでは、複数のページがページインされることがある
pflt/s	保護エラーによるページフォルト数。保護フォルトの例には、ページへの不正なアクセスや、「書き込み時コピー」などがある。通常、この数値は主に「書き込み時コピー」からなっている
vflt/s	1 秒当りのアドレス変換ページフォルト数。これは、有効性フォルトと呼ばれ、所定の仮想アドレスに有効なプロセステーブルエントリが存在しないときに発生する

表 24-14 sar -p コマンドからの出力 (続き)

フィールド名	説明
slock/s	物理入出力を要求するソフトウェアロック要求によって発生する1秒当りのフォルト数。ソフトロック要求の発生例には、ディスクからメモリーへのデータ転送などがある。システムはデータを受信しないページをロックするので、別のプロセスはそれを回収して使用できない

例 — ページイン動作をチェックする

次の例は、sar -p コマンドからの出力を示します。

```
$ sar -p
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00  atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00    1.00    0.00    0.00    0.26    1.13    0.00
02:00:00    1.00    0.00    0.00    0.26    1.13    0.00
03:00:00    1.01    0.00    0.00    0.29    1.17    0.00
04:00:00    1.03    0.00    0.00    0.34    1.27    0.00
05:00:00    1.00    0.00    0.00    0.26    1.13    0.00
06:00:00    1.00    0.00    0.00    0.26    1.13    0.00
07:00:00    1.00    0.00    0.00    0.26    1.13    0.00
08:00:00    1.00    0.00    0.00    0.26    1.13    0.00
08:20:01    1.03    0.00    0.00    0.32    1.26    0.00
08:40:00    1.03    0.00    0.00    0.32    1.26    0.00
09:00:00    1.03    0.00    0.00    0.42    1.43    0.00
09:20:01    2.17    1.37    1.95    2.59   50.06    0.00
09:40:01    0.09    0.52    0.63    0.27  105.42    0.00
10:00:02    0.65    2.59    3.11    1.19   42.19    0.00
10:20:03    1.69   11.45   13.77    4.36   39.58    0.00
10:40:01    0.08    0.31    0.32    1.56   12.23    0.00

Average    1.00    0.51    0.62    0.55    8.79    0.00
```

待ち行列動作をチェックする方法 (sar -q)

sar -q コマンドを使用すると、待ち行列に要求が入っている平均待ち行列の長さ、その間の時間の割合が表示されます。

```
$ sar -q
00:00:00  runq-sz  %runocc  swpq-sz  %swpocc
```

次の表に -q オプションからの出力について説明します。

表 24-15 sar -q コマンドの出力

フィールド名	説明
runq-sz	CPU を実行するためにメモリー内で待機中のカーネルスレッド数。通常、この値は 2 未満になる。値が常に 2 より大きい場合は、システムが CPU の限界に到達している可能性がある
%runocc	ディスクパッチ待ち行列が使用されている時間の割合
swpq-sz	sar では出力されなくなった
%swpocc	sar では出力されなくなった

例 — 待ち行列動作をチェックする

次の例は、sar -q コマンドからの出力を示します。%runocc の値が大きく (90 パーセント超)、runq-sz が 2 より大きい場合は、CPU の負荷が大きく、応答速度が低下しています。この場合は、CPU の容量を増やしてシステムの応答速度を適正化する必要があります。

```
$ sar -q
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00      0.0      0      0.0      0
02:00:00      0.0      0      0.0      0
03:00:00      0.0      0      0.0      0
04:00:00      1.0      0      0.0      0
05:00:00      0.0      0      0.0      0
06:00:00      0.0      0      0.0      0
07:00:00      1.0      0      0.0      0
08:00:00      0.0      0      0.0      0
08:20:01      0.0      0      0.0      0
08:40:00      0.0      0      0.0      0
09:00:00      1.0      0      0.0      0
09:20:01      1.6      3      0.0      0
09:40:01      1.5      8      0.0      0
10:00:02      1.6      7      0.0      0
10:20:03      1.5      2      0.0      0
10:40:01      1.9      1      0.0      0
11:00:01      1.4      0      0.0      0
11:20:01      1.6      0      0.0      0

Average      1.6      1      0.0      0
```

未使用のメモリーをチェックする方法 (sar -r)

sar -r コマンドを使用すると、現在使用されていないメモリーページ数とスワップファイルのディスクブロック数が表示されます。

```
$ sar -r
00:00:00 freemem freeswap
01:00:00      2135   401922
```

次の表に `-r` オプションからの出力について説明します。

表 24-16 `sar -r` コマンドからの出力

フィールド名	説明
freemem	コマンドによるサンプル収集間隔の間にユーザープロセスに利用できる平均メモリーページ数。ページサイズはマシンに応じて異なる
freeswap	ページスワップに使用可能な 512 バイトのディスクブロック数

例 — 未使用のメモリーをチェックする

次の例は、`sar -r` コマンドからの出力を示します。

```
$ sar -r
SunOS touchstone 5.9 Generic sun4u      03/04/2003

00:00:00 freemem freeswap
01:00:00      3454   834587
02:00:00      3449   834549
03:00:00      3446   834552
04:00:00      3440   834722
05:00:00      3433   834535
06:00:00      3432   834557
07:00:00      3426   834526
08:00:00      3423   834502
08:20:01      3425   835158
08:40:00      3421   834453
09:00:00      3422   834473
09:20:01      2455   817457
09:40:01         651   768238
10:00:02         862   748302
10:20:03         373   713652
10:40:01         506   717459
11:00:01         668   716669
11:20:01         537   715802

Average      2907   815627
```

CPU の使用状況をチェックする方法 (`sar -u`)

`sar -u` コマンドを使用すると、CPU 使用状況の統計情報が表示されます。

```
$ sar -u
00:00:00      %usr      %sys      %wio      %idle
01:00:00          0          0          0      100
```

オプションを指定しない sar コマンドは、sar -u と同じです。特定の瞬間では、プロセッサはビジー状態またはアイドル状態になっています。ビジー状態のときは、プロセッサはユーザーモードまたはシステムモードになっています。アイドル状態のときは、プロセッサは入出力の完了を待っているか、何も処理することがないので「待機」している状態です。

次の表に -u オプションからの出力について説明します。

表 24-17 sar -u コマンドからの出力

フィールド名	説明
%usr	プロセッサがユーザーモードになっている時間の割合が表示される
%sys	プロセッサがシステムモードになっている時間の割合が表示される
%wio	プロセッサがアイドル状態で入出力の完了を待っている時間の割合が表示される
%idle	プロセッサがアイドル状態で入出力を待っていない時間の割合が表示される

一般に、%wio の値が大きい場合は、ディスクの処理速度が低下していることを意味します。

例 — CPU の使用状況をチェックする

次の例は、sar -u コマンドからの出力を示します。

```
$ sar -u
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00    %usr    %sys    %wio    %idle
01:00:00         0         0         0    100
02:00:00         0         0         0    100
03:00:00         0         0         0    100
04:00:00         0         0         0    100
05:00:00         0         0         0    100
06:00:00         0         0         0    100
07:00:00         0         0         0    100
08:00:00         0         0         0    100
08:20:01         0         0         0    100
08:40:00         0         0         0    100
09:00:00         0         0         0    100
09:20:01         6         1         1     91
09:40:01        15         2         0     82
10:00:02        13         1         2     83
10:20:03         5         2         8     84
10:40:01         3         1         0     96
11:00:01         1         0         1     97
```

```

11:20:01      2      0      0      97
Average      2      0      0      98

```

システムテーブルの状態をチェックする方法 (sar -v)

sar -v コマンドを使用すると、プロセステーブル、i ノードテーブル、ファイルテーブル、および共有メモリーレコードテーブルの状態が表示されます。

```

$ sar -v
00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov  lock-sz
01:00:00  43/922    0  2984/4236  0  322/322    0  0/0

```

次の表に -v オプションからの出力について説明します。

表 24-18 sar -v コマンドからの出力

フィールド名	説明
proc-sz	現在カーネル内で使用されているか、割り当てられているプロセスエントリ (proc 構造体) の数
inod-sz	メモリー内の合計 i ノード数とカーネル内で割り当て済みの最大 i ノード数の比。これは厳密な上限ではなく、超えることもできる
file-sz	開いているシステムファイルテーブルのサイズ。ファイルテーブルには領域が動的に割り当てられるので、sz は 0 として表示される
ov	現在カーネル内で使用されているか割り当てられている共有メモリーレコードテーブルのエントリ数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、sz は 0 として表示される
lock-sz	現在カーネル内で使用されているか割り当てられている共有メモリーレコードテーブルのエントリ数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、sz は 0 として表示される

例 — システムテーブルの状態をチェックする

次の例は、sar -v コマンドからの一部省略した出力を示します。この例は、すべてのテーブルに十分なサイズがあり、オーバーフローは発生しないことを示します。これらのテーブルには、いずれも物理メモリーの容量に基づいて領域が動的に割り当てられます。

```

$ sar -v
SunOS touchstone 5.9 Generic sun4u    03/04/2003

```

```

00:00:00  proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
02:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
03:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
04:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
05:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
06:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
07:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
08:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
08:20:01  64/1898   0  4280/8467  0   0/0       0   0/0
08:40:00  64/1898   0  4280/8467  0   0/0       0   0/0
09:00:00  64/1898   0  4280/8467  0   0/0       0   0/0
09:20:01  70/1898   0  4281/8467  0   0/0       0   0/0
09:40:01  70/1898   0  4281/8467  0   0/0       0   0/0
10:00:02  71/1898   0  4285/8467  0   0/0       0   0/0

```

スワップ動作をチェックする方法 (sar -w)

sar -w コマンドを使用すると、スワッピングと切り替え動作が表示されます。

```

$ sar -w
00:00:00  swpin/s  bswin/s  swpot/s  bswot/s  pswch/s
01:00:00   0.00    0.0     0.00    0.0      22

```

次の表に、対象となる値と説明を示します。

表 24-19 sar -w コマンドからの出力

フィールド名	説明
swpin/s	メモリーに転送される 1 秒当りの軽量プロセス数
bswin/s	メモリーからスワップアウトされる 1 秒当りの平均プロセス数。 この数値が 1 より大きい場合は、メモリーを増やす必要がある
swpot/s	メモリーからスワップアウトされる 1 秒当りの平均プロセス数。 この数値が 1 より大きい場合は、メモリーを増やす必要がある
bswot/s	スワップアウト用に転送される 1 秒当りのブロック数
pswch/s	1 秒当りのカーネルスレッド切り替え数

注 - すべてのプロセスのスワップインには、プロセスの初期化が含まれます。

例 — スワップ動作をチェックする

次の例は、sar -w コマンドからの出力を示します。

```

$ sar -w
SunOS touchstone 5.9 Generic sun4u   03/04/2003

```

```

00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 63
02:00:00 0.00 0.0 0.00 0.0 60
03:00:00 0.00 0.0 0.00 0.0 62
04:00:00 0.00 0.0 0.00 0.0 63
05:00:00 0.00 0.0 0.00 0.0 63
06:00:00 0.00 0.0 0.00 0.0 61
07:00:00 0.00 0.0 0.00 0.0 60
08:00:00 0.00 0.0 0.00 0.0 61
08:20:01 0.00 0.0 0.00 0.0 63
08:40:00 0.00 0.0 0.00 0.0 63
09:00:00 0.00 0.0 0.00 0.0 63
09:20:01 0.00 0.0 0.00 0.0 132
09:40:01 0.00 0.0 0.00 0.0 187
10:00:02 0.00 0.0 0.00 0.0 185
10:20:03 0.00 0.0 0.00 0.0 258
10:40:01 0.00 0.0 0.00 0.0 161
11:00:01 0.00 0.0 0.00 0.0 128
11:20:01 0.00 0.0 0.00 0.0 117
11:40:00 0.00 0.0 0.00 0.0 171
12:00:01 0.00 0.0 0.00 0.0 94
12:20:01 0.00 0.0 0.00 0.0 95
12:40:01 0.00 0.0 0.00 0.0 95
13:00:01 0.00 0.0 0.00 0.0 95
13:20:01 0.00 0.0 0.00 0.0 96
13:40:01 0.00 0.0 0.00 0.0 98
14:00:01 0.00 0.0 0.00 0.0 96
14:20:00 0.00 0.0 0.00 0.0 91
14:40:01 0.00 0.0 0.00 0.0 92
15:00:01 0.00 0.0 0.00 0.0 103

Average 0.00 0.0 0.00 0.0 88

```

端末動作をチェックする方法 (sar -y)

sar -y コマンドを使用すると、端末デバイスの動作を監視できます。

```

$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s madmin/s
01:00:00 0 0 0 0 0 0

```

大量の端末入出力がある場合は、このレポートを使用して不良な行がないかどうかを判別できます。次の表に、記録される動作を示します。

表 24-20 sar -y コマンドからの出力

フィールド名	説明
rawch/s	1 秒当りの入力文字数 (raw 待ち行列)

表 24-20 sar -y コマンドからの出力 (続き)

フィールド名	説明
canch/s	標準待ち行列で処理される 1 秒当りの文字数
outch/s	1 秒当りの出力文字数 (出力待ち行列)
rcvin/s	1 秒当りの受信側ハードウェア割り込み数
xmtin/s	1 秒当りの送信側ハードウェア割り込み数
mdmin/s	1 秒当りのモデム割り込み数

1 秒当りのモデム割り込み数 (mdmin/s) は、0 に近い値になります。また、1 秒当りの送受信側ハードウェア割り込み数 (xmtin/s と rcvin/s) は、それぞれ着信または発信文字数以下になります。そうでない場合は、不良回線がないかどうかをチェックしてください。

例 — 端末動作をチェックする

次の例は、sar -y コマンドからの出力を示します。

```
$ sar -y
SunOS touchstone 5.9 Generic sun4u    03/04/2003

00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0
02:00:00      0      0      0      0      0      0
03:00:00      0      0      0      0      0      0
04:00:00      0      0      0      0      0      0
05:00:00      0      0      0      0      0      0
06:00:00      0      0      0      0      0      0
07:00:00      0      0      0      0      0      0
08:00:00      0      0      0      0      0      0
08:20:01      0      0      0      0      0      0
08:40:00      0      0      0      0      0      0
09:00:00      0      0      0      0      0      0
09:20:01      0      0      6      0      0      0
09:40:01      0      0      0      0      0      0
10:00:02      0      0      1      0      0      0
10:20:03      0      0      6      0      0      0
10:40:01      0      0     15      0      0      0
11:00:01      0      0     20      0      0      0
11:20:01      0      0      4      0      0      0
11:40:00      0      0      4      0      0      0
12:00:01      0      0      0      0      0      0

Average      0      0      2      0      0      0
```

システム全体のパフォーマンスをチェックする方法 (sar -A)

sar -A コマンドを使用すると、すべてのオプションを指定した場合と同じように、システム全体を示す統計情報が表示されます。

このコマンドを使用すると、全体像を把握できます。複数のタイムセグメントからのデータが表示される場合は、レポートに平均値が含まれます。

システム動作データの自動収集 (sar)

システム動作データを自動的に収集するには、sadc、sa1、sa2 という3つのコマンドを使用します。

sadc データ収集ユーティリティは、システム動作に関するデータを定期的に収集し、24時間ごとに1つのファイルに2進形式で保存します。sadc コマンドを定期的に (通常は1時間ごとに) 実行したり、システムがマルチユーザーモードでブートするときにも実行するように設定できます。データファイルは、/var/adm/sa ディレクトリに格納されます。各ファイルには sadd という名前が与えられます。この場合、dd は現在の日付です。このコマンドの書式は次のとおりです。

```
/usr/lib/sa/sadc [t n] [ofile]
```

このコマンドは、t 秒 (5 秒より長くする必要があります) 間隔でサンプルデータを n 回収集します。次に、2進形式 ofile ファイルまたは標準出力に書き込みます。

ブート時に sadc コマンドを実行する

カウンタが0にリセットされることから統計情報を記録するために、sadc コマンドをシステムのブート時に実行する必要があります。sadc をブート時に確実に実行するために、日ごとのデータファイルにレコードを書き込むコマンド行が /etc/init.d/perf ファイルに含まれます。

コマンドエントリの書式は次のとおりです。

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```


sa1 スクリプトを使用して sadc コマンドを定期的に実行する

定期的にレコードを生成するには、sadc コマンドを定期的に実行する必要があります。そのためには、`/var/spool/cron/crontab/sys` ファイルの次の行をコメント解除するのが最も簡単な方法です。

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

`sys` の `crontab` エントリによって、次のように動作します。

- 最初の 2 つの `crontab` エントリによって、月曜から金曜までは午前 8 時から午後 5 時まで 20 分ごとに、それ以外の曜日は 1 時間ごとに、レコードが `/var/adm/sa/sadd` ファイルに書き込まれます。
- 3 番目のエントリは、月曜から金曜までは 1 時間ごとに、レコードを `/var/adm/sa/sardd` ファイルに書き込み、すべての `sar` オプションが含まれます。

これらのデフォルトは、必要に応じて変更できます。

sa2 スクリプトを使用してレポートを生成する

もう 1 つのシェルスクリプト `sa2` は、2 進データファイルではなくレポートを生成します。`sa2` コマンドは `sar` コマンドを呼び出して、レポートファイルに ASCII 出力を書き込みます。

データの自動収集を設定する (sar)

`sar` コマンドを使用すると、システム動作データそのものを収集するか、`sadc` コマンドで作成された日ごとの動作ファイルに収集された情報をレポートできます。

`sar` コマンドの書式は次のとおりです。

```
sar [-aAbcdgkmpqruvw] [-o file] t [n]
sar [-aAbcdgkmpqruvw] [-s time] [-e time] [-i sec] [-f file]
```

次の `sar` コマンドは、オペレーティングシステム内の累積動作カウンタから t 秒間隔で n 回データを収集します。 t は、5 秒以上の値にします。それ以外の値にすると、コマンドそのものがサンプルに影響を与えることがあります。また、サンプルの収集間隔を指定する必要があります。指定しないと、このコマンドは第 2 の書式に従って動作します。 n のデフォルト値は 1 です。次の例では、10 秒間隔で 2 つのサンプルが収集されます。`-o` オプションを指定すると、サンプルは 2 進形式でファイルに保存されます。

```
$ sar -u 10 2
```

その他に、sar では次の点に注意する必要があります。

- サンプル間隔またはサンプル数を指定しなければ、sar コマンドはデータを以前に記録されたファイルから抽出します。その場合は、-f オプションで指定したファイル、またはデフォルトでは最新日付分の標準の日ごとの動作ファイル /var/adm/sa/sadd から抽出されます。
- -s オプションと -e オプションでは、レポートの開始時刻と終了時刻を定義します。開始時刻と終了時刻の書式は hh[:mm[:ss]] です (この場合、h、m、s は、それぞれ時間、分、秒を表します)。
- -i オプションでは、レコードの選択間隔を秒単位で指定します。-i オプションを指定しなければ、日ごとの動作ファイル内で見つかったすべての間隔がレポートされます。

次の表に sar コマンドのオプションとその動作を示します。

表 24-21 sar コマンドのオプション

オプション	動作
-a	ファイルアクセス操作をチェックする
-b	バッファ動作をチェックする
-c	システムコールをチェックする
-d	各ブロックデバイスの動作をチェックする
-g	ページアウトとメモリの解放をチェックする
-k	カーネルメモリの割り当てをチェックする
-m	プロセス間通信をチェックする
-p	スワップとディスパッチ動作をチェックする
-q	待ち行列動作をチェックする
-r	未使用メモリーをチェックする
-u	CPU の使用率をチェックする
-nv	システムテーブルの状態をチェックする
-w	ボリュームのスワッピングと切り替えをチェックする
-y	端末動作をチェックする
-A	システム全体のパフォーマンスをレポートする (すべてのオプションを入力した場合と同じです)

オプションを使用しなければ、-u オプションを指定してコマンドを呼び出すのと同じです。

自動データ収集を設定する方法

1. スーパーユーザーになります。
2. `/etc/init.d/perf` ファイルを編集して次の行をコメント解除します。

```
# if [ -z "$_INIT_RUN_LEVEL" ]; then
#     set -- `/usr/bin/who -r`
#     _INIT_RUN_LEVEL="$7"
#     _INIT_RUN_NPREV="$8"
#     _INIT_PREV_LEVEL="$9"
# fi
#
# if [ $_INIT_RUN_LEVEL -ge 2 -a $_INIT_RUN_LEVEL -le 4 -a \
#     $_INIT_RUN_NPREV -eq 0 -a \( $_INIT_PREV_LEVEL = 1 -o \
#     $_INIT_PREV_LEVEL = S \) ]; then
#
#     /usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
# fi
```

このバージョンの `sadc` コマンドは、カウンタが0にリセットされる時間(ブート時)を示す特殊なレコードを書き込みます。

3. `crontab` ファイル `/var/spool/cron/crontabs/sys` の次の行をコメント解除します。

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```


第 25 章

Solaris ソフトウェアで発生する問題の 解決 (トピック)

以下の各章で、Solaris ソフトウェアで発生する問題を解決する手順を説明します。

第 26 章	一般的なソフトウェアの問題を解決する方法の概要と、システムクラッシュを解決する手順を説明します。
第 27 章	コアファイルを管理する手順を説明します。
第 28 章	クラッシュダンプを保存する手順とシステムエラー記録をカスタマイズする手順を説明します。
第 29 章	一般的なソフトウェアの問題 (システムがハングする、システムがブートしないなど) の状況と可能な解決策を説明します。
第 30 章	一般的なファイルアクセスについての問題 (正しくないコマンド検索パスやファイルのアクセス権など) の解決策を説明します。
第 31 章	一般的なプリンタの問題 (出力が出ない、出力が正しくないなど) の解決策を説明します。
第 32 章	UFS ファイルシステム以外の問題について、特定の fsck エラーメッセージとその解決策を説明します。
第 33 章	ソフトウェアパッケージを追加および削除するときに発生する問題について、特定のエラーメッセージと可能な解決策を説明します。

第 26 章

ソフトウェアの問題解決 (概要)

この章では、ソフトウェアの問題の解決についての概要を説明します。システムクラッシュの問題の解決とシステムメッセージの表示などが含まれます。

この章の内容は次のとおりです。

- 375 ページの「ソフトウェアの問題解決に関する新機能」
- 377 ページの「ソフトウェアの問題の解決方法の参照先」
- 377 ページの「システムクラッシュの問題の解決」
- 379 ページの「システムクラッシュを解決するためのチェックリスト」
- 380 ページの「システムメッセージの表示」
- 382 ページの「システムのメッセージ記録のカスタマイズ」

ソフトウェアの問題解決に関する新機能

この節では、Solaris 9 リリースでの新機能について説明します。

新しいシステムログローテーション

Solaris 9 リリースでは、システムログファイルローテーションが、ルートの `crontab` ファイルのエントリから `logadm` コマンドによって実行されます。
`/usr/lib/newsyslog` スクリプトは使用されません。

この新しいシステムログローテーションは、`/etc/logadm.conf` ファイルに定義されます。このファイルには、`syslogd` などのプロセスのログローテーションエントリが含まれています。たとえば、`/etc/logadm.conf` ファイルにある 1 つのエントリは、`/var/log/syslog` ファイルが空でなければ、そのローテーションが毎週実行されます。つまり、最新の `syslog` ファイルが `syslog.0` になり、その次に新しい `syslog` ファイルが `syslog.1` になります。最新からさかのぼって 8 つまでの `syslog` ログファイルが保存されます。

また、`/etc/logadm.conf` ファイルには、最後のログローテーション実行時のタイムスタンプも含まれます。

`logadm` コマンドを使用して、必要に応じてシステムログをカスタマイズしたり、`/etc/logadm.conf` ファイルにログを追加したりすることができます。

たとえば、Apache アクセスとエラーログのローテーションを実行するには、次のコマンドを使用します。

```
# logadm -w /var/apache/logs/access_log -s 100m
# logadm -w /var/apache/logs/error_log -s 10m
```

この例では、Apache の `access_log` ファイルのローテーションは、そのサイズが 100M バイトに達したときに実行され、そのファイル名に `.0`、`.1` などのように接尾辞が付けられます。また、古い `access_log` ファイルのコピーが 10 個保存されます。また、`error_log` のローテーションは、そのサイズが 10M バイトに達したときに実行され、`access_log` ファイルと同様に、接尾辞が付けられ、コピーが作成されます。

前述の Apache ログローテーションの例における `/etc/logadm.conf` エントリの例は、次のようになります。

```
# cat /etc/logadm.conf
.
.
.
/var/apache/logs/error_log -s 10m
/var/apache/logs/access_log -s 100m
```

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

スーパーユーザーでログインするか、同等の役割 (ログ管理の権限を持つ) でアクセスすることによって、`logadm` コマンドを使用できます。役割によるアクセス制御 (RBAC) を設定すれば、`logadm` コマンドへのアクセス権を与えることで、`root` 以外のユーザーにログ管理の権限を与えることができます。

たとえば、次のエントリを `/etc/user_attr` ファイルに追加して、`logadm` コマンドを使用できる `andy` をユーザーに与えます

```
andy:::profiles=Log Management
```

または、Solaris 管理コンソールを使用して、ログ管理の役割を設定できます。役割の設定に関する詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

新しい root アカウントに対するシェルのフォールバック

Solaris の前のリリースでは、root のシェルを存在しないシェルに変更すると、CD またはネットワークからシステムをブートし、/etc/passwd ファイルにある root シェルを修正する必要がありました。

Solaris 9 リリースでは、root に存在しないシェルを間違っても入力しても、次のいずれかを実行すると、root のシェルが自動的に /sbin/sh にフォールバックされます。

- su コマンドを使用して、スーパーユーザー (root) のアカウントになる。
- シングルユーザーモードのプロンプトで root のパスワードを入力する。

詳細については、su (1M) のマニュアルページを参照してください。

ソフトウェアの問題の解決方法の参照先

問題解決の手順	参照箇所
システムクラッシュ情報の管理	第 28 章
コアファイルの管理	第 27 章
リポート失敗、バックアップ問題などのソフトウェアの問題解決	第 29 章
ファイルアクセスの問題解決	第 30 章
印刷の問題解決	第 31 章
UFS ファイルシステムの不整合の解決	第 32 章
ソフトウェアパッケージの問題解決	第 33 章

システムクラッシュの問題の解決

Solaris オペレーティング環境が動作しているシステムがクラッシュした場合は、クラッシュダンプファイルを含む、可能なかぎりの情報を購入先に提供してください。

システムがクラッシュした場合の対処方法

最も重要なことは、次のとおりです。

1. システムのコンソールメッセージを書き取ります。

システムがクラッシュした場合は、システムをリポートする前に、まずコンソール画面にメッセージが表示されていないか確認してください。このようなメッセージは、クラッシュした原因を解明するのに役立ちます。システムが自動的にリブートして、コンソールメッセージが画面から消えた場合でも、システムエラーログファイル `/var/adm/messages` を表示すれば、これらのメッセージをチェックできます。システムエラーログファイルを表示する方法の詳細は、[381 ページ](#)の「システムメッセージを表示する方法」を参照してください。

クラッシュが頻繁に発生して、その原因を特定できない場合は、システムのコンソールや `/var/adm/messages` ファイルから得られるすべての情報を収集して、購入先に問い合わせください。購入先に問い合わせるときに必要な問題解決のための情報の完全なリストについては、[377 ページ](#)の「システムクラッシュの問題の解決」を参照してください。

システムのクラッシュ後にリポートが失敗する場合は、[第 29 章](#)を参照してください。

2. 次のように入力してディスクとの同期をとり、リポートします。

```
ok sync
```

システムのクラッシュ後にリポートが失敗する場合は、[第 29 章](#)を参照してください。

また、システムクラッシュダンプがシステムのクラッシュ後に生成されたかどうかを確認してください。デフォルトでは、システムクラッシュダンプが保存されます。クラッシュダンプの詳細については、[第 28 章](#)を参照してください。

問題の解決に使用するデータの収集

システムの問題を特定するために、次の質問に答えてください。クラッシュしたシステムの問題を解決するためのデータを収集するには、[379 ページ](#)の「システムクラッシュを解決するためのチェックリスト」を参照してください。

表 26-1 システムクラッシュに関するデータの収集

質問	説明
問題を再現できるか	この質問は、再現可能なテストケースは実際のハードウェア問題をデバッグするために重要であることが多いために重要である。購入先では、特殊な計測機構を使用してカーネルを構築して問題を再現し、バグを引き起こし、診断、および修正できる

表 26-1 システムクラッシュに関するデータの収集 (続き)

質問	説明
Sun 以外のドライバを使用しているか	ドライバは、カーネルと同じアドレス空間で、カーネルと同じ特権で動作する。したがって、ドライバにバグがあると、システムクラッシュの原因となることがある
クラッシュの直前にシステムは何を 実行していたか	システムが通常でないこと(新しい負荷テストの実行など)を行なったり、通常よりも高い負荷がシステムにかかったりした場合、クラッシュの原因となることがある
クラッシュ直前に、異常なコンソール メッセージが表示されたか	システムがクラッシュする前には、なんらかの兆候を示していることがある。この情報は重要
/etc/system ファイルに調整パラ メータを追加したか	調整パラメータは、システムクラッシュの原因となることがある。たとえば、共有メモリーセグメントを増やした結果、システムが限度以上の多くのメモリーを割り当てようとした
問題は最近発生するようになったか	そうであれば、問題の原因は、システムの変更(たとえば、新しいドライバ、新しいソフトウェア、作業負荷の変化、CPU のアップグレード、メモリーのアップグレードなど)にある可能性がある

システムクラッシュを解決するための チェックリスト

クラッシュしたシステムの問題を解決するためのデータを収集するときは、次のチェックリストを使用します。

項目	ユーザーのデータ
システムクラッシュダンプがあるか	
オペレーティングシステムのリリースと適切なソフトウェアアプリケーションのリリースレベルを確認する	
システムのハードウェアを確認する	
sun4u システムの prtdiag 出力を含める 他 のシステムの Explorer 出力を含める	

項目	ユーザーのデータ
パッチはインストールされているか。そうであれば、 <code>showrev -p</code> 出力を含める	
問題を再現できるか	
Sun 以外のドライバをシステムで使用しているか	
クラッシュ直前のシステムの動作は	
クラッシュ直前に、異常なコンソールメッセージが表示されたか	
<code>/etc/system</code> ファイルにパラメータを追加したか	
問題は最近発生するようになったか	

システムメッセージの表示

システムのメッセージはコンソールデバイスに表示されます。ほとんどのシステムメッセージは次の形式で表示されます。

[ID *msgid facility.priority*]

次に例を示します。

```
[ID 672855 kern.notice] syncing file systems...
```

カーネルから出されるメッセージには、カーネルモジュール名が次のように表示されます。次に例を示します。

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

システムがクラッシュすると、システムのコンソールに次のようなメッセージが表示されることがあります。

```
panic: error message
```

パニックメッセージより頻度は少ないですが、パニックメッセージではなく次のメッセージが表示されることがあります。

```
Watchdog reset !
```

エラー記録デーモン `syslogd` は、自動的に様々なシステムの警告やエラーをメッセージファイルに記録します。デフォルトでは、これらのシステムメッセージの多くは、システムコンソールに表示されて、`/var/adm` ディレクトリに格納されます。システムメッセージ記録を設定することによって、これらのメッセージを格納す

る場所を指示できます。詳細については、383 ページの「システムのメッセージ記録をカスタマイズする方法」を参照してください。これらのメッセージは、失敗の予兆のあるデバイスなど、システム障害をユーザーに警告できます。

`/var/adm` ディレクトリには、いくつかのメッセージファイルが含まれています。最も新しいメッセージは、`/var/adm/messages` (および `messages.*`) にあり、最も古いメッセージは、`messages.3` にあります。一定の期間 (通常は 10 日) ごとに、新しい `messages` ファイルが作成されます。 `messages.0` のファイル名は `messages.1` に、`messages.1` は `messages.2` に、`messages.2` は `messages.3` にそれぞれ変更されます。その時点の `/var/adm/messages.3` は削除されます。

`/var/adm` ディレクトリは、メッセージやクラッシュダンプなどのデータを含んでいる大きなファイルを格納するため、多くのディスク容量を消費します。`/var/adm` ディレクトリが大きくなるようにするために、そして将来のクラッシュダンプが保存できるようにするために、不要なファイルを定期的に削除しなければなりません。 `crontab` ファイルを使用すれば、この作業は自動化できます。この作業の自動化の詳細については、250 ページの「クラッシュダンプファイルを削除する方法」と第 18 章を参照してください。

▼ システムメッセージを表示する方法

システムクラッシュまたはリブートによって生成された最近のメッセージを表示するには、`dmesg` コマンドを使用します。

```
$ dmesg
```

あるいは、`more` コマンドを使用して、メッセージを 1 画面ごとに表示します。

```
$ more /var/adm/messages
```

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

例 — システムメッセージを表示する

次の例は、`dmesg` コマンドからの出力を示しています。

```
$ dmesg
Jan  3 08:44:41 starbug genunix: [ID 540533 kern.notice] SunOS Release 5.9 ...
Jan  3 08:44:41 starbug genunix: [ID 913631 kern.notice] Copyright 1983-2003 ...
Jan  3 08:44:41 starbug genunix: [ID 678236 kern.info] Ethernet address ...
Jan  3 08:44:41 starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
Jan  3 08:44:41 starbug unix: [ID 930857 kern.info] avail mem = 121888768
Jan  3 08:44:41 starbug rootnex: [ID 466748 kern.info] root nexus = Sun Ultra 5/
10 UPA/PCI (UltraSPARC-IIi 333MHz)
Jan  3 08:44:41 starbug rootnex: [ID 349649 kern.info] pcipsy0 at root: UPA 0x1f0x0
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] pcipsy0 is /pci@1f,0
Jan  3 08:44:41 starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1,1, simba0
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] simba0 is /pci@1f,0/pci@1,1
Jan  3 08:44:41 starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1, simba1
```

```
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] simba1 is /pci@1f,0/pci@1
Jan  3 08:44:57 starbug simba: [ID 370704 kern.info] PCI-device: ide@3, uata0
Jan  3 08:44:57 starbug genunix: [ID 936769 kern.info] uata0 is /pci@1f,0/pci@1,
1/ide@3
Jan  3 08:44:57 starbug uata: [ID 114370 kern.info] dad0 at pci1095,6460
.
.
.
```

システムのメッセージ記録のカスタマイズ

/etc/syslog.conf ファイルを変更すると、様々なシステムプロセスが生成するエラーメッセージを記録できます。デフォルトでは、/etc/syslog.conf は、多くのシステムプロセスのメッセージが /var/adm/messages ファイルに格納されるように指示します。クラッシュとブートのメッセージも、同様にこのファイルに格納されます。/var/adm メッセージを表示する方法については、[381 ページの「システムメッセージを表示する方法」](#)を参照してください。

/etc/syslog.conf ファイルは、タブで区切られた 2 つの列から構成されています。

facility.level ... action

facility.level

機能またはメッセージや状態のシステムでの出所。コンマで区切られた機能のリスト。機能の値については [表 26-2](#) を参照。level は、記録する状態の重要度や優先順位を示す。優先レベルについては [表 26-3](#) を参照

action

動作フィールドは、メッセージが転送される場所を示す

次は、デフォルトの /etc/syslog.conf ファイルの例です。

```
user.err /dev/sysmsg
user.err /var/adm/messages
user.alert 'root, operator'
user.emerg *
```

この例は、次のユーザーメッセージが自動的に記録されることを意味します。

- ユーザーエラーはコンソールに出力され、/var/adm/messages ファイルにも記録されます。
- 早急な対応が必要なユーザーメッセージ (alert) は、root ユーザーと operator ユーザーに送信されます。

- ユーザー緊急メッセージは、各ユーザーに送信されます。

最も一般的なエラー状態の送信元を表 26-2 に示します。最も一般的な優先順位を、重要度順に表 26-3 に示します。

表 26-2 syslog.conf メッセージの機能

送信元	説明
kern	カーネル
auth	認証
daemon	すべてのデーモン
mail	メールシステム
lp	スプールシステム
user	ユーザープロセス

注 - /etc/syslog.conf ファイルで有効化できる syslog 機能の数に制限はありません。

表 26-3 syslog.conf メッセージの優先レベル

優先順位	説明
emerg	システムの緊急事態
alert	すぐに修正が必要なエラー
crit	致命的なエラー
err	その他のエラー
info	情報メッセージ
debug	デバッグ用の出力
none	この設定は出力を記録しない

▼ システムのメッセージ記録をカスタマイズする方法

1. スーパーユーザーになります。
2. /etc/syslog.conf ファイルを編集します。syslog.conf(4) のマニュアルページで説明している構文に従って、メッセージの送信元、優先順位、およびメッセージの格納場所を追加または変更します。

3. 変更を保存して編集を終了します。

例 — システムメッセージ記録をカスタマイズする

次の `/etc/syslog.conf` の `user.emerg` 機能の例は、ユーザー緊急メッセージを `root` ユーザーと個別のユーザーに送信します。

```
user.emerg                                'root, *'
```

リモートコンソールメッセージングを有効にする

次の新しいリモートコンソール機能を使うと、リモートシステムの問題を解決しやすくなります。

- `consadm` コマンドでは、補助 (またはリモート) コンソールとしてシリアルデバイスを選択できます。 `consadm` コマンドを使用して、システム管理者は 1 つまたは複数のシリアルポートを構成して、出力先が変更されたコンソールメッセージを表示したり、システムの実行レベルが変わったときに `sulogin` セッションをサポートしたりできます。この機能を使用して、モデム付きのシリアルポートにダイヤルインしてコンソールメッセージを監視し、`init` 状態の変更を表示できます (詳細については、`sulogin(1M)` のマニュアルページと以下の詳しい手順を参照)。

補助コンソールとして構成されたポートからシステムにログインすることもできますが、このポートは主に、デフォルトコンソールに表示される情報を表示する出力デバイスです。ブートスクリプトやその他のアプリケーションがデフォルトコンソールに対して読み書きを行う場合、書き込み出力はすべての補助コンソールに出力されますが、入力にはデフォルトコンソールからだけ読み込まれます (対話型ログインセッションでの `consadm` コマンドの使用方法については、386 ページの「対話型ログインセッション中に `consadm` コマンドを使用する」を参照)。

- コンソール出力はカーネルメッセージと `syslog` メッセージからなり、新しい仮想デバイス `/dev/sysmsg` に書き込まれます。さらに、`rc` スクリプト起動メッセージが `/dev/msglog` に書き込まれます。以前のリリースでは、これらのメッセージはすべて `/dev/console` に書き込まれていました。

スクリプトメッセージを補助コンソールに表示したい場合は、コンソール出力を `/dev/console` に出力しているスクリプトで出力先を `/dev/msglog` に変更する必要があります。メッセージ出力を補助デバイスに出力変更したい場合は、`/dev/console` を参照しているプログラムで `syslog()` または `strlog()` を使用するよう明示的に変更してください。

- `consadm` コマンドは、デーモンを実行して補助コンソールデバイスを監視します。補助コンソールに指定された表示デバイスがハングアップしたりキャリア信号がなくなって切り離されると、そのデバイスは補助コンソールデバイスのリストから削除され、アクティブでなくなります。1 つまたは複数の補助コンソールを有

効にしても、メッセージがデフォルトコンソールに表示されなくなるわけではありません。メッセージは引き続き `/dev/console` に表示されます。

実行レベルの変更中に補助コンソールメッセージングを使用する

実行レベルの変更中に補助コンソールメッセージングを使う場合は、次の点に注意してください。

- システムのブート時に実行する `rc` スクリプトにユーザーの入力がある場合は、補助コンソールから入力を行うことはできません。入力はデフォルトコンソールから行う必要があります。
- 実行レベルの変更中に、スーパーユーザーパスワード入力を要求するために `sulogin` プログラムが `init` によって呼び出されます。このプログラムは、デフォルトのコンソールデバイスだけでなく各補助デバイスにもスーパーユーザーパスワードの入力要求を送信するように変更されています。
- システムがシングルユーザーモードで動作し、1つまたは複数の補助コンソールが `consadm` コマンドによって有効になっていると、最初のデバイスでコンソールログインセッションが実行され、正確なスーパーユーザーパスワードを要求する `sulogin` プロンプトが表示されます。コンソールデバイスから正しいパスワードを受け取ると、`sulogin` は他のすべてのコンソールデバイスからの入力を受信できないようにします。
- コンソールの1つがシングルユーザー特権を取得すると、デフォルトコンソールとその他の補助コンソールにメッセージが出力されます。このメッセージは、どのデバイスから正しいスーパーユーザーパスワードが入力され、コンソールになったかを示します。シングルユーザーシェルが動作する補助コンソールのキャリア信号が失われると、次のどちらかのアクションが起ることがあります。
 - 補助コンソールが実行レベル1のシステムを表している場合は、システムはデフォルトの実行レベルに移行します。
 - 補助コンソールが実行レベルSのシステムを表している場合は、シェルから `init s` または `shutdown` コマンドが入力されたデバイスに「ENTER RUN LEVEL (0-6, s or S):」というメッセージが表示されます。このデバイスのキャリア信号も失われている場合は、キャリア信号を復活して正確な実行レベルを入力する必要があります。 `init` や `shutdown` コマンドを実行しても、実行レベルプロンプトが再表示されることはありません。
- シリアルポートを使用してシステムにログインしている場合には、`init` または `shutdown` コマンドを使用して別の実行レベルに移行すると、このデバイスが補助コンソールかどうかに関係なくログインセッションは失われます。この状況は、補助コンソール機能がない Solaris リリースと同じです。
- `consadm` コマンドを使って補助コンソールにするデバイスを選択すると、システムをリブートするか補助コンソールの選択を解除するまで、そのデバイスは補助コンソールとして有効です。ただし、`consadm` コマンドには、複数のシステムリブートにまたがってデバイスを補助コンソールとして使用するオプションがありま

す (以下の詳しい手順を参照)。

対話型ログインセッション中に `consadm` コマンドを使用する

シリアルポートに接続されている端末からシステムにログインしてから `consadm` コマンドを使ってこの端末にコンソールメッセージを表示して対話型ログインセッションを行う場合、次の点に注意してください。

- この端末で対話型ログインセッションを行う場合、補助コンソールがアクティブだと、コンソールメッセージは `/dev/sysmsg` デバイスまたは `/dev/msglog` デバイスに送られます。
- この端末からコマンドを発行すると、入力はデフォルトコンソール (`/dev/console`) ではなく対話型セッションに送られます。
- `init` コマンドを実行して実行レベルを変更すると、リモートコンソールソフトウェアは対話型セッションを終了し、`sulogin` プログラムを実行します。この時点では、入力はこの端末からだけ可能で、入力はコンソールデバイスから行われたかのように扱われます。そのため、385 ページの「[実行レベルの変更中に補助コンソールメッセージングを使用する](#)」の説明のとおり、`sulogin` プログラムにパスワードを入力できます。

次に、(補助) 端末から正しいパスワードを入力すると、補助コンソールは、対話型 `sulogin` セッションを実行し、デフォルトコンソールおよび競合する補助コンソールを使えなくします。つまり、その端末は実質的にシステムコンソールとして機能します。

- この端末から実行レベル 3 または別の実行レベルに変更できます。実行レベルを変更すると、すべてのコンソールデバイスで `sulogin` が再び実行されます。終了したり、システムが実行レベル 3 で起動されるように指定すると、どの補助コンソールからも入力を行えなくなります。すべての補助コンソールはコンソールメッセージを表示するだけのデバイスに戻ります。

システムが起動する際には、デフォルトのコンソールデバイスから `rc` スクリプトに情報を入力する必要があります。システムが再び起動すると `login` プログラムがシリアルポートで実行されるため、別の対話型セッションを開始できます。そのデバイスを補助コンソールに指定していれば、コンソールメッセージはその端末に引き続き出力されます。ただし、端末からの入力はすべて対話型セッションに送られます。

▼ 補助 (リモート) コンソールを有効にする方法

`consadm` デーモンは、`consadm` コマンドで補助コンソールを追加するまでポートの監視を開始しません。セキュリティ機能として、コンソールメッセージは、キャリア信号が失われるまでか、補助コンソールデバイスの選択が解除されるまでの間だけ出力変更されます。そのため、`consadm` コマンドを使うには、そのポートでキャリア信号が確立されている必要があります。

補助コンソールを有効にする方法の詳細については、`consadm(1M)` のマニュアルページを参照してください。

1. システムにスーパーユーザーとしてログインします。
2. 補助コンソールを有効にします。

```
# consadm -a devicename
```
3. 現在の接続が補助コンソールであることを確認します。

```
# consadm
```

例 — 補助 (リモート) コンソールを有効にする

```
# consadm -a /dev/term/a  
# consadm  
/dev/term/a
```

▼ 補助コンソールのリストを表示する方法

1. システムにスーパーユーザーとしてログインします。
2. 次のどちらかの手順に従います。
 - a. 補助コンソールのリストを表示します。

```
# consadm  
/dev/term/a
```

- b. 持続的補助コンソールのリストを表示します。

```
# consadm -p  
/dev/term/b
```

▼ システムリブート後も補助 (リモート) コンソールを有効にする方法

1. システムにスーパーユーザーとしてログインします。
2. 複数のシステムリブート後も補助コンソールを有効にします。

```
# consadm -a -p devicename
```

このデバイスが持続的な補助コンソールのリストに追加されます。
3. デバイスが持続的な補助コンソールのリストに追加されているか確認します。

```
# consadm
```

例 — システムリブート後も補助 (リモート) コンソールを有効にする

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ 補助 (リモート) コンソールを無効にする方法

1. システムにスーパーユーザーとしてログインします。
2. 次のどちらかの手順に従います。

- a. 補助コンソールを無効にします。

```
# consadm -d devicename
または
```

- b. 補助コンソールを無効にし、持続的な補助コンソールのリストから削除します。

```
# consadm -p -d devicename
```

3. 補助コンソールが無効になっていることを確認します。

```
# consadm
```

例 — 補助 (リモート) コンソールが無効になっていることを確認する

```
# consadm -d /dev/term/a
# consadm
```

第 27 章

コアファイルの管理 (手順)

この章では、`coreadm` コマンドを使って、コマンドファイルを管理する方法について説明します。

コアファイルの管理に関連する手順の詳細については、389 ページの「コアファイルの管理 (作業マップ)」を参照してください。

コアファイルの管理 (作業マップ)

作業	説明	参照先
1. 現在のコアダンプ構成を表示する	<code>coreadm</code> コマンドを使用して、現在のコアダンプ構成を変更する	392 ページの「現在のコアダンプ構成を表示する方法」
2. コアダンプ構成を変更する	次のいずれかの手順を実行して、コアダンプ構成を変更する	
	コアファイル名パターンを設定する	393 ページの「コアファイル名パターンを設定する方法」
	プロセス別コアファイルのパスを有効にする	393 ページの「プロセス別コアファイルパスを有効にする方法」
	グローバルのコアファイルのパスを有効にする	393 ページの「グローバルのコアファイルパスを有効にする方法」
3. コアダンプファイルを調べる	<code>proc</code> ツールを使用して、コアダンプファイルを表示する	394 ページの「コアファイルの検査」

コアファイルの管理の概要

コアファイルは、プロセスまたはアプリケーションが異常終了したときに生成されます。コアファイルは `coreadm` コマンドで管理します。

たとえば、`coreadm` コマンドを使用して、プロセスコアファイルをすべて同じシステムディレクトリに置くようにシステムを構成できます。Solaris のプロセスやデーモンが異常終了した場合に、特定のディレクトリにあるコアファイルを調べればよいため、問題の追跡が容易になります。

構成可能なコアファイルのパス

次の 2 つの構成可能な新しいコアファイルのパスは、個別に有効または無効にすることができます。

- プロセス別コアファイルのパスにはデフォルトで `core` が使用されます。このパスはデフォルトで有効になっています。プロセス別コアファイルのパスが有効になっていると、プロセスが異常終了したときに `core` ファイルが生成されます。プロセス別のパスは、親プロセスから新しいプロセスに継承されます。
プロセス別コアファイルは生成されるとプロセスの所有者によって所有され、所有者には読み取り／書き込み権が与えられます。所有者だけがこのファイルを表示できます。
- グローバルコアファイルのパスにはデフォルトで `core` が使用されます。このパスはデフォルトで無効になっています。このパスが有効になっていると、プロセス別コアファイルのパスと同じ内容のコアファイルがグローバルコアファイルのパスに追加で作成されます。
グローバルコアファイルは生成されるとスーパーユーザーによって所有され、スーパーユーザーだけに読み取り／書き込み権が与えられます。アクセス権のないユーザーはこのファイルを表示できません。

プロセスが異常終了すると、以前の Solaris リリースと同じように `core` ファイルが現在のディレクトリに作成されます。しかし、たとえば、グローバルコアファイルのパスが有効で `/corefiles/core` に設定されていると、プロセスが終了するたびにコアファイルが 2 つ、1 つは現在の作業ディレクトリに、1 つは `/corefiles` ディレクトリにそれぞれ作成されます。

デフォルトでは、`setuid` プロセスは、グローバルの設定やプロセス別のパスを使ってコアファイルを生成することはありません。

拡張されたコアファイル名

グローバルコアファイルディレクトリが有効な場合、次の表に示す変数を使って core ファイルを相互に区別できます。

変数名	変数の定義
%p	プロセス ID
%u	実効ユーザー ID
%g	実効グループ ID
%f	実行可能ファイル名
%n	システムノード名。uname -n の出力と同じ
%m	マシン名。uname -m での出力と同じ
%t	time(2) システム呼び出しの 10 進数
%%	リテラル %

たとえば、グローバルコアファイルパスが次のように設定されている場合、

```
/var/core/core.%.f.%.p
```

PID 12345 の sendmail プロセスが異常終了すると、次の core ファイルが作成されます。

```
/var/core/core.sendmail.12345
```

コアファイル名パターンの設定

コアファイル名パターンは、グローバルに設定したりプロセス単位で設定したりできます。さらに、この設定をシステムリブート後も有効になるように保存するかどうかを指定できます。

たとえば、次の coreadm コマンドでは、init プロセスで起動されたすべてのプロセスに対しグローバルのコアファイルパターンを設定します。このパターンは複数のシステムリブート後も有効です。

```
$ coreadm -i /var/core/core.%.f.%.p
```

グローバルのコア値は /etc/coreadm.conf ファイルに格納されるため、この設定値はシステムリブート後も有効になるように保存されます。

次の coreadm コマンドでは、すべてのプロセスに対しプロセス別コアファイル名パターンを設定します。

```
$ coreadm -p /var/core/core.%.f.%.p $$
```

\$\$ 記号には、現在実行中のシェルのプロセス ID を指定します。プロセス別コアファイル名パターンは、すべての子プロセスに継承されます。

グローバルまたはプロセス別のコアファイル名パターンを設定したら、これを `coreadm -e` コマンドで有効にする必要があります。詳細については次の手順を参照してください。

このコマンドをユーザーの `$HOME/.profile` または `.login` ファイルに入れておけば、ユーザーのログインセッションで実行するすべてのプロセスに対しコアファイル名パターンを設定できます。

setuid プログラムがコアファイルを作成できるようにする

`coreadm` コマンドを使って `setuid` プログラムを有効または無効にすれば、次の設定を行うことによって、すべてのシステムプロセスに対して、または各プロセスに対してコアファイルを作成できます。

- グローバル `setuid` オプションが有効になっていると、グローバルコアファイルパスに従って、システムのすべての `setuid` プログラムが `core` ファイルを作成します。
- プロセス別 `setuid` オプションが有効になっていると、プロセス別コアファイルパスに従って、特定の `setuid` プロセスが `core` ファイルを作成します。

デフォルトでは、両方のフラグが無効になっています。セキュリティ上の理由により、グローバルコアファイルパスは、`/` で始まるフルパス名であることが必要です。スーパーユーザーがプロセス別コアファイルは無効にすると、個別のユーザーがコアファイルを得ることはできなくなります。

`setuid` コアファイルはスーパーユーザーによって所有され、スーパーユーザーだけに読み取り/書き込み権が与えられます。通常ユーザーは、たとえ `setuid` コアファイルを生成したプロセスを所有していても、それらのファイルにアクセスできません。

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

現在のコアダンプ構成を表示する方法

現在のコアダンプ構成を表示するには、オプションを指定しないで `coreadm` コマンドを実行します。

```
$ coreadm
      global core file pattern: /var/core/core.%f.%p
      init core file pattern: core
      global core dumps: enabled
      per-process core dumps: enabled
```



```
global setid core dumps: enabled
per-process setid core dumps: disabled
global core dump logging: disabled
```

コアファイル名パターンを設定する方法

1. プロセス別コアファイルを設定するのか、グローバルコアファイルを設定するのかを決めて、次のどちらかの手順に従います。

- a. プロセス別コアファイル名パターンを設定します。

```
$ coreadm -p $HOME/corefiles/%f.%p $$
```

- b. グローバルコアファイル名パターンを設定します。
まずスーパーユーザーになる必要があります。

```
# coreadm -g /var/corefiles/%f.%p
```

▼ プロセス別コアファイルパスを有効にする方法

1. スーパーユーザーになります。
2. プロセス別コアファイルのパスを有効にする
3. 現在のプロセスのコアファイルパスを表示して構成を確認します。

```
$ coreadm $$
1180: /home/kryten/corefiles/%f.%p
```

▼ グローバルのコアファイルパスを有効にする方法

1. スーパーユーザーになります。
2. グローバルのコアファイルのパスを有効にする
3. 現在のプロセスのコアファイルパスを表示して構成を確認します。

```
# coreadm
global core file pattern: /var/core/core.%f.%p
init core file pattern: core
global core dumps: enabled
per-process core dumps: enabled
global setid core dumps: disabled
```

```
per-process setid core dumps: disabled
global core dump logging: disabled
```

コアファイルの問題解決

エラーメッセージ

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

エラーの発生原因

setuid コアファイルを許容する古いパラメータが /etc/system ファイルにあります。

解決方法

/etc/system ファイルから allow_setid_core=1 を削除し、coreadm コマンドを使ってグローバル setuid コアファイルの設定を有効にします。

コアファイルの検査

一部の proc ツールが拡張されてプロセスのコアファイルやライブプロセスが調べられるようになりました。proc ツールは、/proc ファイルシステムの機能を操作するユーティリティです。

現在、コアファイルを処理できるツールは /usr/proc/bin ディレクトリにある pstack、pmap、pldd、pflags、pcred です。これらのツールを使用するには、プロセス ID を指定するように、コアファイルの名前をコマンド行に指定します。

proc ツールを使用してコアファイルを調べる方法については、proc(1) のマニュアルページを参照してください。

例 — proc ツールを使用したコアファイルの検査

```
$ ./a.out
Segmentation Fault (coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
000108c4 main      (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
00010880 _start    (0, 0, 0, 0, 0, 0) + b8
```

第 28 章

システムクラッシュ情報の管理 (手順)

この章では、Solaris 環境でシステムクラッシュ情報を管理する方法を説明します。

システムクラッシュ情報の管理に関連する手順については、395 ページの「システムクラッシュ情報 (作業マップ)」を参照してください。

システムクラッシュ情報 (作業マップ)

次の作業マップは、システムクラッシュ情報の管理に必要な手順を示します。

作業	説明	参照先
1. 現在のクラッシュダンプ構成を表示する	dumpadm コマンドを使用して、現在のクラッシュダンプ構成を表示する	399 ページの「現在のクラッシュダンプ構成を表示する方法」
2. クラッシュダンプ構成を変更する	dumpadm コマンドを使用して、ダンプするデータの種類、システムが専用のダンプデバイスを使用するかどうか、クラッシュダンプファイルを保存するディレクトリ、およびクラッシュダンプファイルが書き込まれた後に残っていなければならない容量を指定する	399 ページの「クラッシュダンプ構成を変更する方法」
3. クラッシュダンプファイルを調べる	mdb コマンドを使用して、クラッシュダンプファイルを表示する	401 ページの「クラッシュダンプを検査する方法」

作業	説明	参照先
4. (省略可能) クラッシュダンプディレクトリが一杯になった場合に復元する	システムがクラッシュした際にメモリーイメージを格納する十分な空き容量が <code>savecore</code> ディレクトリになくても、一部の重要なシステムクラッシュダンプ情報を保存したい場合	402 ページの「クラッシュダンプディレクトリが一杯になった場合に復元する方法 (省略可能)」
5. (省略可能) クラッシュダンプファイルの保存を有効または無効にする	<code>dumpadm</code> コマンドを使用して、クラッシュダンプファイルの保存を有効または無効にする。デフォルトでは、クラッシュダンプファイルは保存される	402 ページの「クラッシュダンプの保存を無効または有効にする方法」

システムクラッシュ (概要)

ハードウェアの障害、入出力の問題、ソフトウェアエラーなどが原因でシステムがクラッシュすることがあります。システムがクラッシュすると、システムはエラーメッセージをコンソールに表示し、物理メモリーのコピーをダンプデバイスに書き込みます。その後、システムは自動的にリポートします。システムがリポートすると、`savecore` コマンドが実行され、ダンプデバイスのデータを取り出して保存されたクラッシュダンプを `savecore` ディレクトリに書き込みます。このクラッシュダンプファイルは、サポートプロバイダにとって、問題を診断する上で貴重な情報となります。

システムクラッシュダンプファイル

システムクラッシュの後で自動的に実行される `savecore` コマンドは、ダンプデバイスからクラッシュダンプ情報を取り出し、`unix.X` と `vmcore.X` という 1 対のファイルを作成します。X はダンプの通し番号です。これらのファイルは 2 つで、保存されたシステムクラッシュダンプの情報を表します。

クラッシュダンプファイルはコアファイルと混同されることがあります。コアファイルは、アプリケーションが異常終了したときに書き込まれるユーザーアプリケーションのイメージです。

クラッシュダンプファイルは、あらかじめ決められたディレクトリに保存されます。これはデフォルトでは `/var/crash/hostname` です。以前の Solaris リリースでは、システムを手動で有効にして物理メモリーのイメージをクラッシュダンプファイルに保存しない限り、システムがリポートされた時にクラッシュダンプファイルが上書きされていました。このリリースでは、クラッシュダンプファイルの保存がデフォルトで有効です。

システムクラッシュ情報は `dumpadm` コマンドで管理します。詳細は、[397 ページの「dumpadm コマンド」](#)を参照してください。

クラッシュダンプの保存

制御構造体、アクティブなテーブル、動作中またはクラッシュしたシステムカーネルのメモリのイメージなど、カーネルの動作状況についての情報を調べるには、mdb ユーティリティを使用します。mdb を完全に使いこなすには、カーネルについての詳細な知識が必要ですが、このマニュアルでは説明を省きます。このユーティリティについては、mdb (1) のマニュアルページを参照してください。

savecore で保存したクラッシュダンプを購入先に送って、システムがクラッシュした原因を解析してもらうことも可能です。

dumpadm コマンド

dumpadm コマンドを使用して、Solaris 環境でシステムクラッシュダンプ情報を管理します。

- オペレーティングシステムのクラッシュダンプを構成することもできます。dumpadm 構成パラメータでは、ダンプ内容、ダンプデバイス、クラッシュダンプファイルが保存されるディレクトリなどを指定します。
- ダンプデータは、圧縮した形式でダンプデバイスに格納されます。カーネルのクラッシュダンプイメージは 4G バイトを超える場合があります。データを圧縮することにより、ダンプが速くなり、ダンプデバイスのディスク領域も少なくて済みます。
- スワップ領域ではなく、専用のダンプデバイスがダンプ構成の一部にあると、クラッシュダンプファイルの保存はバックグラウンドで行われます。つまり、システムを起動する際、savecore コマンドが完了するのを待たなくても、次の段階に進むことができます。大容量のメモリーを搭載したシステムでは、savecore コマンドが完了する前にシステムが使用可能になります。
- savecore コマンドで生成されるシステムクラッシュダンプファイルは、デフォルトで保存されます。
- savecore -L コマンドは、動作中の Solaris オペレーティング環境でクラッシュダンプを取得できる新しい機能です。たとえば、パフォーマンスに問題が発生しているときやサービスが停止しているときなどにメモリーのスナップショットをとって、実行中のシステムの問題を解決するのに使用します。システムが実行中で、一部のコマンドがまだ使用できる場合は、savecore -L コマンドを使用してシステムのスナップショットをダンプデバイスに保存し、クラッシュダンプファイルをただちに savecore ディレクトリに書き込むことができます。システムが実行中であるため、専用のダンプデバイスを構成してある場合のみ、savecore -L コマンドを使用できます。

次の表で dumpadm の構成パラメータを説明します。

ダンプパラメータ	説明
ダンプデバイス	システムがクラッシュしたときにダンプデータを一時的に保存するデバイス。ダンプデバイスがスワップ領域でない場合は、savecore がバックグラウンドで実行されるため、ブートプロセスの速度が上がる
savecore ディレクトリ	システムのクラッシュダンプファイルを保存するディレクトリ
ダンプ内容	ダンプするメモリーデータの種類
最小空き容量	クラッシュダンプファイルを保存した後で savecore ディレクトリに必要な最小空き容量。空き容量を指定しないと、デフォルトで 1M バイトになる

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

dumpadm コマンドで管理するダンプ構成パラメータは、/etc/dumpadm.conf ファイルに保存されます。

注 - /etc/dumpadm.conf ファイルは、手作業で編集しないでください。このファイルを手作業で編集すると、システムダンプ構成の整合性が失われる恐れがあります。

dumpadm コマンドの動作

dumpadm コマンドは、システム起動時に /etc/init.d/savecore スクリプトによって呼び出され、/etc/dumpadm.conf ファイルの情報に基づいてクラッシュダンプパラメータの構成を行います。

dumpadm コマンドは、/dev/dump インタフェースを通してダンプデバイスとダンプ内容を初期化します。

ダンプ構成が完了すると、savecore スクリプトは、/etc/dumpadm.conf ファイルの内容を解析してクラッシュダンプファイルのディレクトリの場所を探します。次に、savecore を呼び出して、クラッシュダンプがあるかどうかを調べたり、クラッシュダンプディレクトリにある minfree ファイルの内容を確認したりします。

ダンプデバイスとボリュームマネージャ

可用性とパフォーマンス上の理由のため、Solaris ボリュームマネージャで管理されている専用ダンプデバイスを構成しないでください。スワップ領域を Solaris ボリュームマネージャの管理下に置くことはできますが (この方法を推奨します)、ダンプデバイスは別に確保してください。

システムクラッシュダンプ情報の管理

システムクラッシュ情報を処理する場合には、次の点に注意してください。

- システムクラッシュ情報にアクセスして管理するには、スーパーユーザーでログインする必要があります。
- システムクラッシュダンプを保存するオプションを無効にしないでください。システムクラッシュファイルにより、システムクラッシュの原因を判断する非常に有効な方法が提供されます。
- また、重要なシステムクラッシュ情報は、カスタマサービス担当者へ送信するまでは削除しないでください。

▼ 現在のクラッシュダンプ構成を表示する方法

1. スーパーユーザーになります。
2. 現在のクラッシュダンプ構成を表示します。

```
# dumpadm          Dump content: kernel pages
                   Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/venus
                   Savecore enabled: yes
```

上記の出力例の意味は次のとおりです。

- ダンプの内容は、カーネルメモリーページである
- カーネルメモリーがスワップデバイス /dev/dsk/c0t3d0s1 にダンプされる。 swap -l コマンドにより、すべてのスワップ領域を識別できる
- システムクラッシュダンプファイルは /var/crash/venus ディレクトリに保存される
- システムクラッシュダンプファイルの保存は有効に設定されている

▼ クラッシュダンプ構成を変更する方法

1. スーパーユーザーになります。
2. 現在のクラッシュダンプ構成を確認します。

```
# dumpadm
                   Dump content: kernel pages
                   Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
                   Savecore enabled: yes
```

上記の出力は、Solaris 9 リリースを実行するシステムのデフォルトダンプ構成を示しています。

3. クラッシュダンプ構成を変更します。

```
# dumpadm -c content -d dump-device -m nnnk | nnnm | nnn% -n -s savecore-dir
```

-c content	ダンプするデータの種類を指定する。すべてのカーネルメモリーをダンプするには kernel を、すべてのメモリーをダンプするには all を、カーネルメモリーとクラッシュ時に実行中だったスレッドを持つプロセスのメモリーページとをダンプするには curproc を使用する。デフォルトはカーネルメモリー
-d dump-device	システムがクラッシュしたときに、ダンプデータを一時的に保存するデバイスを指定する。デフォルトのダンプデバイスは 1 次スワップデバイス
-m nnnk nnnm nnn%	現在の savecore ディレクトリに minfree ファイルを作成することにより、クラッシュダンプファイルを保存する最小限の空き容量を指定する。このパラメータは K バイト (nnnk)、M バイト (nnnm)、またはファイルシステムサイズのパーセント (nnn%) で指定できる。savecore コマンドは、クラッシュダンプファイルを書き込む前にこのファイルを調べる。クラッシュダンプファイルを書き込むと空き容量が minfree の値より少なくなる場合、ダンプファイルは書き込まれず、エラーメッセージが記録される。このような問題を解決するには、402 ページの「クラッシュダンプディレクトリが一杯になった場合に復元する方法 (省略可能)」を参照
-n	システムがリブートするときに、savecore を実行しないように指定する。このダンプ構成は推奨できない。システムクラッシュ情報がスワップデバイスに書き込まれているときに、savecore が実行されないと、クラッシュダンプ情報はシステムがスワップを開始すると上書きされる
-s	クラッシュダンプファイルを保存する別のディレクトリを指定する。デフォルトのディレクトリは /var/crash/hostname で、hostname は uname -n コマンドの出力

例 —クラッシュダンプ構成を変更する

次の例は、すべてのメモリーを専用のダンプデバイス /dev/dsk/c0t1d0s1 にダンプします。また、クラッシュダンプファイルを保存した後に残っていなければならない最小空き容量は、ファイルシステム容量の 10% です。

```
# dumpadm
  Dump content: kernel pages
  Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
# dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
  Dump content: all pages
```



```
Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
Savecore enabled: yes
```

▼ クラッシュダンプを検査する方法

1. スーパーユーザーになります。
2. `mdb` ユーティリティを使用して、クラッシュダンプを検査します。

```
# /usr/bin/mdb [-k] crashdump-file
```

`-k`

オペレーティングシステムのクラッシュダンプファイルの場合のカーネルデバッグモードを指定します。

`crashdump-file`

オペレーティングシステムのクラッシュダンプファイルを指定します。

3. クラッシュ状態情報を表示します。

```
# /usr/bin/mdb file-name
> ::status
.
.
.
> ::system
.
.
.
```

例 — クラッシュダンプを検査する

次の例は、`mdb` ユーティリティからのサンプル出力を示します。このシステムのシステム情報と `/etc/system` ファイルに設定される調整可能パラメータが表示されます。

```
# /usr/bin/mdb -k unix.0
Loading modules: [ unix krtld genunix ip nfs ipc ptm ]
> ::status
debugging crash dump /dev/mem (64-bit) from ozlo
operating system: 5.9 Generic (sun4u)
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsz=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
```

▼ クラッシュダンプディレクトリが一杯になった場合に復元する方法 (省略可能)

ここでは、システムがクラッシュしてもメモリーイメージを格納する十分な空き容量が `savecore` ディレクトリにないが、それでも、一部の重要なシステムクラッシュダンプ情報を保存したい場合を考えます。

1. システムがリブートした後で、スーパーユーザーとしてログインします。
2. すでにサービスプロバイダに送ってある既存のクラッシュダンプファイルを削除して、`savecore` ディレクトリ (通常は `/var/crash/hostname`) を整理します。あるいは、`savecore` コマンドを実行し、十分な容量を持つ別のディレクトリを指定します。次の手順を参照してください。
3. 手作業で `savecore` コマンドを実行し、必要なら別の `savecore` ディレクトリを指定します。

```
# savecore [ directory ]
```

▼ クラッシュダンプの保存を無効または有効にする方法

1. スーパーユーザーになります。
2. システム上のクラッシュダンプの保存を有効または無効にします。

```
# dumpadm -n | -y
```

例 — クラッシュダンプの保存を無効にする

次の例は、システムでのクラッシュダンプの保存を無効にします。

```
# dumpadm -n
  Dump content: all pages
  Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: no
```

例 — クラッシュダンプの保存を有効にする

次の例は、システムでのクラッシュダンプの保存を有効にします。

```
# dumpadm -y
  Dump content: all pages
  Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
```

Savecore enabled: yes

第 29 章

ソフトウェアで発生するさまざまな問題の解決 (手順)

この章では、ときどき発生するが比較的修正しやすい、さまざまなソフトウェアの問題について説明します。特定のソフトウェアアプリケーションや内容に関連しない問題 (リブートの失敗やファイルシステムがフルになるなど) の解決方法も含まれます。これらの問題の解決方法は、この後の節で説明します。

この章で説明する情報は次のとおりです。

- 405 ページの「リブートが失敗した場合の対処」
- 409 ページの「システムがハングした場合の対処」
- 410 ページの「ファイルシステムが一杯になった場合の対処」
- 411 ページの「コピーまたは復元後にファイルの ACL が消失した場合の対処」
- 412 ページの「バックアップ時の問題の解決」

リブートが失敗した場合の対処

システムがリブートに失敗した場合またはリブートしたがクラッシュした場合は、システムのブートを妨害しているソフトウェアまたはハードウェアの障害があると考えられます。

システムがブートしない原因	問題の解決方法
システムが <code>/platform/`uname -m`/kernel/unix</code> を見つけられない	SPARC システムの PROM 内の <code>boot-device</code> 設定を変更しなければならない。デフォルトのブートシステムの変更については、『Solaris のシステム管理 (基本編)』の「SPARC: システムのブート (手順)」を参照してください。

システムがブートしない原因	問題の解決方法
x86 システムで、デフォルトのブートデバイスが存在しない。「Not a UFS filesystem.」というメッセージが表示される	Configuration Assistant/Boot (構成用補助) フロッピーディスクを使用してシステムをブートし、ブートするディスクを選択する
Not a UFS filesystem. /etc/passwd ファイル内に無効なエントリが存在する	無効な passwd ファイルから復元する方法については、『Solaris のシステム管理 (基本編)』の「SPARC: システムのブート (手順)」または『Solaris のシステム管理 (基本編)』の「x86: システムのブート (手順)」を参照してください。
ディスクなどのデバイスに、ハードウェアの問題がある	ハードウェアの接続を確認する <ul style="list-style-type: none"> ■ 装置が接続されていることを確認する ■ すべてのスイッチが適切に設定されていることを確認する ■ すべてのコネクタおよびケーブル (Ethernet ケーブルも含む) を検査する ■ すべて異常がなければ、システムの電源を切り、10 秒 ~ 20 秒ほど待って、もう一度電源を投入する

上記のリストで問題が解決できない場合は、ご購入先にお問い合わせください。

SPARC: 64 ビット Solaris のブートで発生する問題の解決

64 ビット Solaris リリースを UltraSPARC システムにインストールすると、次のどの条件も該当しない場合は、64 ビットカーネルが自動的にブートされます。

- 64 ビットカーネルをブートするのに、UltraSPARC システムに FLASH PROM アップグレードが必要な場合があります。UltraSPARC システムにファームウェアのアップグレードが必要かどうかを知るには、ハードウェアメーカーの資料を参照してください。
- Open Boot PROM の boot-file パラメータに kernel/unix が設定されています。64 ビットカーネルのブートができない場合に、このパラメータがそのように設定されているなら設定を解除してシステムをリブートします。
- 64 ビット Solaris のすべての構成要素がシステムに完全にインストールされ、正しいファームウェアがインストールされていても、UltraSPARC システムによっては、デフォルトで 64 ビット Solaris カーネルがブートされない場合があります。64 ビット Solaris カーネルがブートされなければ、64 ビットアプリケーションを実行することはできません。

この問題の詳細や 64 ビット Solaris カーネルのブートをデフォルトで有効にする方法については、boot (1M) のマニュアルページを参照してください。

システムがどの Solaris カーネルを実行しているかを知るには、`isainfo -kv` コマンドが常に使用できます。

```
$ isainfo -kv
64-bit sparcv9 kernel modules
```

この例では、64 ビット Solaris カーネルが実行されています。

32 ビット Solaris システムで 64 ビット Solaris オペレーティングシステムをブートすることはできません。

ルートパスワードを忘れた場合の対処

ルートパスワードを忘れると、システムにログインできなくなります。その場合、次の手順を実行する必要があります。

- キーボードの停止シーケンスを使用して、システムを停止する。
- ブートサーバー、インストールサーバー、またはローカル CD-ROM からブートする。
- ルート (/) ファイルシステムをマウントする。
- `/etc/shadow` ファイルからルートパスワードを削除する。
- システムをリブートする。
- ログインして、ルートのパスワードを設定する。

この手順の詳細については、『Solaris のシステム管理 (基本編)』の「SPARC: システムのブート」および『Solaris のシステム管理 (基本編)』の「x86: システムのブート」を参照してください。

次の例では、SPARC システムおよび x86 システムでルートパスワードを忘れた場合の対処方法について説明します。

SPARC: 例 — ルートパスワードを忘れた場合の対処方法

次の例では、ルートパスワードを忘れた場合にネットワークからブートして回復する方法について説明します。この例では、ブートサーバーがすでに有効になっているものとします。システムのリブート後に、必ず新しいルートパスワードを適用してください。

```
(キーボードのシステム停止シーケンスを使用します — Stop-A キーを押してシステムを停止します。)  
ok boot net -s  
# mount /dev/dsk/c0t3d0s0 /a  
# cd /a/etc  
# TERM=vt100
```

```
# export TERM
# vi shadow
(ルートの暗号化パスワード文字列を削除します。)
# cd /
# umount /a
# init 6
```

x86: 例 — ルートパスワードを忘れた場合の対処方法

次の例では、ルートパスワードを忘れた場合に、ネットワークからブートして回復する方法について説明します。この例では、ブートサーバーがすでに有効になっているものとします。システムのリブート後に、必ず新しいルートパスワードを適用してください。

*(Use keyboard abort sequence to stop the system
Type any key to continue*

SunOS Secondary Boot version 3.00

Solaris Intel Platform Edition Booting System

Running Configuration Assistant...

Autobooting from Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

.
.
.

Boot Solaris

Select one of the identified devices to boot the Solaris kernel and choose Continue.

To perform optional features, such as modifying the autoboot and property settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

```
[X] NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ] DISK: (*) Target 0, QUANTUM FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
```



```

[ ] DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] CD : Target 1:TOSHIBA CD-ROM XM-5602B 1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1

F2_Continue F3_Back F4_Boot Tasks F6_Help
.
.
.
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.

Select type of installation: b -s
.
.
.
# mount /dev/dsk/c0t0d0s0 /a
.
.
.
# cd /a/etc
# vi shadow
(ルートの暗号化パスワード文字列を削除します。)
# cd /
# umount /a
# init 6

```

システムがハングした場合の対処

ソフトウェアプロセスに問題がある場合、システムは完全にクラッシュせずに凍結、つまりハングすることがあります。ハングしたシステムから回復するには、次の手順に従ってください。

1. システムがウィンドウ環境を実行していたかどうかを調べて、次の推奨事項に従ってください。これらのリストで問題が解決できなかった場合は、手順2に進みます。

- コマンドを入力しているウィンドウの中に、ポインタがあることを確認します。
 - 間違っ て Control-s キー (画面を凍結する) を押した場合は、Control-q キーを押します。Control-s キーはウィンドウだけを凍結し、画面全体は凍結しません。ウィンドウが凍結している場合は、他のウィンドウを試します。
 - 可能であれば、ネットワーク上の他のシステムからリモートでログインします。pgrep コマンドを使用して、ハングしているプロセスを見つけます。ウィンドウシステムがハングしている場合は、そのプロセスを特定して強制終了します。
2. Control-\ キーを押して、動作しているプログラムを強制終了します。core ファイルが書き出されることがあります。
 3. Control-c キーを押して、動作している可能性があるプログラムに割り込みをかけます。
 4. リモートからログインして、システムをハングさせているプロセスを特定して強制終了します。
 5. リモートからログインしてスーパーユーザーになり、システムをリブートします。
 6. システムがまだ応答しない場合は、強制的にクラッシュダンプしてリブートします。強制的にクラッシュダンプしてブートする方法については、『Solaris のシステム管理 (基本編)』の「SPARC: クラッシュダンプを強制してシステムをリブートする」または『Solaris のシステム管理 (基本編)』の「x86: クラッシュダンプを強制してシステムをリブートする」を参照してください。
 7. システムがまだ応答しない場合は、電源を切ってから数分待ち、もう一度電源を入れます。
 8. システムがまったく応答しない場合は、ご購入先にお問い合わせください。

ファイルシステムが一杯になった場合の対処

ルート (/) ファイルシステムや他のファイルシステムが一杯になると、次のようなメッセージがコンソールウィンドウに表示されます。

```
.... file system full
```

ファイルシステムが一杯になる原因はいくつかあります。次の節では、一杯になったファイルシステムを回復する方法をいくつか説明します。ファイルシステムが一杯にならないように、古い使用されていないファイルを日常的に整理する方法については、第 16 章を参照してください。

大規模ファイルまたはディレクトリを作成したために、ファイルシステムが一杯になる

エラーの原因	問題の解決方法
ファイルかディレクトリを間違った場所にコピーした。これは、アプリケーションがクラッシュして、大きな core ファイルをファイルシステムに書き込んだときにも発生する	スーパーユーザーとしてログインし、特定のファイルシステムで <code>ls -tl</code> コマンドを使用し、新しく作成された大きなファイルを特定して削除する。core ファイルを削除する方法については、 250 ページの「core ファイルを見つけて削除する方法」 を参照してください。

システムのメモリーが不足したために、tmpfs ファイルシステムが一杯になる

エラーの原因	問題の解決方法
これは、tmpfs に許可されているよりも多く書き込もうとした、または現在のプロセスがメモリーを多く使用している場合に発生する	tmpfs に関連するエラーメッセージから回復する方法については、 tmpfs (7FS) のマニュアルページ を参照してください。

コピーまたは復元後にファイルの ACL が消失した場合の対処

エラーの原因	問題の解決方法
ACL を持つファイルまたはディレクトリを /tmp ディレクトリにコピーすると、ACL 属性が消失する。/tmp ディレクトリは、通常、一時ファイルシステムとしてマウントされ、ACL などの UFS ファイルシステム属性はサポートしない	代わりに、 <code>/var/tmp</code> ディレクトリにファイルをコピーまたは復元する

バックアップ時の問題の解決

この節では、データをバックアップまたは復元するときのいくつかの基本的な問題の解決方法について説明します。

ファイルシステムのバックアップ中に、ルート (/) ファイルシステムが一杯になる

ファイルシステムをバックアップしている際に、ルート (/) ファイルシステムが一杯になる場合があります。このとき、媒体には何も書き込まれず、`ufsdump` コマンドは、媒体の 2 番目のボリュームを挿入するようにプロンプトを表示します。

エラーの原因	問題の解決方法
<code>-f</code> オプションに無効な宛先デバイス名を使用した場合、 <code>ufsdump</code> コマンドはファイルシステムを /dev ディレクトリに書き込み、このファイルシステムが一杯になる。たとえば、 <code>/dev/rmt/0</code> ではなく <code>/dev/rmt/st0</code> と入力した場合、バックアップファイルはテープドライブには送信されず、 <code>/dev/rmt/st0</code> がディスクに作成される	<code>/dev</code> ディレクトリで <code>ls -tl</code> コマンドを使用して、新しく作成された異常に大きなファイルを特定して削除する

バックアップコマンドと復元コマンドが対応していることを確認する

`ufsrestore` コマンドを使用できるのは、`ufsdump` コマンドを使用してバックアップしたファイルを復元する場合だけです。`tar` コマンドを使用してバックアップした場合は、`tar` コマンドを使用して復元します。他のコマンドで書き込まれたテープを `ufsrestore` コマンドを使用して復元しようとした場合、テープが `ufsdump` フォーマットでないことを知らせるエラーメッセージが表示されます。

現在のディレクトリが間違っていないことを確認する

ファイルを復元する場合に、間違った場所に復元してしまうことがよくあります。ufsdump コマンドは、常にファイルシステムのルートからのフルパス名でファイルをコピーします。したがって ufsrestore コマンドを実行する前に、ファイルシステムのルートディレクトリに移動しなければなりません。それよりも下のディレクトリでファイルを復元すると、そのディレクトリの下に完全なファイルツリーが作成されます。

対話型コマンド

対話型コマンドを使用すると、次の例のような ufsrestore> プロンプトが表示されます。

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Thu Aug 30 10:13:46 2003
Dumped from: the epoch
Level 0 dump of /export/home on starbug:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore >
```

ufsrestore> プロンプトでは、『Solaris のシステム管理 (デバイスとファイルシステム)』の「UFS バックアップおよび復元コマンド (参照情報)」に記載されているコマンドを使用して、ファイルを検索し、復元するファイルの一覧を作成し、そして、それらのファイルを復元できます。

第 30 章

ファイルアクセスでの問題の解決 (手順)

この章では、適切でないアクセス権と検索パスに関連する問題などのファイルアクセスでの問題を解決する手順について説明します。

この章で説明する手順は次のとおりです。

- 415 ページの「検索パスに関連する問題を解決する (コマンドが見つかりません)」
- 418 ページの「ファイルアクセスの問題を解決する」
- 418 ページの「ネットワークアクセスで発生する問題の把握」

以前は使用できていたプログラム、ファイル、またはディレクトリにアクセスできないため、システム管理者に問い合わせる場合があります。このようなときは、次の 3 点を調べてください。

- ユーザーの検索パスが変更されているか、または検索パス中のディレクトリが適切な順序であるか
- ファイルまたはディレクトリに適切なアクセス権や所有権があるか
- ネットワーク経由でアクセスするシステムの構成が変更されているか

この章では、これらの 3 点を確認する方法を簡単に説明して、可能な解決策を提案します。

検索パスに関連する問題を解決する (コマンドが見つかりません)

「コマンドが見つかりません」というメッセージは、次のいずれかを示しています。

- コマンドがそのシステムに存在しない
- コマンドのディレクトリが検索パスに存在しない

検索パスの問題を解決するには、コマンドが格納されているディレクトリのパス名を知る必要があります。

間違ったバージョンのコマンドが見つかってしまうのは、同じ名前のコマンドを持つディレクトリが検索パスにある場合です。この場合、正しいディレクトリが検索パスの後ろの方にあるか、まったく存在しない可能性があります。

現在の検索パスを表示するには、`echo $PATH` コマンドを使用します。次に例を示します。

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

間違ったバージョンのコマンドを実行しているかどうかを調べるには、`which` コマンドを使用します。次に例を示します。

```
$ which acroread
/usr/doctools/bin/acroread
```

注 - `which` コマンドは、`.cshrc` ファイルの中のパス情報を調べます。`.cshrc` ファイルに `which` コマンドの認識する別名を定義している場合に、`Bourne` シェルか `Korn` シェルから `which` コマンドを実行すると、間違った結果が返される場合があります。正しい結果を得るために、`which` コマンドは `C` シェルで使用してください。`Korn` シェルの場合は、`whence` コマンドを使用します。

▼ 検索パスの問題を診断して解決する方法

1. 現在の検索パスを表示して、コマンドが入っているディレクトリがユーザーのパス内に存在しない (あるいはスペルが間違っている) ことを確認します。

```
$ echo $PATH
```

2. 次のことをチェックします。

- 検索パスは正しいか
- 検索パスは、コマンドの他のバージョンが存在する他の検索パスの前に指定されているか
- 検索パスのいずれかにコマンドが存在するか

パスを修正する必要がある場合は、手順 3 に進みます。修正する必要がある場合は、手順 4 に進みます。

3. 次の表に示すように、適切なファイルでパスを追加します。

シェル	ファイル	構文	注
Bourne と Korn	<code>\$HOME/.profile</code>	<code>PATH=\$HOME/bin:/sbin:/usr/local/bin ...</code> <code>\$ export PATH</code>	パス名はコロンで区切る
C	<code>\$HOME/.cshrc</code> または <code>\$HOME/.login</code>	<code>hostname% set path=(~bin /sbin /usr/local/bin ...)</code>	パス名は空白文字で区切る

4. 次のように、新しいパスを有効にします。

シェル	パスが指定されているファイル	パスを有効にするコマンド
Bourne と Korn	<code>.profile</code>	<code>\$./profile</code>
C	<code>.cshrc</code> <code>.login</code>	<code>hostname% source .cshrc</code> <code>hostname% source .login</code>

5. 新しいパスを確認します。

```
$ which command
```

例 — 検索パスの問題を診断および修正する

この例は、`which` コマンドを使用して、`mytool` の実行可能ファイルが検索パス中のどのディレクトリにも存在しないことを示しています。

```
venus% mytool
mytool: Command not found
venus% which mytool
no mytool in /sbin /usr/sbin /usr/bin /etc /home/ignatz/bin .
venus% echo $PATH
/sbin /usr/sbin /usr/bin /etc /home/ignatz/bin
venus% vi ~/.cshrc
(Add appropriate command directory to the search path)
venus% source .cshrc
venus% mytool
```

コマンドを見つけることができなかった場合は、マニュアルページでそのディレクトリパスを調べます。たとえば、`lpsched` コマンド (`lp` プリンタデーモン) を見つけることができなかった場合、`lpsched(1M)` のマニュアルページを調べると、そのパスが `/usr/lib/lp/lpsched` であることがわかります。

ファイルアクセスの問題を解決する

以前はアクセスできていたファイルまたはディレクトリにアクセスできない場合は、そのファイルまたはディレクトリのアクセス権または所有権が変更されていることがあります。

ファイルとグループの所有権の変更

誰かがスーパーユーザーとしてファイルを編集したために、ファイルやディレクトリの所有権が変更されていることがあります。新しいユーザーのホームディレクトリを作成するときには、必ず、そのユーザーをホームディレクトリ内のドット(.)ファイルの所有者にしてください。ユーザーをドット(.)ファイルの所有者にしなかった場合、そのユーザーは自分のホームディレクトリにファイルを作成できません。

アクセスに関する問題は、グループの所有権が変更されたとき、またはユーザーがメンバーであるグループが /etc/group データベースから削除されたときにも発生します。

アクセスに問題があるファイルのアクセス権や所有権の変更方法については、『Solaris のシステム管理 (セキュリティサービス)』の「ファイルのセキュリティの適用 (手順)」を参照してください。

ネットワークアクセスで発生する問題の把握

リモートコピーコマンド rcp を使用してネットワーク上でファイルをコピーするとき問題が発生した場合、リモートシステム上のディレクトリやファイルは、アクセス権の設定によりアクセスが制限されている可能性があります。他に考えられる問題の原因は、リモートシステムとローカルシステムがアクセスを許可するように構成されていないことです。

ネットワークアクセスで発生する問題と AutoFS 経由でシステムにアクセスする場合に発生する問題については、『Solaris のシステム管理 (資源管理とネットワークサービス)』の「NFS の障害追跡の方法」を参照してください。

第 31 章

印刷時の問題の解決 (手順)

この章では、印刷サービスの設定または管理の際に発生する可能性のある印刷上の問題を解決する方法について説明します。

この章で説明する手順は次のとおりです。

- 425 ページの「プリンタに出力されない問題を解決する方法」
- 431 ページの「出力が正しくない場合の問題を解決する方法」
- 435 ページの「LP 印刷サービスのハングを解除する方法」
- 436 ページの「アイドル状態になった (ハングした) プリンタの問題を解決する方法」
- 438 ページの「矛盾したプリンタ状態メッセージを解決する方法」

印刷と LP 印刷サービスについては、第 2 章を参照してください。

印刷時の問題解決のヒント

プリンタを設定後に、何も印刷されないことがあります。また、若干は処理されるものの、何か印刷しても正しく出力されない、読みづらいなど、期待どおりの結果が得られないことがあります。このような問題が発生すると、他にも次のような問題が発生することがあります。

- LP コマンドがハングする
- プリンタがアイドル状態になる
- ユーザーが矛盾したメッセージを受け取る

注 - この章の推奨事項の多くはパラレルプリンタに関連しますが、より一般的なシリアルプリンタにも当てはまります。

出力されない (印刷されない) 場合の対処

何も印刷されないときは、次の部分をチェックします。

- プリンタハードウェア
- ネットワーク
- LP 印刷サービス

バナーページは印刷されるのに他には何も印刷されない場合は、不正な出力の特殊ケースです。421 ページの「出力が正しくない場合の対処」を参照してください。

ハードウェアのチェック

ハードウェアは、最初にチェックすべきポイントです。プリンタが電源に接続され、電源がオンになっているかどうかを確認してください。また、ハードウェア付属のマニュアルを参照して、ハードウェアの設定値を調べてください。コンピュータによっては、プリンタポートの特性を変更するハードウェアスイッチが付いているものがあります。

プリンタハードウェアには、プリンタ、コンピュータへの接続ケーブル、ケーブルの先端を接続するポートが含まれます。一般的なアプローチとしては、プリンタからコンピュータへと順番に調べてください。まず、プリンタをチェックします。次に、ケーブルがプリンタに接続される箇所をチェックします。次に、ケーブルをチェックします。最後に、ケーブルがコンピュータに接続されている箇所をチェックします。

ネットワークのチェック

よく問題が発生するのは、印刷クライアントからプリンタサーバーに送られるリモート印刷要求です。プリンタサーバーと印刷クライアント間でネットワークアクセスが使用可能になっているかどうかを確認してください。

ネットワークがネットワーク情報サービスプラス (NIS+) を実行している場合は、システム間のアクセスを有効にする方法について、『Solaris のシステム管理 (ネーミングとディレクトリサービス: FNS, NIS+ 編)』を参照してください。ネットワークがネットワーク情報サービス (NIS) または NIS+ を実行していない場合は、プリンタサーバーと印刷クライアントを設定する前に、プリンタサーバー上の `/etc/hosts` ファイルに各クライアントシステムのインターネットアドレスとシステム名を組み込んでください。また、プリンタサーバーの IP アドレスとシステム名を、各印刷クライアントシステムの `/etc/hosts` ファイルに組み込まなければなりません。

LP 印刷サービスのチェック

正常に印刷するには、プリンタサーバーと印刷クライアント上で LP スケジューラが動作していなければなりません。動作していない場合は、`/usr/lib/lp/lpsched` コマンドを使用して起動する必要があります。スケジューラの起動に問題がある場合は、93 ページの「印刷スケジューラを再起動する方法」を参照してください。

スケジューラが動作している他に、出力する前にプリンタが使用可能になっていて、印刷要求を受け付けられる状態になっていなければなりません。LP 印刷サービスがプリンタへの要求を受け付けなければ、依頼した印刷要求は拒否されます。その場合、一般にユーザーは印刷要求を依頼すると警告メッセージを受け取ります。LP 印刷サービスがプリンタで使用可能になっていないと、印刷要求はプリンタが使用可能になるまでシステム上の待ち行列に残ります。

通常は、次の手順で印刷時の問題を分析してください。

- 手順ごとに印刷要求の経路を追跡します。
- 手順ごとに LP 印刷サービスの状態を調べます。
 - 構成は正しいか
 - プリンタは要求を受け付けるか
 - プリンタは要求を処理できるか
- 要求が転送時にハングしている場合は、`syslog.conf` 内の `lpr.debug` を設定して、転送状況を表示します。
- 要求がローカルでハングしている場合は、`lp sched` ログ (`/var/lp/logs/lpsched`) を調べます。
- 要求がローカルでハングしている場合は、プリンタデバイスエラー (障害) の通知を送らせ、プリンタを再度使用可能にします。

425 ページの「印刷時の問題の解決」に掲載されている手順では、この方法を使用して LP 印刷サービスに関する各種の問題に対処する方法を説明します。

出力が正しくない場合の対処

プリンタと印刷サービスソフトウェアが正しく構成されていない場合は、プリンタで印刷されても、期待どおりに出力されないことがあります。

プリンタタイプとファイル内容形式のチェック

LP 印刷サービスでプリンタを設定するときに間違ったプリンタタイプを使用すると、不適切なプリンタ制御文字がプリンタに送られる可能性があります。その結果は予測できません。何も印刷されない、出力が読みづらい、正しい文字セットやフォントで印刷されないなどの結果となります。

間違ったファイル内容形式を指定した場合、バナーページは印刷できますが、他には何も印刷されません。プリンタに指定されたファイル内容形式は、プリンタがフィルタなしで直接印刷できるファイル形式を示します。ユーザーがプリンタにファイルを送信すると、ファイルはフィルタなしでプリンタに直接送信されます。プリンタがその形式を処理できないときは、問題が発生します。

印刷クライアントの設定時には、ファイル内容形式がプリンタサーバーと印刷クライアントの両方で正しくなければならぬので、間違いをおかす機会が多くなります。推奨する方法は、印刷クライアントのファイル内容形式を any に設定することです。こうすると、ファイルはサーバーに直接送信され、フィルタが必要かどうかはサーバー側で決定されます。したがってファイル内容形式は、サーバー側だけで正しく指定すればよいことになります。

印刷クライアント側でファイル内容を指定し、フィルタリングの負荷をサーバーからクライアントに移すことができますが、内容の形式はプリンタサーバー側でサポートしなければなりません。

stty 設定値のチェック

デフォルトの stty (標準端末) 設定値がプリンタから要求される設定値と一致しないと、多数のフォーマット上の問題が生じる可能性があります。この後の節では、設定値の一部が間違っているときに発生する問題について説明します。

ボーレート設定値が正しくない場合

コンピュータのボーレート設定値がプリンタのボーレート設定値と一致しないときは、通常何か出力されますが、希望する出力は得られません。特殊文字や不要なスペースが異常に混じったランダムな出力が表示されます。LP 印刷サービスのデフォルトは 9600 ボーレートです。

注 - プリンタがパラレルポートで接続されている場合、ボーレートの設定値は無関係です。

パリティ設定値が正しくない場合

プリンタによっては、パリティビットを使用して、印刷用に受け取ったデータに伝送中に誤りがなかったことを確認するものがあります。コンピュータとプリンタのパリティビットの設定値は一致しなければなりません。一致しない場合、文字によってはまったく印刷されないか、他の文字で置き換えられることもあります。その出力は文字間隔が正しく、ほとんどの文字が正しい位置にあるので、一見正しいように見えます。LP 印刷サービスの場合、デフォルトではパリティビットは設定されません。

タブ設定値が正しくない場合

ファイルにタブが含まれていても、プリンタがタブを予期していなければ、印刷出力にはファイルの内容が完全に印刷されますが、テキストは右マージンに対して正確に配置されないことがあります。また、プリンタのタブ設定が間違っていると、テキストに左マージンがない、テキストがつながってしまう、テキストがページの一部分に集中する、間違っダブルスペースになってしまうなどの問題が発生します。デフォルトでは、タブは 8 スペースごとに設定されます。

Return 設定値が正しくない場合

出力がシングルスペースのはずなのにダブルスペースになる場合は、プリンタのタブ設定値が間違っているか、プリンタが Return の後に 1 行追加されています。LP 印刷サービスは、改行の前に 1 つ Return を追加するので、その組み合わせによって 2 行の改行が発生します。

ジグザグに印刷される場合は、改行の前に Return を送る stty オプションの onlcr が設定されていません。stty=onlcr オプションはデフォルトで設定されますが、他の印刷問題を解決しようとしたときに、それを消去した可能性があります。

LP コマンドがハングした場合の対処

lp コマンド (lpssystem、lpadmin、lpstat など) を入力しても何も発生しない (エラーメッセージ、状態情報、またはプロンプトが表示されない) 場合は、LP スケジューラに問題が発生した可能性があります。このような問題は、通常は LP スケジューラを停止して再起動すれば解決できます。操作手順については、[92 ページ](#)の「印刷スケジューラを停止する方法」を参照してください。

プリンタがアイドル状態になった (ハングした) 場合の対処

プリンタが印刷要求を待ち行列に入れているのに、アイドル状態になっていることがあります。プリンタがアイドル状態になっている場合は、次の原因が考えられます。

- 現在の印刷要求にフィルタがかけられている
- プリンタに障害がある
- ネットワーク上の問題が原因で、印刷処理が中断されている

印刷フィルタのチェック

低速印刷フィルタは、プリンタを拘束しないようにバックグラウンドで実行されます。フィルタリングが必要な印刷要求は、フィルタリングが終わるまで印刷されません。

プリンタ障害のチェック

LP 印刷サービスが障害を検出すると、印刷はすぐにではありませんが自動的に再開されます。LP 印刷サービスは約 5 分間待機し、要求が正常に印刷されるまで試行し続けます。プリンタを使用可能にすると、すぐに再試行できます。

ネットワーク上の問題のチェック

ネットワーク経由でファイルを印刷するときには、次の問題が発生することがあります。

- プリンタサーバーに送られた要求が、クライアントシステム (ローカル) の待ち行列で停止する
- プリンタサーバーに送られた要求が、プリンタサーバー (リモート) の待ち行列で停止する

ローカル待ち行列で停止する印刷要求

プリンタサーバーに依頼された印刷要求は、次の原因でクライアントシステムの待ち行列で停止することがあります。

- プリンタサーバーがダウンしている
- プリンタがプリンタサーバー側で使用不可にされている
- 印刷クライアントとプリンタサーバー間のネットワークがダウンしている
- ベースになる互換バージョンのネットワークソフトウェアが適切に設定されていない

問題の原因を突き止めるときには、新しい要求を待ち行列に追加しないでください。詳細は、110 ページの「プリンタへの印刷要求を受け付けるまたは拒否する方法」を参照してください。

リモート待ち行列で停止する印刷要求

印刷要求がプリンタサーバーの待ち行列で停止する場合は、プリンタが使用不可になっている可能性があります。プリンタが要求を受け付けても処理しないとき、その要求は印刷するために待ち行列に入れられます。プリンタを使用可能にすると、それ以外に問題がなければ、待ち行列内の印刷要求は印刷されます。

矛盾した状態メッセージの対処

ユーザーが印刷要求を入力すると、クライアントシステムからは受け付けられたことが通知され、プリンタサーバーからは印刷要求が拒否されたことを示すメールを受け取ることがあります。これらの矛盾したメッセージは、次の原因で発生することがあります。

- 印刷クライアントは要求を受け付けることができても、プリンタサーバーは要求を拒否している場合
- 印刷クライアント側のプリンタの定義が、プリンタサーバー側のプリンタの定義と一致しない場合。特に、フィルタ、文字セット、印字ホイール、フォームなど、印刷ジョブコンポーネントの定義が、クライアントとサーバーシステムの間で一致していない場合

ローカルユーザーがプリンタサーバー上でプリンタにアクセスできるように、これらのジョブコンポーネントの定義が印刷クライアントとプリンタサーバーの両方で登録されているかどうかを確認してください。

印刷時の問題の解決

この節では、次の手順について説明します。

- 出力されない問題を解決する方法
- 出力が正しくない問題を解決する方法
- LP コマンドのハングを解除する方法
- アイドル状態になった (ハングした) プリンタの問題を解決する方法
- 矛盾した状態メッセージを解決する方法

プリンタに出力されない問題を解決する方法

この作業には、次の問題解決の手順が含まれています。印刷要求をプリンタに出したのに何も印刷されない場合は、これらの手順を試してください。

- ハードウェアをチェックする (425 ページの「プリンタハードウェアをチェックする方法」を参照)
- ネットワークをチェックする (426 ページの「プリンタのネットワーク接続をチェックする方法」を参照)
- LP 印刷サービスの基本機能をチェックする (427 ページの「LP 印刷サービスの基本機能をチェックする方法」を参照)
- Solaris 印刷クライアントから Solaris プリンタサーバーへの印刷をチェックする (430 ページの「Solaris 印刷クライアントから Solaris プリンタサーバーへの印刷をチェックする方法」を参照)

印刷クライアント/プリンタサーバーの節に進む前に、上記のうち最初の 3 つの手順をリストの順に試してください。ただし、バナーページは印刷されるが他に何も印刷されない場合は、431 ページの「出力が正しくない場合の問題を解決する方法」の説明に進んでください。

▼ プリンタハードウェアをチェックする方法

1. プリンタがコンセントに接続され、電源がオンになっているか確認します。
2. ケーブルがプリンタのポートと、システムまたはサーバーのポートに接続されているか確認します。
3. そのケーブルが正しいケーブルであり、欠陥がないことを確認します。
詳細は、ハードウェア付属のマニュアルを参照してください。プリンタがシリアルポートに接続されている場合は、そのケーブルでハードウェアフロー制御がサポートされることを確認してください。ヌルモデムアダプタでは、この機能がサポートされます。次の表は、ヌルモデムケーブル用のピン構成を示しています。

表 31-1 スルモデムケーブル用のピン構成

	ホスト	プリンタ
Mini-Din-8	25-Pin D-sub	25-Pin D-sub
-	1(FG)	1(FG)
3(TD)	2(TD)	3(RD)
5(RD)	3(RD)	2(TD)
6(RTS)	4(RTS)	5(CTS)
2(CTS)	5(CTS)	4(RTS)
4(SG)	7(SG)	7(SG)
7(DCD)	6(DSR)、8(DCD)	20(DTR)
1(DTR)	20(DTR)	6(DSR)、8(DCD)

4. ポート用のハードウェアスイッチが正しく設定されていることを確認します。
正しい設定については、プリンタのマニュアルを参照してください。
5. プリンタが動作するか確認します。
プリンタにセルフテスト機能が付いている場合は、その機能を使用します。プリンタのセルフテストの詳細は、プリンタのマニュアルを参照してください。
6. コンピュータとプリンタのボーレートの設定値が正しいか確認します。
コンピュータとプリンタのボーレートの設定値が一致しなければ、何も印刷されないことがあり、さらに正しく出力されない場合もあります。詳細は、431 ページの「出力が正しくない場合の問題を解決する方法」を参照してください。

▼ プリンタのネットワーク接続をチェックする方法

1. **ping** コマンドを使用すると、プリンタサーバーと印刷クライアント間のネットワークが正しく設定されているか確認できます。

```
print_client# ping print_server
print_server is alive
print_server# ping print_client
print_client not available
```

システムが動作していることを示すメッセージが表示されれば、そのシステムにアクセスできることがわかるので、そのネットワークは正常です。また、このメッセージは、入力したホスト(システム) 名が、ネームサービスまたはローカルの /etc/hosts ファイルによって IP アドレスに変換されたことを示します。変換されていない場合は、IP アドレスを入力する必要があります。

「not available」というメッセージが表示された場合は、次の 3 点を確認してください。まず、NIS または NIS+ はサイトでどのように設定されているか。次

に、プリンタサーバーと印刷クライアントが相互に通信できるように付加的な作業が必要か。最後に、サイトが NIS または NIS+ を実行していない場合、各印刷クライアントの /etc/hosts ファイルにプリンタサーバーの IP アドレスを入力し、プリンタサーバーの /etc/hosts ファイルにすべての印刷クライアントの IP アドレスを入力したか確認します。

2. (SunOS 5.0 – 5.1 プリンタサーバーのみ) **listen** ポートモニターが正しく構成されているか確認します。
3. (SunOS 5.0 – 5.1 プリンタサーバーのみ) ネットワーク待機サービスがプリンタサーバー上のポートモニターに登録されているか確認します。

▼ LP 印刷サービスの基本機能をチェックする方法

この手順では、基本 LP 印刷サービス機能をチェックする例として、プリンタ luna を使用しています。

1. プリンタサーバー上で、LP 印刷サービスが動作していることを確認します。

- a. このコマンドは、LP スケジューラが動作しているか表示します。

```
# lpstat -r
scheduler is running
```

- b. スケジューラが動作していない場合は、スーパーユーザーまたは **lp** になり、スケジューラを起動します。

```
# /usr/lib/lp/lpsched
```

スケジューラを起動できない場合は、[435 ページの「LP 印刷サービスのハングを解除する方法」](#)を参照してください。

2. プリンタサーバー上と印刷クライアント上で、プリンタが要求を受け付けていることを確認します。

- a. プリンタが要求を受け付けていることを確認します。

```
# lpstat -a
mars accepting requests since Jul 12 14:23 2003
luna not accepting requests since Jul 12 14:23 2003
unknown reason
```

このコマンドは、LP システムがシステム用に構成された各プリンタの要求を受け付けているか確認します。

- b. プリンタが要求を受け付けていない場合は、スーパーユーザーまたは **lp** になり、プリンタが印刷要求を受け付けるようにします。

```
# accept luna
```

これで、指定したプリンタは要求を受け付けます。

3. プリンタサーバー上と印刷クライアント上で、プリンタが依頼された印刷要求の印刷で使用可能になっているか確認します。

- a. プリンタが使用可能になっていることを確認します。

```
# lpstat -p luna
printer luna disabled since Jul 12 14:25 2003.
available.
unknown reason
```

このコマンドは、プリンタの状態に関する情報を表示します。プリンタ名を省略すると、システム用に設定されたすべてのプリンタに関する情報を表示できます。次の例は、使用不可になっているプリンタを示しています。

- b. プリンタが使用不可になっている場合は、スーパーユーザーまたは **lp** になり、プリンタを使用可能にします。

```
# enable luna
printer "luna" now enabled.
```

指定したプリンタが、印刷要求の処理に使用可能になります。

4. プリンタサーバー上で、プリンタが正しいシリアルポートに接続されていることを確認します。

- a. プリンタが正しいシリアルポートに接続されていることを確認します。

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
```

「device for *printer-name*」というメッセージは、ポートアドレスを示します。LP 印刷サービスの接続先のポートにケーブルが接続されているか確認します。ポートが正しければ、手順 5 に進みます。

- b. スーパーユーザーまたは **lp** になります。

- c. ポートを表すデバイスファイルのファイル所有権を変更します。

```
# chown lp device-filename
```

このコマンドは、特殊なユーザー **lp** をデバイスファイルの所有者として割り当てます。このコマンドで、*device-filename* はデバイスファイル名です。

- d. プリンタポートのデバイスファイルのアクセス権を変更します。

```
# chmod 600 device-filename
```

このコマンドにより、スーパーユーザーまたは **lp** だけがプリンタポートデバイスファイルにアクセスできます。

5. プリンタサーバー上と印刷クライアント上で、プリンタが正しく構成されていることを確認します。

- a. プリンタが適切に設定されていることを確認します。

```
# lpstat -p luna -l
printer luna is idle. enabled since Jul 12 14:24 2003. available
Content types: postscript
Printer types: PS
```

上の例は、正しく設定された PostScript プリンタと、そのプリンタを印刷要求の処理に利用できることを示しています。プリンタタイプとファイル内容形式が正しい場合は、手順 6 に進みます。

- b. プリンタタイプまたはファイル内容形式が違っている場合は、印刷クライアント上で、プリンタタイプを **unknown** に設定し、内容形式を **any** に設定してください。

```
# lpadmin -p printer-name -T printer-type -I file-content-type
```

6. プリンタサーバー上で、プリンタがプリンタ障害のために待機していないことを確認します。

- a. プリンタ障害のためにプリンタが待機していないことを確認します。

```
# lpadmin -p printer-name -F continue
```

このコマンドは LP 印刷サービスに対して、障害のために待機していない場合は続行するように指示します。

- b. プリンタを再び使用可能にすることによって、すぐに再試行させます。

```
# enable printer-name
```

- c. (省略可能) プリンタ障害をすぐに通知するように、LP 印刷サービスに指示します。

```
# lpadmin -p printer-name -A 'write root'
```

このコマンドは LP 印刷サービスに対して、プリンタが障害を起こした場合に、root に書き込むというデフォルトポリシーを設定し、root がログインした端末にプリンタ障害メッセージを送るように指示します。これにより、問題を修正するときに障害通知をすぐに受け取れます。

7. プリンタがログイン端末として間違った設定になっていないか確認します。

注 - ログイン端末としてプリンタを設定する作業では誤りをおかしやすいので、当てはまらないと思われる場合にも、必ず設定値を確認してください。

- a. **ps -ef** コマンドの出力で、プリンタポートのエントリを探します。

```
# ps -ef
root    169    167    0    Apr 04 ?          0:08 /usr/lib/saf/listen tcp
root    939      1    0 19:30:47 ?          0:02 /usr/lib/lpsched
root    859    858    0 19:18:54 term/a    0:01 /bin/sh -c \ /etc/lp
/interfaces/luna
luna-294 rocket!smith "passwd\n##
#
```

このコマンドの出力で、プリンタポートのエントリを探します。上の例で、ポート /dev/term/a はログイン端末として間違っ設定されています。この行の最後に "passwd\n## 情報が付いているのでわかります。ポートが正しく設定されている場合は、この手順の最後を飛ばしてください。

- b. 印刷要求を取り消します。

```
# cancel request-id
```

このコマンドで、*request-id* は取り消したい印刷要求の要求 ID 番号です。

- c. プリンタポートをログインデバイス以外のものとして設定します。

```
# lpadmin -p printer-name -h
```

- d. **ps -ef** コマンドからの出力をチェックして、プリンタポートがログインデバイスではなくなったことを確認します。

基本的な LP 印刷サービス機能に印刷時の問題の原因が見つからない場合は、次の中から該当するクライアント/サーバーの手順に進んでください。

▼ Solaris 印刷クライアントから Solaris プリンタサーバーへの印刷をチェックする方法

1. まだチェックしていなければ、プリンタサーバー上で LP 印刷サービスの基本機能をチェックします。

基本機能をチェックする手順については、427 ページの「LP 印刷サービスの基本機能をチェックする方法」を参照してください。印刷クライアントから要求が出されたときに何も印刷されない原因を調べる前に、プリンタがローカルで動作していることを確認してください。

2. まだチェックしていなければ、印刷クライアント上で LP 印刷サービスの基本機能をチェックします。

印刷クライアントから印刷を要求する前に、そのプリンタを有効にして、その要求を受け付けるようにする必要があります。

注 - 次の手順のほとんどは、スーパーユーザーまたは lp としてログインして実行しなければなりません。

3. プリンタサーバーがアクセス可能であることを確認します。

- a. 印刷クライアント上で、応答を求める要求をプリンタサーバーに送信します。

```
# ping print_server
```

「*print_server not available*」というメッセージが表示される場合は、ネットワークに問題があります。

4. プリンタサーバーが正常に動作しているか確認します。

```
# lpstat -t luna
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Jul 12 14:26 2003
printer luna now printing luna-314. enabled since Jul 12 14:26 2003.
available.
luna-129          root                488    Jul 12 14:32
#
```

上記の例は、プリンタサーバーが動作していることを示します。

5. プリンタサーバーが正常に動作していない場合は、手順 1 に戻ります。

▼ 出力が正しくない場合の問題を解決する方法

1. スーパーユーザーまたは **lp** としてログインします。
2. プリンタタイプが正しいことを確認します。

プリンタタイプが正しくないと、正しく出力されないことがあります。たとえば、プリンタタイプ PS を指定してもページが意図した順序の逆に印刷される場合は、プリンタタイプ PSR を試してください(この 2 つのタイプ名は大文字で指定しなければなりません)。また、プリンタタイプが正しくないと、テキストの欠落、読みづらいテキスト、または間違っただフォントのテキストが出力されることがあります。プリンタタイプを判別するには、terminfo データベース内のエントリを調べます。terminfo データベースの構造については、51 ページの「プリンタタイプ」を参照してください。

- a. プリンタサーバー上で、プリンタの特性を表示します。

```
$ lpstat -p luna -l
printer luna is idle. enabled since Thu Jul 12 15:02:32 ...
  Form mounted:
  Content types: postscript
  Printer types: PS
  Description:
  Connection: direct
  Interface: /usr/lib/lp/model/standard
  After fault: continue
  Users allowed:
    (all)
  Forms allowed:
    (none)
  Banner not required
  Character sets:

  Default pitch:
  Default page size: 80 wide 66 long
  Default port settings:
$
```

- b. プリンタのマニュアルを参照して、プリンタのモデルを調べます。
- c. プリンタタイプが正しくない場合は、**Solaris** プリンタマネージャの「プリンタのプロパティを変更 (**Modify Printer Properties**)」ウィンドウを使用して変更するか、次の **lpadmin** コマンドを使用します。

```
# lpstat -p printer-name -T printer-type
```

印刷クライアント上では、プリンタタイプを unknown にしてください。プリンタサーバー上では、プリンタタイプは使用するプリンタのモデルをサポートするように定義された terminfo エントリと一致しなければなりません。使用するプリンタのタイプに関する terminfo エントリがない場合は、150 ページの「サポートされていないプリンタの terminfo エントリを追加する方法」を参照してください。

- 3. バナーページは印刷されるが文書の本文が印刷されない場合は、ファイル内容形式を確認します。

プリンタに指定したファイル内容形式は、プリンタがフィルタなしで直接印刷できるファイル形式を示します。ファイル内容形式が正しくなければ、必要なときにフィルタリングがバイパスされることがあります。

- a. 前の手順の **lpstat** コマンドで表示されたファイル内容形式に関する情報をメモします。

印刷クライアント上では、1 つ以上の明示的な内容形式を指定する理由がない限り、ファイル内容形式を any にしてください。クライアント上で内容を指定すると、プリンタサーバー上ではなく印刷クライアント上でフィルタリングが実行されます。また、クライアント上の内容形式は、プリンタサーバー上で指定した内容形式と一致しなければならず、プリンタサーバー上の内容形式はプリンタの機能を反映していなければなりません。

- b. プリンタのマニュアルを参照し、プリンタで直接印刷できるファイルのタイプを判別します。

これらのファイル形式を参照するために使用する名前は、プリンタメーカーが使用している名前と一致しなくてもかまいません。ただし、使用する名前は LP 印刷サービスに認識されるフィルタで使用する名前と一致しなければなりません。

- c. ファイル内容形式が正しくない場合は、**Solaris** プリンタマネージャの「プリンタのプロパティを変更 (**Modify Printer Properties**)」ウィンドウを使用して変更するか、次の **lpadmin** コマンドを使用します。

```
# lpadmin -p printer-name -I file-content-type(s)
```

必要に応じて、このコマンドを印刷クライアント上、プリンタサーバー上、またはその両方で実行します。印刷クライアント上で **-I any** を試し、プリンタサーバー上で **-I ""** を試してください。 **-I ""** は、NULL のファイル内容形式リストを指定します。これは、プリンタはそのプリンタタイプと正確に一致するファイルしか直接印刷できないので、すべてのファイルをフィルタにかけることを意味します。

ファイルが印刷されないときは、まずこの組み合わせを選択してみるとよいでしょう。それで成功したら、プリンタサーバー上で明示的な内容形式を指定し、不要なフィルタリングを減らすことができます。ローカルの PostScript プリンタでは、プリンタでサポートされている場合は、`postscript` または `postscript, simple` を使用してください。PS と PSR はファイル内容形式ではなく、プリンタタイプなので注意してください。

`-I` を省略すると、ファイル内容のリストはデフォルトの `simple` になります。 `-I` オプションを使用し、`simple` 以外にもファイル内容形式を指定したい場合は、リストに `simple` を含めなければなりません。

複数のファイル内容形式を指定するときは、名前をコンマで区切ります。また、名前をスペースで区切り、リストを引用符で囲むこともできます。ファイル内容形式として `any` を指定すると、フィルタリングは行われないので、プリンタで直接印刷できるファイルタイプのみを送信する必要があります。

4. フォントのダウンロードに必要なフィルタリングを、印刷要求がバイパスしていないかどうかをチェックします。

ユーザーがコマンド `lp -T PS` を使用して印刷要求を PostScript プリンタに依頼すると、フィルタリングは実行されません。フィルタリングを強制するコマンド `lp -T postscript` を使用して要求を依頼しようとする、文書に必要な非常駐フォントがダウンロードされることがあります。

5. プリンタポートの `stty` 設定値が正しいことを確認します。
 - a. プリンタのマニュアルを参照して、プリンタポートに合った `stty` 設定値を判別します。

注 - プリンタがパラレルポートまたは USB ポートで接続されている場合、ボーレートの設定値は無関係です。

- b. 現在の設定値を調べるには、`stty` コマンドを使用します。

```
# stty -a < /dev/term/a
speed 9600 baud;
rows = 0; columns = 0; ypixels = 0; xpixels = 0;
eucw 1:0:0:0, scrw 1:0:0:0
intr = ^c; quit = ^|; erase = ^?; kill = ^u;
eof = ^d; eol = <undef>; eol2 = <undef>; swtch = <undef>;
start = ^q; stop = ^s; susp = ^z; dsusp = ^y;
rprnt = ^r; flush = ^o; werase = ^w; lnext = ^v;
parenb -parodd cs7 -cstopb -hupcl cread -clocal -loblk -parext
-ignbrk brkint -ignpar -parmrk -inpck istrip -inlcr -igncr icrnl -iuclic
ixon -ixany -ixoff imaxbel
isig icanon -xcase echo echoe echok -echonl -noflsh
-tostop echoctl -echoprt echoke -defecho -flusho -pendin iexten
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel tab3
#
```

このコマンドでは、プリンタポートの現在の `stty` 設定値が表示されます。

LP 印刷サービスの標準プリンタインタフェースプログラムで使用されるデフォルトの stty オプションを表 31-2 に示します。

表 31-2 標準インタフェースプログラムで使用されるデフォルト stty 設定値

オプション	説明
-9600	ボーレートを 9600 に設定
-cs8	8 ビットバイトを設定
-cstopb	1 バイト当たり 1 ストップビットを送信
-parity	パリティを生成しない
-ixon	XON/XOFF (START/STOP または DC1/DC3 ともいう) を使用可能にする
-opost	以下にリストされた設定値をすべて使用して「処理後出力」を実行する
-olcuc	小文字を大文字に割り当てない
-onlcr	改行をキャリッジリターン/改行に変更する
-ocrnl	キャリッジリターンを改行に変更しない
-onocr	カラム 0 でもキャリッジリターンを出力する
-nl0	改行後の遅延なし
-cr0	キャリッジリターン後の遅延なし
-tab0	タブ後の遅延なし
-bs0	バックスペース後の遅延なし
-vt0	垂直タブ後の遅延なし
-ff0	用紙送り後の遅延なし

c. stty 設定値を変更します。

```
# lpadmin -p printer-name -o "stty= options"
```

表 31-3 を使用して、印刷出力に影響する様々な問題を解決する stty オプションを選択します。

表 31-3 印刷出力の問題を解決する stty オプション

stty 値	結果	間違った設定から起こり得る問題
110, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400	ボーレートを指定した値に設定する (ボーレートを 1 つだけ入力する)	ランダム文字と特殊文字が印刷され、間隔がバラバラになることがある

表 31-3 印刷出力の問題を解決する stty オプション (続き)

stty 値	結果	間違っただ設定から起こり得る問題
oddp	奇数パリティを設定する	文字が欠落または間違っただ文字がランダムに表示される
evenp	偶数パリティを設定する	
-parity	パリティを設定しない	
-tabs	タブを設定しない	テキストが右マージンにくっついてしまう
tabs	8 スペースごとにタブを設定する	テキストに左マージンがなく、つながってしまうか、くっついてしまう
-onlcr	行頭でキャリッジリターンを設定しない	間違っただダブルスペース
onlcr	行頭でキャリッジリターンを設定する	ジグザグに印刷される

オプションをスペースで区切り、オプションリストを単一引用符で囲むと、複数のオプションの設定を変更できます。たとえば、奇数パリティを使用可能にし、7ビットの文字サイズを設定する必要があるプリンタを仮定します。そのためには、次の例のようなコマンドを入力します。

```
# lpadmin -p neptune -o "stty='parenb parodd cs7'"
```

stty オプション parenb でパリティチェック/生成を使用可能にし、parodd で奇数パリティの生成を設定し、cs7 で文字サイズを7ビットに生成します。

6. 文書が正しく印刷されることを確認します。

```
# lp -d printer-name filename
```

▼ LP 印刷サービスのハングを解除する方法

1. スーパーユーザーまたは **lp** としてログインします。

2. LP 印刷サービスを停止します。

```
# /etc/init.d/lp stop
```

3. LP 印刷サービスを再起動します。

```
# /etc/init.d/lp start
```

LP 印刷サービスが再起動されます。スケジューラが再起動されない場合は、93 ページの「印刷スケジューラを再起動する方法」を参照してください。

アイドル状態になった (ハングした) プリンタの問題を解決する方法

この作業には、プリンタがアイドル状態であってはならないのにアイドル状態になるときに使用する多数の手順が含まれています。通常は各手順を順番に試しますが、順番どおりでなくてもかまいません。

▼ プリンタの準備ができているかチェックする方法

1. プリンタ状態情報を表示します。

```
# lpstat -p printer-name
```

表示される情報は、プリンタがアイドル状態かアクティブ状態か、使用可能か使用不可か、または印刷要求を利用できるか受け付けていないかを示します。すべて正常と思われる場合は、この節の他の手順に進んでください。lpstat コマンドを実行できない場合は、435 ページの「LP 印刷サービスのハングを解除する方法」を参照してください。

2. プリンタが利用できない (要求を受け付けていない) 場合は、プリンタが要求を受け付けるようにします

```
# accept printer-name
```

プリンタは、その印刷待ち行列に要求を受け付け始めます。

3. プリンタが使用不可になっている場合は、再び使用可能にします。

```
# enable printer-name
```

このコマンドでは、待ち行列にある要求を処理するように、プリンタを再び使用可能にします。

印刷のフィルタリングをチェックする方法

lpstat -o コマンドを使用して、印刷のフィルタリングをチェックします。

```
$ lpstat -o luna
luna-10      fred          1261    Mar 12 17:34 being filtered
luna-11      iggy          1261    Mar 12 17:36 on terra
luna-12      jack          1261    Mar 12 17:39 on terra
$
```

待機している最初の要求がフィルタリングされているかどうかを調べます。上の例のような出力になる場合は、ファイルがフィルタリングされています。プリンタはハングせず、要求の処理に少し時間がかかっているだけです。

▼ プリンタ障害の後に印刷を再開する方法

1. プリンタ障害に関するメッセージがある場合は、その障害を解決してください。
プリンタ障害の警告がどのように指定されているかに応じて、メッセージを電子メールで root に送らせるか、root がログインした端末に書き出すことができます。
2. プリンタを再び使用可能にします。

```
# enable printer-name
```

プリンタ障害によって要求がブロックされた場合は、このコマンドで強制的に再試行します。このコマンドが動作しない場合は、この節の他の手順を続行します。

▼ ローカル待ち行列で停止している印刷要求をリモートプリンタに送信する方法

1. 印刷クライアント上で、プリンタサーバーへの印刷要求を、それ以上待ち行列に入れないようにします。

```
# reject printer-name
```

2. 印刷クライアント上で、応答を求める要求をプリンタサーバーに送信します。

```
print_client# ping print_server  
print_server is alive
```

「*print_server not available*」というメッセージが表示される場合は、ネットワークに問題があります。

3. 問題を解決したら、新しい印刷要求を待ち行列に入れられるようにします。

```
# accept printer-name
```

4. 必要であれば、再びプリンタを使用可能にします。

```
# enable printer-name
```

▼ プリンタサーバーの待ち行列で停止する印刷クライアントからの印刷要求を使用可能にする方法

1. プリンタサーバー上で、印刷クライアントからプリンタサーバーへの印刷要求を、それ以上待ち行列に入れないようにします。

```
# reject printer-name
```

2. **lp sched** ログファイルを表示します。

```
# more /var/lp/logs/lpsched
```

表示される情報を参考にして、印刷クライアントからプリンタサーバーへの印刷要求が印刷されない原因を正確に把握できます。

3. 問題を解決したら、新しい印刷要求を待ち行列に入れられるようにします。

```
# accept printer-name
```

4. 必要であれば、プリンタサーバー上で再びプリンタを使用可能にします。

```
# enable printer-name
```

▼ 矛盾したプリンタ状態メッセージを解決する方法

1. プリンタサーバー上でプリンタが使用可能になっており、要求を受け付けているかどうかを確認します。

```
# lpstat -p printer-name
```

印刷クライアントが要求を受け付けているのに、プリンタサーバーが要求を拒否しているときは、矛盾した状態メッセージが表示されます。

2. プリンタサーバー上で、印刷クライアント上のプリンタの定義が、プリンタサーバー上のプリンタの定義と一致するかどうかを確認します。

```
# lpstat -p -l printer-name
```

印刷フィルタ、文字セット、印字ホイール、フォームなど、印刷ジョブコンポーネントの定義を調べて、印刷クライアントとサーバー上で一致し、ローカルユーザーがプリンタサーバーシステムのプリンタにアクセスできることを確認します。

第 32 章

UFS ファイルシステムの不整合解決 (手順)

この章では、fsck エラーメッセージとエラーメッセージの解決に回答できる内容について説明します。

この章で説明する情報は次のとおりです。

- 441 ページの「fsck の一般エラーメッセージ」
- 442 ページの「初期化フェーズでの fsck メッセージ」
- 445 ページの「フェーズ 1: ブロックとサイズの検査のメッセージ」
- 448 ページの「フェーズ 1B: 操作し直して DUPS メッセージを表示する」
- 449 ページの「フェーズ 2: パス名の検査のメッセージ」
- 454 ページの「フェーズ 3: 接続性の検査のメッセージ」
- 457 ページの「フェーズ 4: 参照数の検査のメッセージ」
- 459 ページの「フェーズ 5: シリンダグループ検査のメッセージ」
- 461 ページの「クリーンアップ (後処理) フェーズのメッセージ」

fsck コマンドと、そのコマンドを使用してファイルシステムの完全性を確認する方法については、『Solaris のシステム管理 (デバイスとファイルシステム)』の「UFS ファイルシステムの整合性チェック (手順)」を参照してください。

fsck エラーメッセージ

通常、システムが異常終了し、ファイルシステムの最新の変更がディスクに書き込まれなかった場合に、fsck コマンドが非対話形式で実行され、ファイルシステムが修復されます。ファイルシステムの基本的な非整合状態は自動的に修正されますが、より重大なエラーは修復されません。ファイルシステムを修復する間に、fsck コマンドはこの種の異常終了から予想される非整合状態を修正します。より重大な状況の場合は、エラーが表示されて終了します。

fsck コマンドを対話形式で実行すると、見つかった各非整合状態を表示して小さなエラーを修正します。ただし、より重大なエラーの場合は、非整合状態を表示し、応答を選択するように促します。-y または -n オプションを指定して fsck コマンドを実行する場合、それぞれのエラー条件に対する fsck コマンドのデフォルトの応答は、yes または no に置き換えられます。

修正処置によっては、若干のデータが失われます。失われるデータの量は、fsck の診断出力から判断できます。

fsck コマンドはファイルシステムの複数の段階による検査プログラムです。各段階ごとに、fsck コマンドの異なるフェーズを呼び出し、メッセージも異なります。初期化後に、fsck コマンドはファイルシステムごとに各パスを逐次実行して、ブロックとサイズ、パス名、接続状態、参照数、空きブロックマップを検査します (再構築することもあります)。また、後処理も実行します。

UFS バージョンの fsck コマンドによって実行されるフェーズ (段階) は次のとおりです。

- 初期化
- フェーズ 1 - ブロックとサイズの検査
- フェーズ 2 - パス名の検査
- フェーズ 3 - 接続状態の検査
- フェーズ 4 - 参照数の検査
- フェーズ 5 - シリンダグループの検査

この後の各節では、各フェーズで検出できるエラー条件、表示されるメッセージとプロンプト、および応答できる内容について説明します。

複数のフェーズで表示されるメッセージについては、[441 ページの「fsck の一般エラーメッセージ」](#)を参照してください。それ以外の場合、メッセージは発生するフェーズのアルファベット順に掲載されています。

fsck エラーメッセージには、次の表に示す省略形が含まれています。

表 32-1 エラーメッセージの省略形

省略形	説明
BLK	ブロック番号
DUP	重複ブロック番号
DIR	ディレクトリ名
CG	シリンダグループ
MTIME	ファイルの最終変更時刻
UNREF	非参照

また、多くのメッセージには、i ノード番号などの変数フィールドが含まれています。このマニュアルでは、i ノード番号を *inode-number* のようにイタリック体で記載してあります。たとえば、次の画面メッセージは、

```
INCORRECT BLOCK COUNT I=2529
```

次の例のように記載されています。

```
INCORRECT BLOCK COUNT I=inode-number
```

fsck の一般エラーメッセージ

この節のエラーメッセージは、初期化後のどのフェーズでも表示されることがあります。処理を続けるかどうかのオプションは表示されますが、通常は、致命的だと見なすのが最善の処置です。これらのエラーメッセージは重大なシステム障害を反映しており、ただちに処理する必要があります。この種のメッセージが表示された場合は、n(o) を入力してプログラムを終了してください。問題の原因を判断できない場合は、ご購入先に問い合わせてください。

```
CANNOT SEEK: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で、指定されたブロック番号 *block-number* へ移動させるという要求に失敗した。このメッセージは重大な問題、おそらくハードウェア障害を示す。

ファイルシステムのチェックを続けると、fsck は移動を再び行い、移動できなかったセクタ番号のリストを表示する。ブロックが仮想メモリーバッファークャッシュの一部であれば、fsck は致命的な入出力エラーメッセージを表示して終了する。

対処方法

ディスクにハードウェア障害が発生していると、この問題は継続する。もう一度 fsck を実行してファイルシステムをチェックする。

このチェックでも解決しない場合、ご購入先に問い合わせる。

```
CANNOT READ: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で指定されたブロック番号 *block-number* を読み込むという要求に失敗した。このメッセージは重大な問題、おそらくハードウェア障害を示す。

ファイルシステムのチェックを続けたい場合、fsck は読み取りを再試行して、読み込めなかったセクタ番号のリストを表示する。ブロックが仮想メモリーバッファークャッシュの一部であれば、fsck は致命的な入出力エラーメッセージを表示して終了する。fsck が読み取りに失敗したブロックのいずれかに書き込もうとすると、次のメッセージが表示される。

```
WRITING ZERO'ED BLOCK sector-numbers TO DISK
```

対処方法

ディスクにハードウェア障害が発生していると、この問題は継続する。もう一度 `fsck` を実行してファイルシステムをチェックする。このチェックでも解決しない場合、購入先に問い合わせる。

```
CANNOT WRITE: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で、指定されたブロック番号 *block-number* への書き込みに失敗した。

ファイルシステムのチェックを続けると、`fsck` は書き込みを再度実行し、書き込めなかったセクタ番号のリストを表示する。ブロックが仮想メモリーバッファークャッシュの一部であれば、`fsck` は致命的な入出力エラーメッセージを表示して終了する。

対処方法

ディスクが書き込み保護されている可能性がある。ドライブ上で書き込み保護ロックをチェックする。ディスクにハードウェア障害がある場合、問題は解決しない。もう一度 `fsck` を実行してファイルシステムをチェックする。書き込み保護が原因でない場合、あるいはファイルシステムを再チェックしても問題が解決しない場合は、購入先に問い合わせる。

初期化フェーズでの `fsck` メッセージ

初期化フェーズでは、コマンド行構文がチェックされます。ファイルシステムのチェックを実行する前に、`fsck` はテーブルを設定してファイルを開きます。

この節のメッセージは、コマンド行オプション、メモリー要求、ファイルのオープン、ファイルの状態、ファイルシステムのサイズチェック、およびスクラッチファイルの作成によるエラー条件に関するものです。ファイルシステムを修復する間に、どんな初期化エラーが発生した場合も、`fsck` は終了します。

```
Can't roll the log for device-name.
```

```
DISCARDING THE LOG MAY DISCARD PENDING TRANSACTIONS.
```

```
DISCARD THE LOG AND CONTINUE?
```

エラーの発生原因

UFS ファイルシステムのエラーチェックを行う前に、記録されている UFS ファイルシステムのトランザクションログをフラッシュできなかった。

対処方法

`yes` と応答する場合には、ファイルシステム操作がログに記録されていても、ファイルシステムに適用されなければ、そのファイルシステム操作は失われる。この場合、`fsck` は、通常と同様の検査を実行してフェーズ 5 で次の質問が表示される。

```
FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)
```

ここで yes と応答すると、ログに使用したブロックを回収する。次回にログを有効にしてファイルシステムをマウントすると、ログが再作成される。

no と応答すると、ログを保持したまま終了するが、ファイルシステムはマウントできない。

```
bad inode number inode-number to ginode
```

エラーの発生原因

inode-number が存在しないため、内部エラーが発生した。fsck は終了する。

対処方法

ご購入先に問い合わせる。

```
cannot alloc size-of-block map bytes for blockmap  
cannot alloc size-of-free map bytes for freemap  
cannot alloc size-of-state map bytes for statemap  
cannot alloc size-of-lncntp bytes for lncntp
```

エラーの発生原因

内部テーブル用のメモリー要求に失敗した。fsck は終了する。このメッセージは、即座に処理しなければならない重大なシステム障害を示す。他のプロセスが大量のシステム資源を使用していると、このエラー条件が発生することがある。

対処方法

他のプロセスを終了すると問題を解決できることがある。解決できない場合は、ご購入先に問い合わせる。

```
Can't open checklist file: filename
```

エラーの発生原因

ファイルシステムの検査リストファイル *filename* (通常は /etc/vfstab) を開いて読み込めない。fsck は終了する。

対処方法

ファイルの有無と、そのアクセスモードで読み取りが可能かどうかを検査する。

```
Can't open filename
```

エラーの発生原因

fsck はファイルシステム *filename* を開けなかった。対話形式で実行している場合、fsck はこのファイルシステムを無視し、次に指定されたファイルシステムの検査を続ける。

対処方法

そのファイルシステムの raw デバイスファイルに読み取り、または書き込みができるかどうかをチェックする。

```
Can't stat root
```

エラーの発生原因

fsck はルートディレクトリに関する統計情報要求に失敗した。fsck は終了する。

対処方法

このメッセージは、重大なシステム障害を示す。ご購入先に問い合わせる。

Can't stat *filename*
Can't make sense out of name *filename*

エラーの発生原因

`fsck` はファイルシステム *filename* に関する統計情報要求に失敗した。対話形式で実行している場合、`fsck` はこのファイルシステムを無視し、次に指定されたファイルシステムの検査を続ける。

対処方法

ファイルシステムの有無とそのアクセスモードをチェックする。

filename: (NO WRITE)

エラーの発生原因

`-n` オプションが指定されているか、`fsck` はファイルシステム *filename* を書き込み用に開けなかった。`fsck` を非書き込みモードで実行中であれば、診断メッセージはすべて表示されるが、`fsck` は何も修正しない。

対処方法

`-n` を指定しなかった場合は、指定したファイルのタイプを検査する。通常ファイル名の可能性がある。

IMPOSSIBLE MINFREE=*percent* IN SUPERBLOCK (SET TO DEFAULT)

エラーの発生原因

スーパーブロックの最小容量が 99 パーセントを超えているか、0 パーセント未満である。

対処方法

`minfree` パラメータをデフォルトの 10 パーセントに設定するには、デフォルトプロンプトから *y* と入力する。エラー条件を無視するには、デフォルトプロンプトから *n* と入力する。

filename: BAD SUPER BLOCK: *message*
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g., `fsck[-f ufs] -o b=# [special ...]`
where # is the alternate superblock. See `fsck_ufs(1M)`

エラーの発生原因

スーパーブロックが破損している。

対処方法

次のいずれかのメッセージが表示される。

CPG OUT OF RANGE
FRAGS PER BLOCK OR FRAGSIZE WRONG
INODES PER GROUP OUT OF RANGE
INOPB NONSENSICAL RELATIVE TO BSIZE
MAGIC NUMBER WRONG
NCG OUT OF RANGE
NCYL IS INCONSISTENT WITH NCG*CPG
NUMBER OF DATA BLOCKS OUT OF RANGE
NUMBER OF DIRECTORIES OUT OF RANGE
ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
SIZE TOO LARGE

BAD VALUES IN SUPERBLOCK

代替スーパーブロックを使用して `fsck` を再実行してみる。手始めにブロック 32 を指定するとよい。スライスに対して `newfs -N` コマンドを実行すると、スーパーブロックの代替コピーの位置を調べることができる。-N を指定しないと、`newfs` は既存のファイルシステムを上書きするので注意する。

UNDEFINED OPTIMIZATION IN SUPERBLOCK (SET TO DEFAULT)

エラーの発生原因

スーパーブロックの最適化パラメータが `OPT_TIME` でも `OPT_SPACE` でもない。

対処方法

ファイルシステム上で処理の実行時間を最小限度まで短縮するには、`SET TO DEFAULT` プロンプトから `y` を入力する。このエラー条件を無視するには、`n` と入力する。

フェーズ 1: ブロックとサイズの検査のメッセージ

このフェーズでは、`i` ノードリストを検査します。次の処理中に検出されたエラー条件が表示されます。

- `i` ノードのタイプを検査する
- ゼロリンク数テーブルを設定する
- 不良ブロックまたは重複ブロックの有無を `i` ノードブロック番号で検査する
- `i` ノードのサイズを検査する
- `i` ノードの形式を検査する

ファイルシステムの修復 (`preen`) 中は、`INCORRECT BLOCK COUNT`、`PARTIALLY TRUNCATED INODE`、`PARTIALLY ALLOCATED INODE`、および `UNKNOWN FILE TYPE` を除き、このフェーズ中にどのエラーが発生した場合も、`fsck` が終了します。

フェーズ 1 では、次のメッセージ (アルファベット順) が発生する可能性があります。

```
block-number BAD I=inode-number
```

エラーの発生原因

`i` ノード `inode-number` に、ファイルシステム内の最初のデータブロックより小さい番号または最後のデータブロックより大きい番号が付いたブロック番号 `block-number` が入っている。`i` ノード `inode-number` 内にファイルシステムの範囲外のブロック番号が多すぎると、このエラー条件のためにフェーズ 1 で「`EXCESSIVE BAD BLKS`」エラーメッセージが生成されることがある。フェーズ 2 と 4 では、このエラー条件が原因で「`BAD/DUP`」エラーメッセージが生成される。

対処方法

なし

```
BAD MODE: MAKE IT A FILE?
```

エラーの発生原因

指定された *i* ノードの状態がすべて、ファイルシステムの損傷を示す 1 に設定されている。このメッセージは、`fsck -y` が実行された後で繰り返し表示される場合以外は、物理的なディスクの損傷を示すものではない。

対処方法

y と入力して *i* ノードを妥当な値に初期化し直す。

```
BAD STATE state-number TO BLKERR
```

エラーの発生原因

内部エラーによって `fsck` の状態マップが破壊されたため、不可能な値 *state-number* を示す。`fsck` は即座に終了する。

対処方法

ご購入先に問い合わせる。

```
block-number DUP I=inode-number
```

エラーの発生原因

i ノード *inode-number* には、同じ *i* ノードまたは別の *i* ノードがすでに取得したブロック番号 *block-number* が入っている。このエラー条件が発生した場合に、*i* ノード *inode-number* 内にこの種のブロック番号が多すぎると、フェーズ 1 では「EXCESSIVE DUP BLKS」エラーメッセージが生成されることがある。このエラー条件によってフェーズ 1B が呼び出され、フェーズ 2 と 4 で「BAD/DUP」エラーメッセージが生成される。

対処方法

なし

```
DUP TABLE OVERFLOW (CONTINUE)
```

エラーの発生原因

`fsck` の内部テーブルには、重複するブロック番号が入る余地がない。 `-op` (`preen`、修復) オプションが指定されていると、プログラムは終了する。

対処方法

プログラムを続行するには、`CONTINUE` プロンプトから *y* と入力する。このエラーが発生すると、ファイルシステムを完全には検査できない。別の重複ブロックが見つかり、このエラー条件が再発する。使用可能な仮想メモリの容量を (プロセスを終了し、スワップ空間を拡張して) 大きくし、もう一度 `fsck` を実行してファイルシステムを検査し直す。プログラムを終了するには *n* と入力する。

```
EXCESSIVE BAD BLOCKS I=inode-number (CONTINUE)
```

エラーの発生原因

i ノード *inode-number* に関連付けられたファイルシステム内の最初のデータブロックより小さい番号か、最後のブロックより大きい番号を持つブロックが多すぎる (通常は 10 以上)。 `-op` (`preen`、修復) オプションを指定すると、プログラムは終了する

対処方法

プログラムを続行するには、`CONTINUE` プロンプトから *y* と入力する。このエラーが発生すると、ファイルシステムを完全には検査できない。もう一度 `fsck` を実行

してファイルシステムを検査し直す必要がある。プログラムを終了するには `n` と入力する。

EXCESSIVE DUP BLKS I=*inode-number* (CONTINUE)

エラーの発生原因

同じ *i* ノード、別の *i* ノード、または空きリストが取得するブロック数が多すぎる (通常は 10 以上)。 `-o p` (`preen`、修復) オプションが指定されていると、プログラムは終了する。

対処方法

プログラムを続行するには、CONTINUE プロンプトから `y` と入力する。このエラーが発生すると、ファイルシステムを完全には検査できない。もう一度 `fsck` を実行してファイルシステムを検査し直す必要がある。プログラムを終了するには `n` と入力する。

INCORRECT BLOCK COUNT I=*inode-number* (*number-of-BAD-DUP-or-missing-blocks* should be *number-of-blocks-in-filesystem*) (CORRECT)

エラーの発生原因

i ノード *inode-number* のブロック数は *number-of-BAD-DUP-or-missing-blocks* であるが、*number-of-blocks-in-filesystem* でなければならない。修復 (`preen`) の場合、`fsck` は数を訂正する。

対処方法

i ノード *inode-number* のブロック数を *number-of-blocks-in-filesystem* に置き換えるには、CORRECT プロンプトから `y` と入力する。プログラムを終了するには `n` と入力する。

LINK COUNT TABLE OVERFLOW (CONTINUE)

エラーの発生原因

`fsck` の内部テーブルには、リンク数が 0 の割り当て済み *i* ノードが入る余地がない。 `-o p` (`preen`、修復) オプションを指定すると、プログラムは終了するので、`fsck` を手作業で実行する必要がある。

対処方法

プログラムを続行するには、CONTINUE プロンプトから `y` と入力する。リンク数が 0 の別の割り当て済みブロックが見つかったら、このエラー条件が再発する。このエラーが発生すると、ファイルシステムを完全には検査できない。もう一度 `fsck` を実行してファイルシステムを検査し直す必要がある。プロセスをいくつか終了するか、スワップ領域を拡張して、使用可能な仮想メモリーを増やしてから、`fsck` を実行し直す。プログラムを終了するには `n` と入力する。

PARTIALLY ALLOCATED INODE I=*inode-number* (CLEAR)

エラーの発生原因

i ノード *inode-number* は割り当て済みでも未割り当てでもない。 `-o p` (`preen`、修復) オプションを指定すると、この *i* ノードは消去される。

対処方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、`y` と入力する。これにより、この *i* ノードを指すディレクトリごとに、フェーズ 2 でエラー条件 UNALLOCATED が生成されることがある。このエラー条件を無視するには、`n` と入

力する。応答しなくてよいのは、この問題を他の手段で解決しようとする場合だけである。

PARTIALLY TRUNCATED INODE I=*inode-number* (SALVAGE)

エラーの発生原因

`fsck` で、割り当てられたブロック数よりも短い *i* ノード *inode-number* が見つかった。この条件が発生するのは、ファイルの切り捨て中にシステムがクラッシュした場合だけである。ファイルシステムを修復しているとき、`fsck` は指定されたサイズへの切り捨てを完了する。

対処方法

i ノード内で指定したサイズへの切り捨てを完了するには、`SALVAGE` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

UNKNOWN FILE TYPE I=*inode-number* (CLEAR)

エラーの発生原因

i ノード *inode-number* のモードのワードは、この *i* ノードがパイプ、特殊文字 *i* ノード、特殊ブロック *i* ノード、通常 *i* ノード、シンボリックリンク、FIFO ファイル、またはディレクトリ *i* ノードでないことを示す。-`op` (`preen`、修復) オプションを指定すると、この *i* ノードは消去される。

対処方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、`CLEAR` プロンプトから `y` と入力する。これにより、この *i* ノードを指すディレクトリエントリごとに、フェーズ 2 でエラー条件 `UNALLOCATED` が生成される。このエラー条件を無視するには、`n` と入力する。

フェーズ 1B: 操作し直して DUPS メッセージを表示する

ファイルシステム内で重複ブロックが見つかったと、次のメッセージが表示されます。

block-number DUP I=*inode-number*

エラーの発生原因

i ノード *inode-number* には、すでに同じ *i* ノードまたは別の *i* ノードによって取得されたブロック番号 *block-number* が入っている。このエラー条件によって、フェーズ 2 で `BAD/DUP` エラーメッセージが生成される。重複ブロックを持つ *i* ノードは、このエラー条件とフェーズ 1 の `DUP` エラー条件を検査すれば判断できる。

対処方法

重複ブロックが見つかったと、ファイルシステムが再び走査され、以前にそのブロックを取得した *i* ノードが検索される。

フェーズ 2: パス名の検査のメッセージ

このフェーズでは、フェーズ 1 と 1B で見つかった不良 i ノードを指すディレクトリエントリが削除される。次の原因によるエラー条件が表示されます。

- 不正なルート i ノードモードと状態
- 範囲外のディレクトリ i ノードポインタ
- 不良 i ノードを指すディレクトリエントリ
- ディレクトリ完全性検査

ファイルシステムを修復している場合は (-o p (preen、修復) オプション)、このフェーズでどのエラーが発生した場合も、fsck が終了します。ただし、ブロックサイズの倍数でないディレクトリ、重複ブロックと不良ブロック、範囲外の i ノード、過剰なハードリンクに関連するエラーは除きます。

フェーズ 2 では、次のメッセージ (アルファベット順) が表示される可能性があります。

```
BAD INODE state-number TO DESCEND
```

エラーの発生原因

fsck の内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンに、無効な状態 *state-number* が渡された。fsck は終了する。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
BAD INODE NUMBER FOR '.' I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「..」の i ノード番号が *inode-number* に等しくないディレクトリ *inode-number* が見つかった。

対処方法

「..」の i ノード番号を *inode-number* に等しくなるように変更するには、FIX プロンプトから *y* と入力する。「..」の i ノード番号を変更しない場合は、*n* と入力する。

```
BAD INODE NUMBER FOR '..' I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「..」の i ノード番号が *inode-number* の親に等しくないディレクトリ *inode-number* が見つかった。

対処方法

「..」の i ノード番号を *inode-number* の親に等しくなるように変更するには、FIX プロンプトに *y* を入力する (ルート i ノード内の「..」は、それ自体を指すので注意する)。「..」の i ノード番号を変更しない場合は、*n* と入力する。

```
BAD RETURN STATE state-number FROM DESCEND
```

エラーの発生原因

fsck の内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンから、不可能な状態 *state-number* が返された。fsck は終了する。

対処方法

このメッセージが表示される場合は、ご購入先に問い合わせる。

```
BAD STATE state-number FOR ROOT INODE
```

エラーの発生原因

内部エラーによって、ルート *i* ノードに不可能な状態 *state-number* が割り当てられた。fsck は終了する。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
BAD STATE state-number FOR INODE=inode-number
```

エラーの発生原因

内部エラーによって、*i* ノード *inode-number* に不可能な状態 *state-number* が割り当てられた。fsck は終了する。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
DIRECTORY TOO SHORT I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

サイズ *file-size* が最小ディレクトリサイズより小さいディレクトリ *filename* が見つかった。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

対処方法

ディレクトリのサイズを最小ディレクトリサイズまで大きくするには、FIX プロンプトから *y* と入力する。このディレクトリを無視するには *n* と入力する。

```
DIRECTORY filename: LENGTH file-size NOT MULTIPLE OF block-number (ADJUST)
```

エラーの発生原因

サイズ *file-size* がディレクトリブロックのサイズ *block-number* の倍数でないディレクトリ *filename* が見つかった。

対処方法

長さを適切なブロックサイズに切り上げるには、*y* と入力する。ファイルシステムを修復しているとき (-o p (preen、修復)、オプション) は、fsck は警告のみを表示してディレクトリを調整する。この条件を無視するには *n* と入力する。

```
DIRECTORY CORRUPTED I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time DIR=filename (SALVAGE)
```

エラーの発生原因

内部状態の整合性がないディレクトリが見つかった。

対処方法

次のディレクトリ境界 (通常は 512 バイトの境界) までのすべてのエンTRIESを放棄するには、SALVAGE プロンプトから *y* と入力する。この処置によって、最高で 42 個のエンTRIESを放棄できる。この処置は、他の回復作業に失敗した場合にのみ実行する。問題のディレクトリを変更せずに、次のディレクトリ境界までスキップして読み取りを再開するには、*n* と入力する。

```
DUP/BAD I=inode-number OWNER=O MODE=M SIZE=file-size
MTIME=modification-time TYPE=filename (REMOVE)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ディレクトリまたはファイルエンTRIES *filename*、*i* ノード *inode-number* に関連付けられた重複ブロックまたは不良ブロックが見つかった。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、ディレクトリまたはファイル名 *filename* が表示される。-p (preen、修復) オプションを指定すると、重複または不良ブロックが削除される。

対処方法

ディレクトリまたはファイルのエンTRIES *filename* を削除するには、REMOVE プロンプトから *y* と入力する。このエラー条件を無視するには、*n* と入力する。

```
DUPS/BAD IN ROOT INODE (REALLOCATE)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ファイルシステムのルート *i* ノード (通常は *i* ノード番号 2) に、重複ブロックまたは不良ブロックが見つかった。

対処方法

ルート *i* ノードの既存の内容を消去して再割り当てを行うには、REALLOCATE プロンプトから *y* と入力する。一般にルート内で検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。*n* と入力すると、CONTINUE プロンプトが表示される。*y* と入力して CONTINUE プロンプトに応答すると、ルート *i* ノード内の DUPS/BAD エラー条件を無視して、ファイルシステムの検査を続行する。ルート *i* ノードが不正であれば、他の多数のエラーメッセージが生成されることがある。*n* の場合は、プログラムを終了する。

```
EXTRA '.' ENTRY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「.」のエンTRIESが複数個入っているディレクトリ *inode-number* が見つかった。

対処方法

「.」の余分なエンTRIESを削除するには、FIX プロンプトから *y* と入力する。問題のディレクトリを変更しない場合は、*n* と入力する。

```
EXTRA '..' ENTRY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「..」(親ディレクトリ)のエントリが複数個入っているディレクトリ *inode-number* が見つかった。

対処方法

「..」(親ディレクトリ)の余分なエントリを削除するには、FIX プロンプトから *y* と入力する。問題のディレクトリを変更しない場合は、*n* と入力する。

```
hard-link-number IS AN EXTRANEOUS HARD LINK TO A DIRECTORY filename (REMOVE)
```

エラーの発生原因

fsck によって、ディレクトリ *filename* へのハードリンク *hard-link-number* にエラーが見つかった。修復 (*preen*) しているとき (*-op* オプション)、*fsck* はエラーのあるハードリンクを無視する。

対処方法

エラーのあるエントリ *hard-link-number* を削除するには、プロンプトから *y* と入力する。このエラー条件を無視するには、*n* と入力する。

```
inode-number OUT OF RANGE I=inode-number NAME=filename (REMOVE)
```

エラーの発生原因

ディレクトリエントリ *filename* には、*i* ノードリストの終わりより大きい *i* ノード番号 *inode-number* が付いている。-*p* (*preen*、修復) オプションを指定すると、*i* ノードが自動的に削除される。

対処方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから *y* と入力する。このエラー条件を無視するには、*n* と入力する。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size  
MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

最初のエントリ(「..」のエントリ)に未割り当てのディレクトリ *inode-number* が見つかった。

対処方法

i ノード番号が *inode-number* に等しい「..」のエントリを構築するには、FIX プロンプトから *y* と入力する。問題のディレクトリを変更しない場合は、*n* と入力する。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size  
MTIME=modification-time DIR=filename CANNOT FIX, FIRST ENTRY IN  
DIRECTORY CONTAINS filename
```

エラーの発生原因

最初のエントリが *filename* となっているディレクトリ *inode-number* が見つかった。*fsck* はこの問題を解決できない。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size  
MTIME=modification-time DIR=filename CANNOT FIX, INSUFFICIENT  
SPACE TO ADD '..'
```

エラーの発生原因

最初のエンタリが「`.`」でないディレクトリ *inode-number* が見つかった。 `fsck` は問題を解決できない。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

第2のエンタリが割り当てられていないディレクトリ *inode-number* が見つかった。

対処方法

i ノード番号が *inode-number* の親に等しい「`..`」のエンタリを構築するには、`FIX` プロンプトから `y` と入力する (ルート *i* ノード内の「`..`」は、それ自体を指すので注意する)。問題のディレクトリを変更しない場合は、`n` と入力する。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, SECOND ENTRY IN
DIRECTORY CONTAINS filename
```

エラーの発生原因

第2のエンタリが *filename* となっているディレクトリ *inode-number* が見つかった。
`fsck` はこの問題を解決できない。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, INSUFFICIENT SPACE
TO ADD '..'
```

エラーの発生原因

第2のエンタリが「`..`」(親ディレクトリ) でないディレクトリ *inode-number* が見つかった。`fsck` はこの問題を解決できない。

対処方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
NAME TOO LONG filename
```

エラーの発生原因

長すぎるパス名が見つかった。通常、これはファイルシステムの名前空間内のループを示す。特権を持つユーザーがディレクトリへの循環リンクを作成すると、このエラーが発生することがある。

対処方法

循環リンクを削除する。

```
ROOT INODE UNALLOCATED (ALLOCATE)
```

エラーの発生原因

ルート *i* ノード (通常は *i* ノード番号 2) に割り当てモードビットがない。

対処方法

i ノード 2 をルート i ノードとして割り当てるには、ALLOCATE プロンプトから *y* と入力する。一般にルート内で検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。プログラムを終了するには *n* と入力する。

```
ROOT INODE NOT DIRECTORY (REALLOCATE)
```

エラーの発生原因

ファイルシステムのルート i ノード (通常は i ノード番号 2) はディレクトリ i ノードではない。

対処方法

ルート i ノードの既存の内容を消去して再割り当てを行うには、REALLOCATE プロンプトから *y* と入力する。一般にルート内で検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。fsck に FIX プロンプトを表示させるには、*n* と入力する。

```
UNALLOCATED I=inode-number OWNER=UID MODE=file-mode SIZE=file-size  
MTIME=modification-time type=filename (REMOVE)
```

エラーの発生原因

ディレクトリまたはファイルのエントリ *filename* は、未割り当ての i ノード *inode-number* を指している。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびファイル名 *filename* が表示される。

対処方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから *y* と入力する。このエラー条件を無視するには、*n* と入力する。

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time DIR=filename (REMOVE)
```

エラーの発生原因

ディレクトリエントリ *filename* のサイズ *file-size* が 0 になっている。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

対処方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから *y* と入力する。これにより、フェーズ 4 で「BAD/DUP」エラーメッセージが表示される。このエラー条件を無視するには、*n* と入力する。

フェーズ 3: 接続性の検査のメッセージ

このフェーズでは、フェーズ 2 で検査したディレクトリが検査され、次の原因によるエラー条件が表示されます。

- 参照されないディレクトリ

■ lost+found ディレクトリが見つからないまたは一杯

フェーズ 3 では、次のメッセージがアルファベット順に表示される可能性があります。

```
BAD INODE state-number TO DESCEND
```

エラーの発生原因

内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンに、不可能な状態 *state-number* が渡された。fsck は終了する。

対処方法

このエラーが発生する場合は、ご購入先に問い合わせる。

```
DIR I=inode-number1 CONNECTED. PARENT WAS I=inode-number2
```

エラーの発生原因

これは、ディレクトリ *i* ノード *inode-number1* が lost+found ディレクトリに正常に接続されていることを示す。ディレクトリ *i* ノード *inode-number1* の親 *i* ノード *inode-number2* は、lost+found ディレクトリの *i* ノード番号に置き換えられる。

対処方法

なし

```
DIRECTORY filename LENGTH file-size NOT MULTIPLE OF block-number (ADJUST)
```

エラーの発生原因

サイズ *file-size* がディレクトリのブロックサイズ *B* の倍数でないディレクトリ *filename* が見つかった (この条件は、フェーズ 2 で調整しなければ、フェーズ 3 で再発することがある)。

対処方法

長さを適切なブロックサイズまで切り上げるには、ADJUST プロンプトから *y* と入力する。修復しているときは、fsck は警告を表示してディレクトリを調整する。このエラー条件を無視するには、*n* と入力する。

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

エラーの発生原因

lost+found のエントリがディレクトリではない。

対処方法

ディレクトリ *i* ノードを割り当てて、それを参照する lost+found ディレクトリを変更するには、REALLOCATE プロンプトから *y* と入力する。以前に lost+found ディレクトリによって参照されていた *i* ノードは消去されず、非参照の *i* ノードとして再び取得されるか、このフェーズの後半でそのリンク数が調整される。

lost+found ディレクトリを作成できない場合は、「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージが表示され、消失 *i* ノードへのリンク試行が中止される。このエラーにより、フェーズ 4 で UNREF エラーメッセージが生成される。フェーズ 4 で UNREF エラーメッセージを生成する消失 *i* ノードへのリンク試行を中止するには、*n* と入力する。

```
NO lost+found DIRECTORY (CREATE)
```

エラーの発生原因

ファイルシステムのルートディレクトリ内に `lost+found` ディレクトリがない。修復するときに、`fsck` は `lost+found` ディレクトリを作成しようとする。

対処方法

ファイルシステムのルート内で `lost+found` ディレクトリを作成するには、`CREATE` プロンプトから `y` と入力する。このため、「`NO SPACE LEFT IN / (EXPAND)`」というメッセージが表示されることがある。`lost+found` ディレクトリを作成できない場合は、`fsck` によって「`CANNOT CREATE lost+found DIRECTORY`」というメッセージが表示され、消失 `i` ノードへのリンク試行が中止される。このエラーにより、フェーズ 4 で `UNREF` エラーメッセージが生成される。消失 `i` ノードへのリンク試行を中止するには、`n` と入力する。

```
NO SPACE LEFT IN /lost+found (EXPAND)
```

エラーの発生原因

使用可能な領域がないため、ファイルシステムのルートディレクトリ内で、`lost+found` ディレクトリに別のエントリを追加できない。修復するときに、`fsck` は `lost+found` ディレクトリを拡張する。

対処方法

`lost+found` ディレクトリを拡張して新しいエントリを追加する余地をつくるには、`EXPAND` プロンプトから `y` と入力する。拡張試行に失敗すると、`fsck` によって「`SORRY. NO SPACE IN lost+found DIRECTORY`」というメッセージが表示され、`lost+found` ディレクトリへのファイルリンク要求が中止される。このエラーにより、フェーズ 4 の後半で `UNREF` エラーメッセージが生成される。`lost+found` ディレクトリ内で不要なエントリを削除する。修復中にこのエラーが起きると、`fsck` は終了する。消失 `i` ノードへのリンク試行を中止するには、`n` と入力する。

```
UNREF DIR I=inode-number OWNER=UID MODE=file-mode SIZE=file-size  
MTIME=modification-time (RECONNECT)
```

エラーの発生原因

ファイルシステムの走査中に、ディレクトリ `i` ノード `inode-number` がディレクトリエントリに接続されなかった。ディレクトリ `i` ノード `inode-number` の所有者 `UID`、モード `file-mode`、サイズ `file-size`、および変更時刻 `modification-time` が表示される。修復しているときは、ディレクトリサイズが 0 でなければ、`fsck` は空でないディレクトリ `i` ノードを接続し直す。それ以外の場合、`fsck` はディレクトリ `i` ノードを消去する。

対処方法

ディレクトリ `i` ノード `inode-number` を `lost+found` ディレクトリに接続し直すには、`RECONNECT` プロンプトから `y` と入力する。ディレクトリが再び正常に接続されると、「`CONNECTED`」というメッセージが表示される。それ以外の場合は、`lost+found` エラーメッセージのいずれかが表示される。このエラー条件を無視するには、`n` と入力する。このエラーにより、フェーズ 4 で `UNREF` エラー条件が発生する。

フェーズ 4: 参照数の検査のメッセージ

このフェーズでは、フェーズ 2 と 3 で取得したリンク数情報が検査されます。次の原因によるエラー条件が表示されます。

- 非参照ファイル
- `lost+found` ディレクトリが見つからないまたは一杯
- ファイル、ディレクトリ、シンボリックリンク、または特殊ファイルの不正なリンク数
- 非参照ファイル、シンボリックリンク、ディレクトリ
- ファイルとディレクトリ内の不良ブロックまたは重複ブロック
- 不正な合計空き `i` ノード数

このフェーズのすべてのエラー (`lost+found` ディレクトリ内の容量不足を除く) は、ファイルシステムを修復するときに解決できます。

フェーズ 4 では、次のメッセージ (アルファベット順) が表示される可能性があります。

```
BAD/DUP type I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ファイルまたはディレクトリ `i` ノード `inode-number` に関連付けられた重複ブロックまたは不良ブロックが見つかった。 `i` ノード `inode-number` の所有者 `UID`、モード `file-mode`、サイズ `file-size`、および変更時刻 `modification-time` が表示される。

対処方法

`i` ノード `inode-number` の内容を消去して割り当てを解除するには、`CLEAR` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

(CLEAR)

エラーの発生原因

直前の `UNREF` エラーメッセージで記述された `i` ノードを再び接続できない。ファイルシステムを修復していると、ファイルを接続し直すには容量が足りないため `fsck` が終了するので、このメッセージは表示されない。

対処方法

`i` ノードの内容を消去して割り当てを解除するには、`CLEAR` プロンプトから `y` と入力する。直前のエラー条件を無視するには、`n` と入力する。

```
LINK COUNT type I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size
MTIME=modification-time COUNT link-count SHOULD BE
corrected-link-count (ADJUST)
```

エラーの発生原因

ディレクトリまたはファイル `i` ノード `inode-number` のリンク数は `link-count` になっているが、`corrected-link-count` でなければならない。 `i` ノード `inode-number` の所有者 `UID`、モード `file-mode`、サイズ `file-size`、および変更時刻 `modification-time` が表示される。 `-o p` (`preen`、修復) オプションを指定すると、参照数が増えていない限

り、リンク数が調整される。この条件は、ハードウェア障害がなければ発生しない。参照数が修復中に増えると、fsckは「LINK COUNT INCREASING」というメッセージを表示して終了する。

対処方法

ディレクトリまたはファイル *i* ノード *inode-number* のリンク数を *corrected-link-count* に置き換えるには、ADJUST プロンプトから *y* と入力する。このエラー条件を無視するには、*n* と入力する。

lost+found IS NOT A DIRECTORY (REALLOCATE)

エラーの発生原因

lost+found のエントリがディレクトリではない。

対処方法

ディレクトリ *i* ノードを割り当てて、それを参照する lost+found ディレクトリを変更するには、REALLOCATE プロンプトから *y* と入力する。lost+found による以前の *i* ノード参照は消去されない。非参照 *i* ノードとして再び取得されるか、そのリンク数がこのフェーズの後半で調整される。lost+found ディレクトリを作成できない場合は、「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージが表示され、消失 *i* ノードへのリンク試行が中止される。これにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。消失 *i* ノードへのリンク試行を中止するには、*n* と入力する。

NO lost+found DIRECTORY (CREATE)

エラーの発生原因

ファイルシステムのルートディレクトリ内に lost+found ディレクトリがない。修復するときに、fsck は lost+found ディレクトリを作成しようとする。

対処方法

ファイルシステムのルート内で lost+found ディレクトリを作成するには、CREATE プロンプトから *y* と入力する。lost+found ディレクトリを作成できない場合は、fsck によって「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージが表示され、消失 *i* ノードへのリンク試行が中止される。これにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。消失 *i* ノードへのリンク試行を中止するには、*n* と入力する。

NO SPACE LEFT IN / lost+found (EXPAND)

エラーの発生原因

ファイルシステムのルートディレクトリ内で、lost+found ディレクトリに別のエントリを追加する容量がない。修復するときに、fsck は lost+found ディレクトリを拡張する。

対処方法

lost+found ディレクトリを拡張して新しいエントリを追加する余地をつくるには、EXPAND プロンプトから *y* と入力する。拡張試行に失敗すると、fsck によって「SORRY. NO SPACE IN lost+found DIRECTORY」というメッセージが表示され、lost+found ディレクトリへのファイルリンク要求が中止される。このエラーにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。lost+found ディレクトリ内で不要なエントリを削除する。修復 (-o p オプション) が有効なときは、このエラーによって fsck が終了する。消失 *i* ノードへ

のリンク試行を中止するには、n と入力する。

```
UNREF FILE I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (RECONNECT)
```

エラーの発生原因

ファイルシステムを走査したときに、ファイル *i* ノード *inode-number* がディレクトリエントリに接続されなかった。 *i* ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。 `fsck` が修復しているときに、ファイルのサイズまたはリンク数が 0 であれば、そのファイルは消去される。それ以外の場合は再び接続される。

対処方法

i ノード *inode-number* を `lost+found` ディレクトリ内のファイルシステムに接続し直すには、y と入力する。 *i* ノード *inode-number* を `lost+found` ディレクトリに接続できないと、このエラーによってフェーズ 4 で `lost+found` エラーメッセージが生成されることがある。このエラー条件を無視するには、n と入力する。このエラーが発生すると、フェーズ 4 で必ず `CLEAR` エラー条件が呼び出される。

```
UNREF type I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

エラーの発生原因

ファイルシステムを走査するときに、*i* ノード *inode-number* (その *type* はディレクトリまたはファイル) がディレクトリエントリに接続されなかった。 *i* ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。 `fsck` が修復しているときに、ファイルのサイズまたはリンク数が 0 であれば、そのファイルは消去される。それ以外の場合は再び接続される。

対処方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、`CLEAR` プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time (CLEAR)
```

エラーの発生原因

ディレクトリエントリ *filename* のサイズ *file-size* が 0 になっている。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

対処方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、y と入力する。このエラー条件を無視するには、n と入力する。

フェーズ 5: シリンダグループ検査のメッセージ

このフェーズでは、空きブロックと使用済み *i* ノードのマップが検査されます。次の原因によるエラー条件が表示されます。

- 使用済み *i* ノードマップから欠落している割り当て済み *i* ノード

- 空きブロックマップから欠落している空きブロック
- 使用済み *i* ノードマップ内の空き *i* ノード
- 不正な合計空きブロック数
- 不正な合計使用済み *i* ノード数

フェーズ 5 では、次のメッセージがアルファベット順に表示される可能性があります。

BLK(S) MISSING IN BIT MAPS (SALVAGE)

エラーの発生原因

シリンダグループのブロックマップから空きブロックがいくつか欠落している。修復中に、`fsck` はマップを作成し直す。

対処方法

空きブロックマップを作成し直すには、`SALVAGE` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

CG *character-for-command-option*: BAD MAGIC NUMBER

エラーの発生原因

シリンダグループ *character-for-command-option* のマジック番号が間違っている。通常、このエラーはシリンダグループマップが破壊されていることを示す。対話形式で実行している場合は、シリンダグループに再度の作成が必要であることを示すマークが付けられる。ファイルシステムを修復している場合は、`fsck` が終了する。

対処方法

このエラーが発生する場合は、ご購入先に問い合わせる。

FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)

エラーの発生原因

空きブロック数の実際の数が、ファイルシステムのスーパーブロック内の空きブロック数と一致しない。 `-o p (preen, 修復)` オプションを指定した場合は、スーパーブロック内の空きブロック数が自動的に修正される。

対処方法

スーパーブロックの空きブロック情報を作成し直すには、`SALVAGE` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

SUMMARY INFORMATION BAD (SALVAGE)

エラーの発生原因

集計情報が間違っている。修復していると、`fsck` は集計情報を計算し直す。

対処方法

集計情報を作成し直すには、`SALVAGE` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

クリーンアップ (後処理) フェーズのメッセージ

ファイルシステムのチェックが終わると、クリーンアップ処理がいくつか実行されます。クリーンアップフェーズでは、次の状態メッセージが表示されます。

```
number-of files, number-of-files  
used, number-of-files free (number-of frags, number-of blocks,  
percent fragmentation)
```

上記のメッセージは、検査されたファイルシステムに、フラグメントサイズの *number-of* 個のブロックを使用中の *number-of* 個のファイルが入っていることと、ファイルシステム内でフラグメントサイズのブロックが *number-of* 個空いていることを示します。括弧内の数は、空いている数を *number-of* 個の空きフラグメント、*number-of* 個の完全サイズの空きブロック、および *percent* のフラグメントに分割したものです。

```
***** FILE SYSTEM WAS MODIFIED *****
```

上記のメッセージは、ファイルシステムが `fsck` によって変更されたことを示します。このファイルシステムがマウントされているか、現在のルート (/) ファイルシステムの場合はリポートします。ファイルシステムがマウントされている場合は、マウント解除して再び `fsck` を実行する必要があります。そうしないと、`fsck` によって実行された処理がテーブルのインコアコピー(カーネル内のコピー)によって取り消されます。

```
filename FILE SYSTEM STATE SET TO OKAY
```

上記のメッセージは、ファイルシステム *filename* に安定を示す印が付けられたことを示します。 `-m` オプションを指定して `fsck` を実行すると、この情報を使用して、ファイルシステムに検査が必要かどうか判断されます。

```
filename FILE SYSTEM STATE NOT SET TO OKAY
```

上記のメッセージは、ファイルシステム *filename* に安定を示す印が付けられなかったことを示します。 `-m` オプションを指定して `fsck` を実行すると、この情報を使用して、ファイルシステムに検査が必要かどうか判断されます。

第 33 章

ソフトウェアパッケージで発生する問題の解決 (手順)

この章では、ソフトウェアパッケージをインストールまたは削除するときに発生する問題について説明します。この章には、2つの節があります。「特定のソフトウェアパッケージのインストールエラー」では、パッケージのインストールエラーと管理エラーについて説明します。「一般的なソフトウェアパッケージのインストールに関する問題」では、特定のエラーメッセージを出さない障害について説明します。

この章の内容は次のとおりです。

- 464 ページの「特定のソフトウェアパッケージのインストールエラー」
- 465 ページの「一般的なソフトウェアパッケージのインストールに関する問題」

ソフトウェアパッケージの管理については、『Solaris のシステム管理 (基本編)』の「ソフトウェアの管理 (概要)」を参照してください。

ソフトウェアパッケージのシンボリックリンクに関する問題の解決

以前の Solaris リリースでは、ソフトウェアパッケージを作成するときに、シンボリックリンクのリンク先を pkgmap ファイルに指定できませんでした。そのため、pkgadd コマンドでパッケージを追加する場合、パッケージまたはパッチ関連のシンボリックリンクには、シンボリックリンクのリンク先ではなくシンボリックリンクのリンク元が使用されていました。したがって、パッケージやパッチパッケージのアップグレードの際にシンボリックリンクのリンク先を別のものに変更する必要があるという問題がありました。

現在のリリースでは、デフォルトの動作では、パッケージのシンボリックリンクのリンク先を変更する必要がある場合、シンボリックリンクのリンク元の代わりにリンク先が pkgadd コマンドによって調べられるようになりました。

しかしながら、パッケージによっては pkgadd のこの新しい動作に準拠していない場合があります。

pkgadd シンボリックリンクの新旧動作に対応するために PKG_NONABI_SYMLINKS 環境変数が使用できます。この環境変数が真に設定されていると、pkgadd はシンボリックリンクのリンク元を使用します。

pkgadd コマンドを使ってパッケージを追加する前に管理者がこの変数を設定すれば、新しい動作に対応していないパッケージを以前の動作で処理できます。

pkgadd コマンドを使って既存のパッケージを追加する場合、pkgadd シンボリックリンクの新しい動作が原因でパッケージを追加できないことがあります。その場合には、次のエラーメッセージが表示されます。

```
unable to create symbolic link to <path>
```

この問題のためにパッケージをインストールできない場合は、次の手順に従います。

1. Sun 提供のパッケージの場合は、ご購入先に新動作に対応していないパッケージ名をお知らせください。
2. PKG_NONABI_SYMLINKS 環境変数を設定し、pkgadd コマンドを使ってパッケージを再び追加してください。

```
# PKG_NONABI_SYMLINKS=true  
# export PKG_NONABI_SYMLINKS  
# pkgadd pkg-name
```

特定のソフトウェアパッケージのインストールエラー

```
WARNING: filename <not present on Read Only file system>
```

エラーの原因	問題の解決方法
このエラーメッセージは、パッケージの一部のファイルがインストールできなかったことを示す。このエラーは、通常、pkgadd を使用してパッケージをクライアントにインストールするときに発生する。この場合、pkgadd は、サーバーからマウントしているファイルシステムにパッケージをインストールしようとする。しかし pkgadd は、そのためのアクセス権を持っていない	パッケージのインストール中にこの警告メッセージが表示された場合、パッケージをサーバーにもインストールしなければならない。詳細については、『Solaris のシステム管理 (基本編)』の「ソフトウェアの管理 (概要)」を参照してください。

一般的なソフトウェアパッケージのインストールに関する問題

エラーの原因	問題の解決方法
Solaris 5.9 およびその互換バージョンより前に開発された一部のパッケージの追加と削除に関連して、既知の問題が存在する。このようなパッケージを追加または削除すると、ユーザーとの対話中にインストールが失敗するか、ユーザーとの対話のためにプロンプトが出されるが、ユーザーの応答は無視されることがある	次の環境変数を設定して、パッケージを追加し直す。 <code>NONABI_SCRIPTS=TRUE</code>

索引

数字・記号

- (PPD), PostScript プリンタ記述, 26
- (RIP), ラスターイメージプロセッサ, 26
- 64 ビット Solaris
 - 機能が有効かどうかを決定, 233
 - システムが実行できるかどうかを決定, 232

A

- accept コマンド, 109
- acct.h 形式ファイル, 309
- acctcms コマンド, 302, 313
- acctcom コマンド, 309
- acctcon コマンド, 296, 302, 311
- acctdusg コマンド, 290, 306, 311
- acctprc コマンド, 302
- acctwtmp コマンド, 289, 291, 304
- active.MMDD ファイル, 297, 311
- active ファイル, 297, 311
- ASCII ファイル, ファイル内容形式, 53
- at.deny ファイル, 282, 285
 - 説明, 270
- atjobs ディレクトリ, 273
 - 説明, 270
- atq コマンド, 283
- at コマンド, 281, 282, 285
 - l オプション (list), 284
 - l オプション (一覧表示), 283
 - m オプション (mail), 282
 - m オプション (メール), 283
 - エラーメッセージ, 285
 - 概要, 270, 271, 281

- at コマンド (続き)
 - 自動スケジュール, 273
 - 使用制御, 282
 - 使用の拒否, 285
 - 使用の制御, 285
 - の制御
 - 概要, 270
- at ジョブファイル, 281, 284
 - 位置, 271
 - 検査, 283
 - 削除, 284
 - 作成, 282, 283
 - 説明, 271
 - 登録, 281
 - 表示, 283, 284
 - 待ち行列の表示, 283

B

- banner オプション, 97

C

- cancel コマンド, 111
- chargefee スクリプト, 290, 291, 306
 - ユーザーへの課金, 295
- ckpacct スクリプト, 291, 292, 294
- closewtmp コマンド, 302
- cmsprev ファイル, 312
- consadm コマンド, 386

- consadm コマンド (続き)
 - 補助コンソールのリストを表示する (方法), 387
 - 補助コンソールを無効にする, 388
 - 補助コンソールを有効にする, 386
 - システムリブート後, 387-388
- coreadm コマンド, 390
 - グローバルコアファイルパスを有効にする, 393
 - コアダンプ構成の表示, 392
 - コアファイルの管理, 390
 - コアファイル名パターンの設定, 393
 - プロセス別コアファイルパスを有効にする, 393
- core ファイル
 - proc ツールを使用した検査, 394
 - 検索と削除, 250
- core ファイル, 自動削除, 281
- CPU (中央演算装置)
 - 情報の表示
 - 時間の使用率, 326, 339
 - 使用率が高いプロセス, 339
- CPU (中央処理装置), 情報の表示, 306
- cron.allow ファイル, 279, 280
- cron.allow ファイル, 278
- cron.deny ファイル, 278, 279
- crontab コマンド, 279
 - cron デーモンと, 273
 - e オプション (edit), 275
 - e オプション (編集), 274
 - l オプション (list), 276
 - r オプション (remove), 277
 - r オプション (削除), 278
 - /var/adm の保守, 381
 - アカウントティングスクリプトの実行, 292, 294
 - アクセスの制御
 - 概要, 270
 - エラーメッセージ, 280
 - 概要, 269, 270
 - 使用されるファイル, 273
 - 使用の制御, 278, 279, 280
 - アクセスを特定のユーザーに制限, 279
 - 概要, 278, 279
 - 使用の拒否, 278, 279
 - 使用を特定のユーザーに制限, 280
 - スケジューリング, 273
 - 変更を保存しないで終了, 275

- crontab コマンド (続き)
 - 毎日のタスク, 270
- crontab ファイル
 - 位置, 272
 - 構文, 273, 274
 - 削除, 277, 278
 - 作成, 274, 275
 - 使用の拒否, 279
 - 説明, 273
 - 存在の確認, 275
 - 表示, 276
 - 編集, 274, 275
- cron デーモン, 271, 273
- ctacct.MMDD ファイル, 302, 311
- ctmp ファイル, 311

D

- date コマンド
 - アカウントティングデータ, 291
 - アカウントティングデータと, 289
 - 説明された, 231
- daytacct ファイル
 - runacct スクリプトと, 302, 313
 - /var/adm/acct/nite ディレクトリ、存在する, 312
 - 日次利用状況レポートと, 306
- df コマンド, 349
 - F オプション (アンマウントされたファイルシステム), 240
 - h オプション, 241
 - k オプション (K バイト), 240, 349
- df コマンド, -k オプション (キロバイト), 349
- df コマンド
 - t オプション (合計ブロック), 241
 - t オプション (合計ブロック数), 240
 - 概要, 239, 349
 - 例, 240, 349
- disable コマンド, 107
- disable コマンド、例, 108
- diskacct.MMDD ファイル, 302
- diskacct ファイル, 291
- diskacct ファイル, 290, 302, 311
- dispadm コマンド, 概要, 333
- dmesg コマンド, 381
- DNLC 機能の向上, 317
- dodisk スクリプト, 289

dodisk スクリプト (続き)
概要, 289, 290, 291
作成されたファイル, 290, 291, 302
作成されるファイル, 311
実行するcrontab エントリ, 294
注意, 290
dtmp ファイル, 311
dumpadm
クラッシュダンプ情報の表示, 399
クラッシュダンプ情報の変更, 399
システムクラッシュ情報の管理, 397
du コマンド, 244, 246
ディレクトリサイズの表示, 245-246

E

edquota コマンド
-p オプション (プロトタイプ), 259
-t オプション (期間制限), 265
概要, 255, 256, 264
各ユーザーのディスク割り当て制限を変更, 265
特定のユーザーのディスク割り当て制限を無効にする, 266, 267
ユーザーディスク割り当て制限の設定, 258, 259
enable コマンド, 107
enable コマンド、例, 108
/etc/acct/holidays ファイル, 294, 306
/etc/cron.d/at.deny ファイル, 282, 285
/etc/cron.d/cron.allow ファイル, 278, 279, 280
/etc/cron.d/cron.deny ファイル, 278, 279
/etc/init.d/acct ファイル, 293
/etc/init.d/perf ファイル, 368, 371
/etc/inittab ファイル, 208
/etc/lp/classes/printer-class ファイル, 99
/etc/lp/default ファイル, 95
/etc/lp/fd ディレクトリ, 127, 183
/etc/lp/filter.table ファイル, 127
削除されたフィルタ, 129
追加されたフィルタ, 128
/etc/lp/forms/form-name/describe ファイル, 133
/etc/lp/forms/form-name ファイル, 133
/etc/lp/forms ディレクトリ, 132

/etc/lp/printers ディレクトリ, 174
印刷クライアント, 89
プリンタサーバー, 90
/etc/lp/Systems ファイル, 89
/etc/lp ディレクトリ, 173
/etc/motd ファイル, 237
/etc/printcap ファイル, 174
/etc/saf/_sactab ファイル, 208
/etc/syslog.conf ファイル, 382
/etc/utmpx ファイル, 210
/etc/vfstab ファイル, 257, 260

F

fcntl 情報, 329, 330
fd2log ファイル, 297, 301, 311
fee ファイル, 291, 295, 302, 311
find コマンド
core ファイル, 250
サイズ制限を超えるファイルの検索, 244
古いファイルまたは使用されていないファイル, 247, 248
fiscrpt*n* ファイル, 313
fsck コマンド, 270
fstat 情報, 329, 330

G

getty, 195

H

holidays ファイル, 294, 306
hostid コマンド, 231, 235

I

init プロセス, 208
iostat コマンド
-xtc オプション (拡張), 348
概要, 346
基本情報の表示, 347
ディスクの使用状況の表示, 346-348

K

kill コマンド, 332
klwp 構造体, 322
kthread 構造体, 322

L

lastdate ファイル, 302, 311
lastlogin コマンド, 302
lineuse ファイル,
「/var/adm/acct/nite/lineuse ファイル」を参照
lock1 ファイル, 302
lock ファイル, 297, 302
log.MMDD ファイル, 311
loginlog ファイル, 302, 312, 313
log ファイル, 311
lpadmin コマンド
-o banner オプション, 97
印字ホイール装着の警告の設定 (方法), 123
印字ホイールの装着 (方法), 122
印字ホイールの定義, 120
デフォルトプリンタの設定 (方法), 95
バナーページを選択可能にする (方法), 96
フォームの装着に関する警告の設定 (方法), 135
フォームの装着 (方法), 134
フォームの取り外し (方法), 134
フォームへのプリンタアクセスの制限, 139
フォントカートリッジの装着 (方法), 122
フォントカートリッジの定義, 120
プリンタクラスの定義 (方法), 99
プリンタの障害回復の設定 (方法), 102
プリンタの障害警告の設定 (方法), 100
プリンタ備考欄の追加 (方法), 93
プリンタへのアクセスの制限 (方法), 104
プリンタポート特性の調整 (方法), 146
LPDEST 環境変数, 95
lpfilter コマンド, 127
lpsched
デーモン (リファレンス), 188
ログファイル, 184
lpsched デーモン, 181, 183
LP 印刷サービス
印字ホイールの確認, 120
インタフェースプログラム, 183
カスタマイズ, 169

LP 印刷サービス (続き)

基本機能のチェック, 427-430, 430
構成ファイル, 173-174
構造, 172-180
参照情報, 172-189
使用するファイル, 175
スケジューラ, 181
定義, 172
ディレクトリ, 172
デーモン, 175
ハングした LP コマンド, 435
フォームの追跡 (概要), 131-132
プリンタ特性の定義, 45
問題の解決, 420, 423
ログファイル, 421
LP コマンド, 435
LP プリントサービス, ログファイル, 175
ls コマンド, 242, 243, 247, 248
checking directory sizes, 242
-l オプション (バイト数単位のサイズ), 243
-s オプション (ブロック数単位のサイズ), 243
-t オプション (最新のファイル), 247
LWP (軽量プロセス)
構造体, 322
情報の表示, 329
定義, 321
プロセス, 321, 322

M

mdb ユーティリティ, 401
messages.n ファイル, 381
messages ファイル, 378, 382
monacct スクリプト
runacct スクリプト, 292
runacct スクリプトと, 301
月次コマンドの要約, 308
月次コマンド要約と, 306
実行スケジューラ, 292
実行する crontab エントリ, 294
使用または生成されるファイル, 313
利用または生成されるファイル, 313
MOTD (その日のメッセージ) 機能, 237
motd ファイル, 237

N

nice コマンド, 337, 339
nice 値, 326, 337
nlsadmin コマンド, 212-213

O

owtmpx ファイル, 312

P

pacctn ファイル
 概要, 291, 302, 311
 サイズの監視, 291, 301
 表示, 309
panic メッセージ, 380
perf ファイル, 368
pfiles コマンド, 329, 330
pflags コマンド, 329
pkill コマンド, 329
pldd コマンド, 329
pmadm コマンド
 ttymon サービスの追加, 216
 ttymon サービスの表示, 217
 ttymon サービスを無効にする, 219
 ttymon サービスを有効にする, 219
pmadm コマンド, 説明, 208
pmap コマンド, 329
PostScript 以外のプリンタ, 54, 118, 120
PostScript フォント, 140
 インストール (方法), 142-143
PostScript プリンタ, 118
 デフォルトの印刷フィルタ, 127
 ファイル内容形式, 53
 プリンタタイプ, 52
 文字セット, 119
PostScript プリンタ記述, (PPD), 26
prdaily スクリプト
 runacct スクリプトと, 301, 313
 回線利用状況レポートと, 313
 概要, 301
 使用されるファイル, 311, 312
.printers ファイル, 設定 (方法), 72
PRINTER 環境変数, 95

priocntl コマンド

-c オプション (スケジューリングクラスの指定), 334, 336
-e オプション (実行), 334
-i オプション (ID タイプ), 335, 336
-l オプション (スケジューリングクラスの表示), 333, 334
-m オプション (最大/最小の優先順位), 334, 335
-p オプション (優先順位の指定), 334, 335
-s オプション (優先順位の上限/優先順位の変更), 335, 336
概要, 333

PROCFS (プロセスファイルシステム), 328

proc 構造体, 322

proc 構造体, 326

proc ツール, コアファイルの検査, 394

/proc ディレクトリ, 328

prtconf コマンド, 231, 235

psig コマンド, 329

PSR プリンタタイプ, 52

pstack コマンド, 329

ps コマンド, 326, 328

-c オプション (スケジューリングクラス), 326, 339

-ecl オプション (グローバル優先順位), 334

-ef オプション (完全な情報), 328

-ef オプション (詳細情報), 327

概要, 326

報告されるフィールド, 327

PS プリンタタイプ, 52

ptacctn.MMDD ファイル, 303

ptime コマンド, 329

ptree コマンド, 329, 330

pwait コマンド, 329

pwdx コマンド, 329, 330

Q

quotacheck コマンド, 255, 256, 260

quotaon コマンド, 256, 261

quotas ファイル, 257, 258

quotas ファイル, 255

quota コマンド, 255, 261, 262

quot コマンド, 246

R

reboots ファイル, 302, 311
reject コマンド, 109
repquota コマンド, 261, 263
residentfonts ファイル, 141
Return 設定、プリンタ, 423
Reverse PostScript プリンタ
 ファイル内容形式, 54
 プリンタタイプ, 52, 54
rm コマンド, 249, 250
root の crontab ファイル, 290
root ファイル, 272
rprrt.MMDD ファイル, 292, 313
rpt.MMDD ファイル, 302, 312
RS-232-C, 「シリアルポート」を参照
runacct スクリプト, 301
 monacct スクリプトと, 301
 prdaily スクリプトと, 301, 313
 エラー保護, 301, 302
 エラーメッセージ, 297
 概要, 291
 壊れたファイルの修復, 296, 302
 壊れたファイルの復元, 296
 再起動, 297, 298, 302, 303
 最後に実行された時間, 311
 実行スケジュール, 292
 実行する crontab エントリ, 301
 失敗, 297
 状態, 302
 使用または生成されるファイル, 311, 313
 診断ファイル, 301
 進捗ファイル, 301
 ユーザー料金の計算と, 295, 306

S

sa1 コマンド, 368
sa2 コマンド, 368, 369
SAC, 「サービスアクセスコントローラ」を参照
sacadm コマンド
 ttymon ポートモニターの起動, 214
 ttymon ポートモニターの終了, 214
 ttymon ポートモニターの追加, 213
sacadm コマンド
 ttymon ポートモニターを有効にする, 215
 説明, 207

sadc コマンド, 368, 369, 371
sadd ファイル, 369
SAF, 「サービスアクセス機能 (SAF)」を参照
sar コマンド, 350, 369
 -A オプション (全体のパフォーマンス), 368, 370
 -a オプション (ファイルアクセス), 350, 351
 -b オプション (バッファ), 351, 352
 -c オプション (システムコール), 353
 -d オプション (ブロックデバイス), 354
 -e オプション (終了時刻), 370
 -f オプション (データを抽出するファイル), 370
 -g オプション (ページアウト/メモリの解放), 355
 -i オプション (間隔), 370
 -k オプション (カーネルメモリ), 356
 -m オプション (プロセス間通信), 358
 -p オプション (ページイン/ページフォルト), 359
 -q オプション (待ち行列), 360, 361
 -r オプション (未使用メモリ), 361
 -s オプション (開始時刻), 370
 -u オプション (CPU 使用率), 362
 -v オプション (システムテーブル), 364
 -w オプション (ボリュームのスワップと切り替え), 365
 -y オプション (端末デバイス), 366
 オプションのリスト, 370
 概要, 350, 369
 すべてのオプションの説明, 370
shutacct スクリプト, 291, 292
shutdown コマンド, 292
simple ファイル内容形式, 53
Solaris と lpd をベースにした印刷システム, 45
Solaris プリンタマネージャ
 概要, 32-33
 前提条件, 61
 プリンタアクセスを追加, 70
 プリンタの新規追加 (方法), 64-66, 66-69, 77-79
 プリンタの定義機能 (概要), 48
Spacctn.MMDD ファイル, 302, 311
startup コマンド, 290
statefile ファイル, 297, 302, 311
stty 設定
 カスタマイズ, 152
 推奨事項, 434

stty 設定 (続き)
デフォルト, 145, 434
問題の解決, 422-423
sys crontab, 369
syslog.conf ファイル, 382
syslogd デーモン, 381
sys ファイル, 272

T

tacct.MMDD ファイル, 297, 302, 312
tacctn ファイル, 313
tacctprev ファイル, 312
tacct ファイル, 297, 302, 312, 313
terminfo エントリ
サポートされていないプリンタのものを追加
(方法), 150
選択可能文字セット, 118
terminfo エントリ
追加 (概要), 147-150
必要項目のリスト, 147
terminfo データベース, 文字セット名, 118
/tmp/disktacct.MMDD ファイル, 302
tmpwtmp ファイル, 302, 311, 313
/tmp ディレクトリのクリア, 249, 250
TOTAL REAL-MIN 列 (日次コマンドの要約)
, 307
total コマンドの要約, 308
TranScript フィルタ, 127
troff フィルタによる PostScript への変
換, 166
ttyadm コマンド, 212
ttypmon サービス
追加, 216
表示, 217
無効にする, 219
有効にする, 219
ttypmon ポートモニター
起動, 214
機能の概要, 208
終了, 214
図, 209
追加, 213
発着信両用サービス, 210
有効にする, 215
tty 回線
使用状況の監視, 304

tty 回線
使用状況の監視, 289
tty 回線
不良回線の問題の解決, 305
tty 回線
不良回線の問題の解決, 305
利用状況の監視, 305, 313
turnacct switch スクリプト, 291
turnacct 切り換えスクリプト, 302

U

UFS ファイルシステム
情報の表示, 246
ユーザーの所有権, 246
uname コマンド, 234
UNIX システム (クラッシュ情報), 396
user 構造体, 322
/usr/adm/messages ファイル, 378
/usr/bin/mdb ユーティリティ, 401
/usr/lib/acct/startup コマンド, 290
/usr/lib/lp/model ディレクトリ, 183
/usr/lib/lp/postscript ディレク
トリー, 126, 183
/usr/lib/lp ディレクトリ, 127
/usr/proc/bin ディレクトリ, 328, 329
utmp2wtmp コマンド, 302

V

/var/adm/acct/fiscal ディレクトリ, 311
/var/adm/acct/nite/active.MMDD
ファイル, 301, 311
/var/adm/acct/nite/active ファイ
ル, 297, 301, 311
/var/adm/acct/nite/cms ファイル, 302
/var/adm/acct/nite/cms ファイル, 311
/var/adm/acct/nite/ctacct.MMDD
ファイル, 302, 311
/var/adm/acct/nite/ctmp ファイル, 311
/var/adm/acct/nite/daycms ファイ
ル, 302, 311, 313
/var/adm/acct/nite/daytacct ファイル,
「daytacct ファイル」を参照
/var/adm/acct/nite/disktacct.MMDD
ファイル, 302

/var/adm/acct/nite/disktacct ファイル, 291
 /var/adm/acct/nite/disktacct ファイル, 290, 291, 302, 311
 /var/adm/acct/nite/fd2log ファイル, 297, 301, 311
 /var/adm/acct/nite/lastdate ファイル, 302, 311
 /var/adm/acct/nite/lineuse ファイル, 302, 311, 313
 /var/adm/acct/nite/lock1 ファイル, 302
 /var/adm/acct/nite/lock ファイル, 297, 302, 311
 /var/adm/acct/nite/log.MMDD ファイル, 311
 /var/adm/acct/nite/log ファイル, 311
 /var/adm/acct/nite/owtmpx ファイル, 312
 /var/adm/acct/nite/reboots ファイル, 302, 311
 /var/adm/acct/nite/statefile ファイル, 297, 302, 311
 /var/adm/acct/nite/tmpwtmp ファイル, 302, 311, 313
 /var/adm/acct/nite/wtmp.MMDD ファイル, 302, 312
 /var/adm/acct/nite/wtmperror.MMDD ファイル, 311
 /var/adm/acct/nite/wtmperror ファイル, 311
 /var/adm/acct/nite ディレクトリ, 311
 /var/adm/acct/sum/cmsprev ファイル, 312
 /var/adm/acct/sum/cms ファイル, 302
 /var/adm/acct/sum/cms ファイル, 312, 313
 /var/adm/acct/sum/daycms ファイル, 302, 312, 313
 /var/adm/acct/sum/loginlog ファイル, 302, 312, 313
 /var/adm/acct/sum/rprt.MMDD ファイル, 313
 /var/adm/acct/sum/rprtMMDD ファイル, 292
 /var/adm/acct/sum/rpt.MMDD ファイル, 302
 /var/adm/acct/sum/tacct.MMDD ファイル, 302, 312
 /var/adm/acct/sum/tacctMMDD ファイル, 297
 /var/adm/acct/sum/tacctprev ファイル, 297, 312
 /var/adm/acct/sum/tacct ファイル, 297, 302, 312, 313
 /var/adm/acct/sum ディレクトリ, 312
 /var/adm/acct/sum ディレクトリ, 292, 311
 /var/adm/acct ディレクトリ, 311
 /var/adm/dtmp ファイル, 311
 /var/adm/fee ファイル, 291, 295, 302, 311
 /var/adm/messages.n ファイル, 381
 /var/adm/messages ファイル, 378, 382
 /var/adm/sa/sadd ファイル, 369
 /var/adm/Spacctn.MMDD ファイル, 302, 311
 /var/adm ディレクトリ
 raw アカウンティングデータ, 290
 サイズの制御, 249
 説明された, 311
 /var/lp/logs/lpsched ファイル, 183, 421
 /var/lp/logs ディレクトリ, 184
 /var/spool/cron/atjobs ディレクトリ, 270, 271, 273
 /var/spool/cron/crontabs/adm ファイル, 272
 /var/spool/cron/crontabs/lp ファイル, 272
 /var/spool/cron/crontabs/lp ファイル, 184
 /var/spool/cron/crontabs/root ファイル, 272, 290
 /var/spool/cron/crontabs/sys crontab, 369
 /var/spool/cron/crontabs/sys ファイル, 272
 /var/spool/cron/crontabs ディレクトリ, 273
 /var/spool/cron/crontab ディレクトリ, 272
 /var/spool/lp/requests ディレクトリ, 176
 /var/spool/lp/tmp ディレクトリ, 176
 /var/spool/lp ディレクトリ, 46, 179
 /var/spool ディレクトリ, 249, 250
 /var/tmp ディレクトリ, 249, 250
 /var パーティション, 46

vfstab ファイルとディスク割り当て制限, 257, 258, 260
vmstat コマンド
-c オプション (キャッシュフラッシュ), 345
-i オプション (割り込み), 346
-s オプション (システムイベント), 344
-S オプション (スワッピング), 345
概要, 342
仮想メモリの表示, 343-344
レポートのフィールド, 342

W

wtmp.MMDD ファイル, 302, 312
wtmperror.MMDD ファイル, 311
wtmperror ファイル, 311
wtmpfix コマンド, 296, 302, 311
wtmpx ファイル
概要, 291, 296, 302
壊れた、修復, 302
壊れたファイルの修復, 296
シャットダウンと, 292
日次レポートと, 304

あ

アカウントティング, 298, 313
「ユーザーへの課金」も参照
raw データ, 290
維持, 296
概要, 288
起動, 294
壊れたファイルの修復
tacct ファイル, 297
wtmpx ファイル, 296, 302
自動実行のための設定 (方法), 293
使用不可, 299
接続, 289
runacct 状態と, 302
/var/adm/acct/nite ディレクトリと, 311
/var/adm/wtmpx ファイル, 305
タイプ, 295
停止, 298-299
ディスク, 289, 290, 291
acctdusg プログラム, 306

アカウントティング (続き)
日次, 290, 313
「アカウントティング、報告」も参照
手順の要約, 292
ファイル, 311, 313
プロセス, 289, 291, 305, 306
保守, 298
ユーザー料金の計算, 290
「ユーザーへの課金」も参照
レポート, 303, 308
概要, 303
合計コマンド要約 (月次), 312, 313
最後のログインレポート, 308
日次コマンド要約, 306, 313
日次利用状況レポート, 305, 306
日次レポート (tty 回線の使用状況), 304
日次レポート (tty 回線の利用状況), 305
アクセス
フォーム (方法)
プリンタの制限, 139
ユーザーの制限, 138-139
アクティブなファイル, 301
アスタリスク (*), crontab ファイル, 273
アダプタボード (シリアルポート), 195
アドレス空間マップ, 329
アプリケーションスレッド, 321, 323

い

一時ディレクトリ, 247, 249, 250
位置揃えパターン
印刷, 135
定義, 168
保護, 132
一覧表示
実行されているプロセス, 328
ファイルとディレクトリ, 242, 243, 247, 248
印刷
状態メッセージ, 438
スプーリングディレクトリ, 179-180
停止 (概要), 107
特殊モード, 157
バナーページ, 95-96
プリンタの受け付けと拒否の設定 (概要)
, 109
ユーザー料金の計算, 295
リモート, 187-189

- 印刷 (続き)
 - ローカル (図), 184
- 印刷クライアント
 - 構成の確認, 428
 - ジョブの解放, 438
 - 定義, 44
 - プリンタへのアクセスの削除 (方法), 88
- 印刷構成
 - Solaris と lpd をベースにした印刷システムの使用, 45
 - 集中化, 44
- 印刷スケジューラ
 - LP システムファイルの更新, 181
 - 起動 (方法), 93
 - 再起動 (方法), 92
 - 停止 (方法), 92-93
 - 動作していない場合, 427
- 印刷デーモン, 175
- 印刷フィルタ
 - PostScript, 127
 - TranScript, 127
 - troff から PostScript への変換, 166
 - 印刷要求の保留, 158
 - オプションのキーワード, 162
 - オプションを定義するテンプレート, 162-165
 - 管理 (概要), 126
 - 高速 (概要), 141
 - 削除 (概要), 126-128
 - 削除 (方法), 128-129
 - 作成 (概要), 126, 156-166
 - 作成 (方法), 165-166
 - 種類, 156
 - 情報の表示 (方法), 129-130
 - ダウンロード (概要), 141
 - 追加 (概要), 126-128
 - 追加 (方法), 128
 - 定義, 126, 183
 - 定義の作成, 159-165
 - 低速 (概要), 141
 - 特殊モードの処理, 157
 - 特性, 163
 - バイパス, 432, 433
 - ファイル内容形式の変換, 126
 - ファイルの変換, 157
 - 復元 (概要), 126-128
 - プリンタ障害からの回復, 102
 - プリンタ障害からの回復 (概要), 101
- 印刷フィルタ (続き)
 - 変更 (概要), 126-128
 - 要件, 158-159
- 印刷フォーム, 「フォーム」を参照
- 印刷待ち行列, ログ, 176
- 印刷待ち行列用のディスク容量, 46
- 印刷要求
 - ID, 95, 106, 111
 - 印刷要求の状態, 107
 - ID (方法), 112, 115
 - 印刷要求の移動, 114
 - 受け付け (方法), 110
 - 拒否 (方法), 89, 110
 - 状態のチェック (方法), 106-107
 - スケジューリング, 182
 - 他のプリンタへの移動 (概要), 113
 - 他のプリンタへの移動 (方法), 113
 - 取り消し (概要), 111
 - 取り消し (方法), 111
 - 特定ユーザー, 112
 - プリンタを使用不可にする, 108
 - 保留状態 (概要), 115
 - 待ち行列の先頭に移動 (概要), 115
 - 優先順位の変更 (概要), 105, 115
 - 優先順位の変更 (方法), 115
 - ログ, 177
 - ログファイルから消去, 184
- 印刷要求の移動 (概要), 113
- 印刷要求の移動 (方法), 113
- 印刷要求の受け付け (概要), 109
- 印刷要求の受け付け (方法), 110
- 印刷要求の拒否 (概要), 109
- 印刷要求の拒否 (方法), 89, 110
- 印刷要求の優先順位, 105, 115
- 印字ホイール, 119
 - 確認, 120
 - 装着 (方法), 121-122
 - 装着を促す警告, 123-124
 - 定義 (方法), 120-121
 - 取り外し (方法), 121-122
 - 命名, 120
- インストール
 - PostScript フォント (方法), 142-143
 - ホスト常駐 PostScript フォント (方法), 143
- インタフェースプログラム (プリンタ)
 - カスタマイズ (概要), 151-155
 - カスタマイズ (方法), 154-155
 - 標準, 155

う

ウォッチドッグリセットメッセージ, 380

え

エラーメッセージ

- at コマンド, 285
- crontab コマンド, 280
- runacct スクリプト, 297
- 格納場所の指定, 381, 382
- クラッシュメッセージ, 380, 381
- 送信元, 382
- 優先順位, 383
- ログのカスタマイズ, 382
- ログファイル, 378, 381

お

オペレーティングシステム, 情報の表示, 234

か

カーネルスレッド

- 構造体, 322, 326
- スケジューリングと, 326

回線, 不良回線の問題の解決, 305

回線制御, 209

回線の使用状況, 日次レポートと, 304

回線の使用率, 接続アカウントと, 289

回線の利用状況,

- /var/adm/acct/nite/lineuse ファイル, 313

回線の利用状況の監視, 305

書き込みシステムコール, 統計情報, 307

確認

- crontab ファイルの存在, 275
- 割り当て制限, 255, 261, 265

カスタマイズ

- LP 印刷サービス, 169
- stty モード, 152
- システムメッセージ記録 (方法), 383-384
- システムメッセージのログ, 382
- 終了コード、プリンタ, 152-153
- プリンタインタフェースプログラム (概要), 151-155

カスタマイズ (続き)

- プリンタインタフェースプログラム (方法), 154-155

カスタマサービス, クラッシュ情報の送信, 378

環境変数

- LPDEST, 95
- PRINTER, 95

管理

- 印刷フィルタ (概要), 126
- フォーム, 130-139, 139
- フォント (概要), 140-143
- 文字セット, 118-126

き

期間, 弱い制限, 265

起動

- Solaris プリンタマネージャ (方法), 61-63
- 印刷スケジューラ (方法), 93

共有メモリー, プロセス仮想メモリー, 323

許可リスト

- フォームへのユーザーアクセス, 138
- プリンタへのユーザーアクセス (概要), 103

拒否リスト

- フォームへのユーザーアクセス, 138
- プリンタへのユーザーアクセス (概要), 103

キロバイト, ファイルシステムのディスク使用, 349

く

クラス (プリンタ), 98

- lpadmin コマンドによる定義 (方法), 99
- 使用可能または使用不可にできない, 108
- 状態のチェック, 106

クラッシュ, 382, 410

- カスタマサービス, 378, 397
- クラッシュ後のリポートの失敗, 405-407
- クラッシュダンプ情報の保存, 396
- クラッシュダンプの検査, 401
- クラッシュダンプファイルの削除, 250
- 生成されたシステム情報の表示, 380, 401
- 他のシステム情報の保存, 381
- 手順, 378, 410

クラッシュダンプ

- 保存を無効にする, 402

クラッシュダンプ (続き)
保存を有効にする, 402
クラッシュダンプ構成
dumpadm を使用した表示, 399
dumpadm を使用した変更, 399
クラッシュダンプ情報の保存, 397
クラッシュダンプディレクトリ, 一杯になった
クラッシュダンプディレクトリからの回復, 402
繰り返すシステムタスク, 279
グローバルコアファイルの設定, coreadm を使用した設定, 390
グローバル優先順位
定義, 333
表示, 334

け

警告
フォームの装着 (概要), 132
フォームの装着 (方法), 135-136
プリンタの障害 (方法), 100
文字セットの装着, 123-124
警告メッセージの優先順位 (syslogd の), 383
月次コマンドの要約, 308
検査, at ジョブ, 283
検索
大きなファイル, 243
サイズ制限を超えるファイル, 244
サイズ制限を超えるファイルの検索, 244
大規模ファイル, 243
古いファイルまたは使用されていないファイルの削除
「削除」を参照
検索パス, 設定ファイル, 416

こ

コアダンプ構成, coreadm を使用した表示, 392
コアファイル, coreadm を使用した管理, 390
コアファイルの検査, proc ツールを使用した, 394
コアファイル名パターン, coreadm を使用した設定, 391
合計コマンド要約, 312

構成

プリンタポート, 50
x86 システム用の, 50
高速印刷フィルタ (概要), 141
構文 (crontab ファイル), 273, 274
コマンド, 利用状況の監視, 311
「コマンドが見つかりません」というメッセージ, 415
コンソール
補助
システムリブート後も有効にする, 387-388

さ

サービスアクセス機能
使用するとき, 196
説明, 196
サービスアクセス機能 (SAF)
概要, 196, 206
関連ファイル (表), 220
関連プログラム (表), 206
使用, 206
制御されるサービス, 222
設定, 196
サービスアクセスコントローラ, 207
再開, プロセス, 329
再起動
runacct スクリプト, 297, 298, 302, 303
印刷スケジューラ (方法), 92
最後のログインレポート, 308
最小
nice 値, 337
優先順位, 334
サイズ
ディレクトリ, 244, 246
du コマンド, 245-246
ファイル, 242, 243, 244, 246
最大
nice 値, 337
最大サイズを超えるファイルの検索, 244
優先順位, 334
最大値, pacct ファイルのサイズ, 289
削除
at ジョブ, 284
core ファイル, 250
crontab ファイル, 277, 278

- 削除 (続き)
 - 一時ファイル, 249, 250
 - 印刷フィルタ (方法), 128-129
 - クラッシュダンプファイル, 250
 - フォーム (方法), 133-134
 - プリンタへのアクセス (方法), 88-90
 - 古いファイルまたは使用されていないファイル, 270
 - 古いファイルまたは使用されていないファイルの検索と削除, 247
 - ログファイル, 275
 - 作成
 - at ジョブ, 282
 - at ジョブ, 283
 - crontab ファイル, 274, 275
 - 印刷フィルタ (概要), 156-166
 - 印刷フィルタ (方法), 165-166
 - フォーム, 167-169
 - フォーム定義 (方法), 169
 - サポートされていないプリンタ, 55, 147-150
- し
- シェル, 307
 - 時間
 - CPU 時間が大量に増えているプロセス, 339
 - CPU 使用, 306
 - CPU 使用率, 326, 339
 - 表示, 231
 - 変更, 236, 237
 - 弱い制限, 264
 - 時刻, 表示, 236
 - システムクラッシュ情報, dumpadm を使用した管理, 397
 - システムクラッシュ情報の管理, dumpadm を使用した, 397
 - システム資源
 - アカウントिंग
 - 概要, 288
 - 概要, 321
 - 監視, 282, 410
 - アカウントिंग, 298
 - アカウントिंगシステム, 313
 - クラッシュ, 382
 - 自動的な, 282
 - ディスク割り当て制限, 263
 - プリンタサーバーの割り当て, 45
 - システムタスク
 - 「crontab コマンド、at コマンド」も参照
 - 自動的なスケジュール, 269
 - スケジューリング
 - 1 回だけのタスク, 271, 281
 - 繰り返すタスク, 270, 271
 - システムタスクの自動実行, 1 つのタスク, 281
 - システムタスクの自動的な実行, 270
 - 繰り返すタスク, 279, 280
 - 単一のタスク, 282, 285
 - システム動作
 - データの自動収集, 368, 371
 - データの手動収集, 350, 369
 - システム動作データの自動収集, 368, 371
 - システム動作の自動報告, 368, 369
 - システムの動作, システム動作の追跡リスト, 323
 - システムメッセージ
 - 格納場所の指定, 381
 - ログのカスタマイズ (方法), 383-384
 - システムメッセージのログ (カスタマイズ), 382
 - シャットダウン
 - 監視, 291, 292, 304
 - 集中化, 印刷構成, 44
 - 修復
 - 壊れた tacct ファイル, 297
 - 壊れた wttmpx ファイル, 296
 - 終了
 - crontab コマンドの変更を保存しないで, 275
 - プログラムの強制終了, 410
 - 終了コード (プリンタインタフェース), 152-153
 - 終了コード (プリンタインタフェース)
 - 表, 152
 - 標準規格, 152
 - 障害回復 (プリンタ), 48
 - 障害追跡, プロセス, 339
 - 障害通知 (プリンタ)
 - lpadmin コマンドによる設定 (概要), 99-100
 - lpadmin コマンドによる設定 (方法), 100
 - Solaris プリンタマネージャによる設定, 48
 - 警告値, 100
 - 使用可能, プリンタ (方法), 108
 - 状態
 - 印刷要求のチェック, 106-107
 - プリンタの状態の取得 (方法), 91-92
 - 状態、(runacct スクリプト), 302

使用不可

システムアカウンティング, 299

プリンタ (方法), 89, 108

所有権, 変更する理由, 418

シリアルプリンタをシステムに追加, 49

シリアルポート

アダプタボード, 195

定義, 194

シリアルポートツール, 端末とモデム, 196

す

スーパーユーザー (ルート) パスワード, 忘れた

SPARC, 407

x86, 408

スケジューリング

「crontab コマンド、at コマンド」も参照

1 回だけのシステムタスク, 271, 281

繰り返すシステムタスク, 270, 271

スケジューリングクラス, 333

指定, 334, 335

情報の表示, 326, 333, 334

デフォルト, 334

変更, 336, 337

優先順位の変更, 335, 337

優先順位のレベル, 333

優先順位レベルと, 334, 335

優先順位を変更, 334

スプーリングディレクトリ, 印刷サービス, 179-180

せ

制御

at コマンドの使用, 282, 285

at コマンドへのアクセス, 270

crontab コマンドの使用, 280

crontab コマンドへのアクセス

概要, 270

crontab コマンドへの使用, 279

フォームへのプリンタアクセス (方法), 139

フォームへのユーザーアクセス (方法), 138-

139

プロセス, 331-332

制御がきかなくなったプロセス, 339

制限

フォームへのプリンタアクセス (方法), 139

フォームへのユーザーアクセス (方法), 138-139

整合性の確認, 260

性能

プロセス管理, 329, 337

セキュリティ

at コマンド, 282

crontab コマンド, 279

接続アカウンティング, 「アカウンティング、接続」を参照

設定

coreadm コマンドを使用したコアファイル名パターン, 393

印刷ホイールの装着を促す警告 (方法), 123-124

選択可能文字セットの別名 (方法), 124-126

選択可能文字セットの別名 (例), 125

フォントカートリッジの装着を促す警告 (方法), 123-124

文字セット (例), 125

選択可能文字セット, 118-119, 119, 124-126

選択可能文字セットの別名, 119, 124-126

そ

装着

印刷ホイール (方法), 121-122

フォーム (概要), 131

フォーム (方法), 134-135

フォントカートリッジ (方法), 121-122

その日のメッセージ (MOTD) 機能, 237

ソフトウェアパッケージ, インストールの問題の解決, 463-465

た

大規模ファイル, 243

タイムシェアリングプロセス

スケジューリングパラメータの変更, 335

優先順位, 333, 337

変更, 335, 337

ダウンロード

フォント, 433

フォント (概要), 140

ダウンロード (続き)
 ホスト常駐フォント (概要), 141-142
ダウンロードされた PostScript フォント, イン
 ストール (方法), 142-143
タブ設定, 422
端末
 英数字, 194
 回線の使用状況
 接続アカウントと, 289
 日次レポートと, 304
 回線の利用状況
 /var/adm/acct/nite/lineuse ファイ
 ル, 313
 日次レポートと, 305
 管理用ツール, 196
 シリアルポートツールの概要, 198
 シリアルポートツールの項目の説明, 198
 設定 (方法), 200-201
 端末間の相違, 194
 定義, 194
 不良回線の問題の解決, 305
 プロセス制御, 326
端末特性, 152

ち

着信専用サービス, 194
チューニング, 日次コマンド要約と, 306
直接印刷, 53

つ

追加
 terminfo エントリ (概要), 147-150
 印刷フィルタ (方法), 128
 フォーム (方法), 132-133
 プリンタ、新規 (方法)
 Solaris プリンタマネージャ, 64-66, 66-69,
 77-79
 プリンタ備考欄 (方法), 93
 リモートプリンタへのアクセス, 70
 ローカルプリンタまたは接続したプリンタ
 (方法), 65, 78
ツール
 システムパフォーマンスの監視, 324
 プロセス, 329

ツール (続き)
 プロセス情報の表示, 328

て

定義
 印刷ホイール (方法), 120-121
 フォントカートリッジ (方法), 120-121
定型タスクの自動実行 (概要), 269
停止
 印刷スケジューラ (方法), 92-93
 システムアカウント, 298-299
 プリンタ (概要), 107
 プロセスの一時的な, 329
ディスクアカウント, 「アカウント
 、ディスク」を参照
ディスクドライブ
 情報の表示, 348
 空きディスク容量, 349
 古いファイルまたは使用されていないファイ
 ルの検索と削除, 275
ディスク容量
 空き容量, 240, 349
 大きなファイルの検索, 243
 サイズ制限を超えるファイルの検索, 244
 最適化, 239-251
 情報の表示, 240
 df コマンド, 240, 349
 各ユーザーのディスク容量, 246
 ディレクトリのサイズ, 244, 246
 ファイルのサイズ, 242, 243, 244
 マウント先, 349
 ユーザーの所有権, 246
 大規模ファイルの検索, 243
 ファイルシステムの使用率, 240, 349
 古いファイルまたは使用されていないファイ
 ルの検索と削除, 247, 251
ディスク容量の最適化, 239-251
ディスク容量のユーザーの所有権, 246
ディスク割り当て, 初期設定, 256
ディスク割り当て制限, 263
 各ユーザー用に変更, 265
 管理, 263
 情報の表示, 262
 初期設定, 260
 整合性の確認, 260
 チェック, 261

ディスク割り当て制限 (続き)
特定のユーザーについて無効にする, 266, 267
ファイルシステムの確認, 263
ファイルシステムの構成, 257, 258
複数ユーザー用のプロトタイプ, 259
変更, 264
有効, 255
有効にする, 261
ユーザー, 258, 259, 262, 265, 266, 267
ユーザーディスク割り当て制限の超過の確認, 262
要件, 256
弱い期間制限
変更, 264
ディスク割り当て制限の初期設定, 260
ディスク割り当て制限を自動的に有効にする, 255
ディスク割り当ての初期設定, 256
低速印刷フィルタ, 141
ディレクトリ
一時ディレクトリのクリア, 247, 249, 250
サイズ, 244, 246
情報の表示, 242, 243, 244, 246
プロセスの現在の作業用ディレクトリ, 329
デーモン
lpsched, 181, 183
lpsched (リファレンス), 188
印刷, 175
テクニカルサポート
クラッシュ情報の送信, 378
クラッシュダンプの分析, 397
デフォルト
cron.deny ファイル, 278
crontab ファイル, 272
nice 値, 337
スケジューリングクラス, 334
その日のメッセージ, 237
弱い期間制限, 264, 265
デフォルトプリンタ
lpadmin コマンドによる設定 (概要), 94-95
lpadmin コマンドによる設定 (方法), 95
Solaris プリンタマネージャによる設定, 48
電源異常からの回復, 304
電源を切ってまた入れる, 410
転送文字, 読み取りおよび書き込みシステム
コール, 307
テンプレート (印刷フィルタ), 162-165

と
取り消し
印刷要求 (方法), 111
特定ユーザー, 112
プリンタを使用不可にする, 108
取り外し
印字ホイール (方法), 121-122
フォーム (方法), 134-135
フォントカートリッジ (方法), 121-122

に
日時
変更, 236, 237
日次アカウントティング, 「アカウントティング、日次」を参照

ね
ネットワーク, アクセスで発生する問題の把握, 418
ネットワークアクセスで発生する問題の把握, 418
ネットワークプリンタ
設定用作業マップ, 59
追加, 72-83
追加 (方法)
lp コマンドによる, 79-83
ベンダー提供ツール, 77
定義, 72

は
ハードディスク, プリンタサーバーに推奨, 47
発信専用サービス, 194
発着信両用サービス, 194, 210
バナーページ
印刷しない理由, 96
出力が正しくない場合の問題の解決, 421-422
設定, 48
lpadmin コマンドによる, 95-96
選択可能にする (方法), 96
lpadmin コマンドにより, 96
本文が印刷されない, 432

バナーページ (続き)
抑制する (方法), 97
-o banner オプションによる, 97
パフォーマンス
監視ツール, 324
システム動作の監視, 350
システム動作の監視, 323, 368
追跡される動作, 323
動作データの自動収集, 368, 371
動作データの手動収集, 350, 369
ファイルアクセス, 350, 351
プロセス管理, 321
レポート, 350
パラレルプリンタ, 49
パリティビット, プリンタ, 422

ひ

備考欄, Solaris プリンタマネージャによる設定, 48
日付, 表示, 236
表示
acct.h 形式ファイル, 309
at ジョブ, 283, 284
at 待ち行列, 283
coreadm を使用したコアダンプ構成, 392
crontab ファイル, 276
dumpadm を使用したクラッシュダンプ構成, 399
iostat を使用したディスクの使用状況, 346-348
LWP 情報, 329
pacctn ファイル, 309
vmstat を使用した仮想メモリー, 343-344
オペレーティングシステム情報, 234
クラッシュ情報, 380, 401
時間, 231
時刻, 236
システム情報, 236
コマンド, 231
システム動作情報, 350, 369
スケジューリングクラス情報, 326, 333, 334
ディスク割り当て制限情報, 262
ディレクトリの情報, 242, 243, 244
日付, 231, 236
ファイルシステム情報, 246
ファイルシステムの情報, 240, 349

表示 (続き)
ファイル情報
ファイルのサイズ, 243
ファイルのサイズ, 242-243
ファイルの情報
du コマンドの使用, 244
最新の一覧表示, 247
ファイルのサイズ, 242
ブートメッセージ, 381
フォームの状態 (方法), 137-138
プロセス, 327
プロセス情報 (方法), 330-331
ホスト ID, 231, 235
優先順位情報, 326, 334
リンクされているライブラリ, 329
割り当て制限の情報, 255, 261, 262
標準プリンタインタフェースプログラム, 155
ピン構成 (マルチモデムケーブル用), 425

ふ

ファイル
fstat と fcntl 情報の表示, 329, 330
fstat と fcntl の情報表示, 329
LP 印刷サービスが使用する, 175
アカウントティング, 311, 313
アクセス操作のチェック, 350, 351
検索パスの設定, 416
壊れた、修復
wtmptx ファイル, 302
壊れたファイルの復元, 296
サイズ, 242, 243, 244, 246
サイズ制限を超えるファイルの検索, 244
サイズの表示, 242-243
削除
「削除」を参照
情報の表示
一覧表示, 242, 243
最新の一覧表示, 247
サイズ, 242, 243, 244, 246
利用状況の監視, 289, 290, 306
ファイルシステム
情報の表示, 240, 349
ディスク容量の使用率, 240, 349
復元, 295, 306
マウントポイント, 349
ファイル内容形式, 53

- ファイル内容形式 (続き)
 - Admintool のメニュー, 53
 - PostScript, 53
 - PostScript 以外のプリンタ, 54
 - simple, 53
 - Solaris プリンタマネージャで設定, 48
 - 印刷フィルタによる変換, 126, 157
 - 共通プリンタ, 54
 - 出力が正しくない場合の問題の解決, 421-422
- フィルタ, 53
 - ダウンロード, 141
 - ダウンロード (概要), 141
 - なしに印刷, 53
- ブート
 - sadc コマンドの実行, 368
 - 生成されるメッセージの表示, 381
- フォーム
 - アクセスの制御 (概要), 132
 - 削除 (概要), 131
 - 削除 (方法), 133-134
 - 作成, 167-169
 - 状態の表示 (方法), 137-138
 - 情報の表示 (方法), 137
 - 装着 (概要), 131
 - 装着したフォームの追跡 (概要), 131-132
 - 装着に関する警告 (概要), 132
 - 装着に関する警告を設定 (方法), 135-136
 - 装着 (方法), 134-135
 - 属性の表示 (概要), 132
 - 追加 (概要), 131
 - 追加 (方法), 132-133
 - 定義、作成 (方法), 169
 - デフォルト値, 167
 - 取り外し (方法), 134-135
 - プリンタアクセスの制限 (方法), 139
 - 変更 (概要), 131
 - ユーザーアクセスの許可 (方法), 138-139
 - ユーザーアクセスの拒否 (方法), 138-139
 - ユーザーアクセスの制限 (方法), 138-139
 - 用紙 (給紙と取り外し), 134
- フォント
 - PostScript, 140
 - インストール (方法), 143
 - 種類, 118
 - 常時ダウンロード, 140
 - ダウンロード, 433
 - ダウンロード (概要), 140
- フォント (続き)
 - ダウンロードされた PostScript, 142-143
 - プリンタ常駐, 140
 - プリンタ常駐 (概要), 141
 - ホスト常駐, 140
 - ホスト常駐 (概要), 141-142, 142
- フォントカートリッジ, 119
 - 装着 (方法), 121-122
 - 装着を促す警告, 123-124
 - 定義 (方法), 120-121
 - 取り外し (方法), 121-122
 - 命名, 120
- 復元, コマンドの使用, 412
- プライベートインタフェース, 173
- フラグのトレース, 329
- プリンタ
 - PostScript, 118, 119
 - PostScript 以外, 118, 156
 - stty 設定, 152, 434
 - stty 設定値, 434
 - 印刷要求の受け付け (概要), 109
 - 印刷要求の拒否 (概要), 109
 - 印字ホイール, 118
 - インタフェースプログラム, 151-152, 152-153, 155
 - インタフェースプログラムのカスタマイズ (概要), 151-155
 - インタフェースプログラムのカスタマイズ (方法), 154-155
 - 機能の制御 (概要), 109
 - 許可リスト (概要), 103
 - 拒否リスト (概要), 103
 - クライアントアクセスの削除 (概要), 88
 - クラス, 98
 - サポートされていないものを追加 (概要), 147-150
 - 障害, 153-154, 157-158, 437
 - 障害回復 (概要), 102
 - 障害警告 (方法), 100
 - 障害通知, 99-100
 - 使用可能 (方法), 108
 - 状態, 424
 - 状態のチェック (方法), 91-92
 - 状態 (方法)
 - チェック, 106-107
 - 使用不可 (方法), 89, 108

- プリンタ (続き)
 - 新規追加 (方法)
 - Solaris プリンタマネージャによる, 64-66, 66-69, 77-79
 - 設定, 57-85, 145, 422, 434
 - Return, 423
 - タブ, 422
 - パリティ, 422
 - 設定値
 - stty, 434
 - デフォルト, 95
 - ネットワーク, 59, 72
 - ネットワークプリンタの追加 (方法)
 - lp コマンドによる, 79-83
 - ベンダー提供ツールによる, 77
 - パリティビット, 422
 - 備考欄, 追加 (方法), 93
 - ファイル内容形式, 432
 - フィルタの影響, 432
 - フォームへのアクセス (例), 133
 - フォントカートリッジ, 118
 - プリンタサーバーから削除 (方法), 88-90
 - プリンタサーバーからの削除 (方法), 89
 - ポーレート設定, 422
 - 無効化 (概要), 107
 - 問題の解決, 421-423, 423, 424, 427, 432, 435, 436, 438
 - 出力が正しくない場合, 431-435
 - プリンタタイプが正しくない場合, 431
 - 有効化 (概要), 107
 - ユーザーアクセスの許可 (方法), 104
 - ユーザーアクセスの拒否 (方法), 104
 - リモート, 187-189
 - ローカル, 44, 59, 184
 - ローカルプリンタまたは接続したプリンタの追加 (方法)
 - Solaris プリンタマネージャによる追加, 65
 - ローカルまたは接続の追加 (方法)
 - Solaris プリンタマネージャによる追加, 78
 - プリンタインタフェースプログラム, 183
 - プリンタクラス, 98
 - lpadmin コマンドによる定義 (方法), 99
 - 使用可能または使用不可にできない, 108
 - 状態のチェック, 106
 - 設定, 48
 - プリンタ、警告、「警告」を参照
- プリンタサーバー
 - アクセスのチェック, 430
 - 構成の確認, 428
 - システム資源の要求, 45
 - 接続の確認, 428
 - 設定 (概要), 64-69
 - 定義, 44, 64-69
 - ハードディスクの要件, 47
 - 必要なスプーリング空間, 46
 - プリンタの削除 (方法), 89
- プリンタサーバーに必要なスプーリング空間, 46
- プリンタ常駐フォント (概要), 141
- プリンタ情報, Solaris 2.5.1 からの変換, 85
- プリンタドライバ, プリンタ定義, 51
- プリンタの出力先
 - lpadmin コマンドによる設定 (概要), 94-95
 - Solaris プリンタマネージャによる設定, 48
- プリンタの状態, 91
- プリンタのタイプ
 - Solaris プリンタマネージャによる設定, 48
 - terminfo データベースにない場合, 52
 - terminfo データベースの定義, 175
 - 設定の問題の解決, 431
- プリンタ備考欄, lpadmin コマンドによる追加 (方法), 93
- プリンタへのアクセス, 削除 (方法), 88-90
- プリンタポート
 - Solaris プリンタマネージャによる設定, 48
 - 構成, 50
 - シリアル, 49
 - 特性、調整 (概要), 145-147
 - 特性の調整 (方法), 146-147
 - パラレル, 49
 - 複数のポートを有効にする, 50
- プリンタポート特性の調整 (概要), 145-147
- プリンタ名, 48
- プリンタメーカー, プリンタ定義, 50
- プリンタモデル, プリンタ定義, 50
- フルクラッシュダンプディレクトリからの復元, 402
- プログラム
 - 強制終了, 410
 - ディスク依存度, 351
 - 割り込み, 410
- プログラムの強制終了, 410
- プログラムの割り込み, 410

プロセス

- nice 値, 326, 337, 339
- proc ツールコマンド, 328
- proc ツールコマンドによる情報の表示, 329
- proc ツールによる情報の表示, 328
- アカウントティングユーティリティ, 289, 291, 306
- アドレス空間マップ, 329
- アプリケーションスレッド, 321, 323
- 一時的な停止, 329
- 強制終了, 329, 332
- 現在の作業用ディレクトリ, 329, 330
- 構造体, 322, 326
- 再開, 329
- シグナルアクション, 329
- 終了, 332
- 障害追跡, 339
- 情報の表示, 326, 333, 334
 - acctcom コマンド, 309
 - LWP, 329
 - ps コマンド, 326, 328
 - 実行されているプロセスの一覧表示, 328
 - 停止したプロセス, 309
 - 日次利用状況レポート, 305
 - プロセスの表示, 327
- 情報の表示 (方法), 330-331
- 情報を表示
 - 日次利用状況レポート, 306
- スケジューリングクラス, 333, 334, 335, 337
 - 指定, 334
 - 情報の表示, 326
 - デフォルト, 334
 - 変更, 336
 - 優先順位の変更, 334, 335, 337
 - 優先順位レベルと, 334, 335
- スタックトレース, 329
- 制御, 331-332
- 制御がきかなくなった, 339
- ツールコマンド, 329
- ツリー, 329, 330
- 定義, 321
- 開いたファイルの fstat 情報と fcntl 情報, 329
- 開いているファイルの fstat と fcntl 情報, 329, 330
- フラグのトレース, 329
- 問題の解決, 339
- 優先順位, 333, 334, 335, 337

プロセス, 優先順位 (続き)

- 概要, 337
- 最大, 334
- 指定, 334, 335
- 情報の表示, 326
- スケジューリングクラスと, 334, 335
- タイムシェアリングプロセスの優先順位の変更, 335, 337
- 用語, 321, 323
- 利用状況のアカウントティング, 305
- リンクされているライブラリ, 329
- プロセスアカウントティング, 289, 291
- プロセスのアカウントティング, 305, 306
- プロセスの強制終了, 329, 332
- プロセスファイルシステム (PROCFS), 328
- プロセス別コアファイルの設定, coreadm を使用した設定, 390

へ

変更

- crontab ファイル, 274
- dumpadm を使用したクラッシュダンプ構成, 399
- 印刷要求の優先順位 (概要), 105
- 印刷要求の優先順位 (方法), 115
- 各ユーザー用のディスク割り当て制限, 265, 266
- 時間, 236, 237
- システムのホスト名, 238
- スケジューリングクラス, 336, 337
- その日のメッセージ, 237
- 日時, 236, 237
- フォーム用紙, 134
- 優先順位, 334, 335, 337
 - タイムシェアリングプロセス, 335, 337
- 弱い期間制限, 264, 265

編集

- crontab ファイル, 274, 275

ほ

- ポート, 50
- 状態 (表), 223
- 初期化プロセス, 209-210

ポート (続き)

初期化 (方法), 202

定義, 194

ポートモニター

ttymon と listen (定義), 210-213

ttymon と listen (定義), 195

状態 (表), 223

定義, 195

ボーレート設定, 422

ホスト常駐フォント, ダウンロード (概要), 141-142

ホスト名, 変更, 238

ま

毎週の作業 (crontab によるスケジューリング), 270

毎月の作業 (crontab によるスケジューリング), 270

毎日のタスク (crontab によるスケジューリング), 270

待ち行列, 表示, 283

む

無効にする

consadm コマンドを使用した補助コンソール, 388

クラッシュダンプの保存, 402

特定のユーザーのディスク割り当て制限, 266, 267

め

メモリー

仮想

プロセス, 323

共有

プロセス仮想メモリー, 323

情報の表示, 235

情報の表示の例, 235

情報を表示するコマンド, 231

プロセス構造体, 322

も

文字セット

選択可能, 118, 119

ソフトウェア, 118

ハードウェア, 118, 119, 120

番号, 118

モデム

管理用ツール, 196

異なる使用方法, 194

シリアルポートツールの概要, 198

シリアルポートツールのモデムテンプレート, 198

設定 (方法), 201-202

着信専用サービス, 194

定義, 194

発信専用サービス, 194

発着信両用サービス, 194, 210

問題の解決, 423

tty 回線, 305

tty 回線, 305

印刷の問題, 419, 438

ソフトウェアパッケージのインストール/削除, 463-465

プロセス, 339

ゆ

有効にする

consadm コマンドを使用した補助コンソール, 386

coreadm を使用したグローバルコアファイルパス, 393

coreadm を使用したプロセス別コアファイルパス, 393

クラッシュダンプを有効にする, 402

システムリブート後の補助コンソール, 387-388

ディスク割り当て制限, 261

ユーザー

印刷要求の取り消し (方法), 113

プリンタへのアクセス, 48

プリンタへのアクセス (概要)

lpadmin コマンドによる制限, 103-104

Solaris プリンタマネージャによる制限, 103-104

ユーザーディスク割り当て制限

各ユーザー用に変更, 265

- ユーザーディスク割り当て制限 (続き)
 - 設定, 258, 259
 - 超過の確認, 262
 - 特定のユーザーについて無効にする, 266, 267
- ユーザーのログイン
 - 時間の監視, 289, 306
- ユーザープロセス
 - CPU 使用, 306
 - 優先順位, 333
 - 優先順位の変更, 337
- ユーザーへの課金, 295
 - 「chargefee スクリプト」も参照
- ユーザーモード優先順位, 333
- ユーザー料金, 290, 291, 295, 306
 - 「ユーザーへの課金」も参照
- ユーザーログイン
 - 最後のログインの監視, 302, 313
 - 最後のログインレポート, 308
 - 時間の監視, 302
 - ログインの数, 306
- ユーザー割り当て制限
 - 超えているかどうかのチェック, 262
 - 個々のユーザーの変更, 266
 - 弱い期間制限, 264
- 優先順位 (プロセス)
 - 概要, 333, 337
 - グローバル, 333, 334
 - 最大, 334
 - 指定, 334, 335
 - 情報の表示, 326, 334
 - スケジューリングクラスと, 334, 335
 - 変更, 334, 335, 337
 - タイムシェアリングプロセス, 335, 337
 - ユーザーモード優先順位, 333

- よ
- 要求 ログ, 184
- 容量 (ディスク)
 - 空き容量, 240, 349
 - 最適化, 239-251
- 読み取りシステムコール, 統計情報, 307
- 弱い期間制限
 - 超えている, 264
 - 変更, 264, 265

- ら
- ラスターイメージプロセッサ, (RIP), 26

- り
- リアルタイムプロセス
 - クラスの変更, 336, 337
 - 優先順位
 - 相対的な指定, 334
- リポート
 - /var/adm/wtmpx ファイルと, 291
 - クラッシュ後の失敗, 405-407
 - 接続アカウントと, 289
 - 日次レポートと, 304
- リモート印刷
 - 処理図, 187-189
 - ユーザー料金の計算, 295
- 料金 (ユーザー), 306
- 料金、ユーザー, 291, 295
- 履歴ログ (印刷要求), 176-179

- る
- ルートパスワードの復元
 - SPARC, 407
 - x86, 408
- ルートパスワード、忘れた
 - SPARC, 407
 - x86, 408

- ろ
- ローカル印刷, 184
- ローカルプリンタ
 - 設定用作業マップ, 59
 - 定義, 44
- ローカルプリンタまたは接続したプリンタ
 - Solaris プリンタマネージャによる追加 (方法), 65, 78
- ログインの監視
 - 最後のログイン, 302, 308, 313
 - 時間の使用状況, 289
 - 時間の利用状況, 306
 - 使用時間, 291
 - ログインの数, 306

- ログファイル, 184
 - LP 印刷サービス, 175-176, 421
 - 印刷待ち行列, 176
 - 印刷要求履歴ログ, 176-179
 - 自動的な削除, 275
 - 消去, 184
 - 要求, 184
 - 要求ログのコード, 177

わ

- 忘れたルートパスワード
 - SPARC, 407
 - x86, 408
- 割り当て制限
 - 概要, 253, 255
 - 確認, 255, 261, 265
 - 超えているユーザー割り当て制限の
チェック, 262
 - 個々のユーザーの変更, 266
 - 削除, 264
 - 情報の表示, 255, 261, 262
 - 設定, 255
 - 無効, 267
 - 無効にする, 256
 - 有効にする, 256
 - 有効、例, 261
 - ユーザー
 - 超えているかどうかのチェック, 262
 - 個々のユーザーの変更, 266
 - 弱い期間制限, 264, 265
 - 弱い期間制限
 - 超えている, 264
 - 変更, 265
- 割り当て制限を無効にする, 256, 267
- 割り当て制限を有効にする, 256
- 割り当て制限を有効にする、例, 261

