

Solaris のシステム管理 (上級編)

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

目次

はじめに	15
1 端末とモデムの管理 (概要)	21
端末とモデムの管理に関する新機能	21
SPARC: Coherent Console	21
SPARC: コンソールの \$TERM 値の設定方法の変更	22
システムコンソールでの ttymon の起動 (SMF による管理)	22
端末、モデム、ポート、サービス	23
端末について	23
モデムについて	23
ポートについて	24
サービスについて	24
ポートモニター	24
端末とモデムを管理するツール	25
シリアルポートツール	26
サービスアクセス機能 (SAF) の概要	26
2 端末とモデムの設定 (タスク)	27
端末とモデムの設定 (タスクマップ)	27
シリアルポートツールによる端末とモデムの設定 (概要)	28
端末の設定	28
モデムの設定	29
端末とモデムを設定し、ポートを初期化する方法 (タスク)	31
▼ 端末を設定する方法	31
▼ モデムを設定する方法	31
▼ ポートを初期化する方法	32
端末とモデムの問題をトラブルシューティングする方法	33

3	サービスアクセス機能によるシリアルポートの管理(タスク)	35
	シリアルポートの管理(タスクマップ)	36
	サービスアクセス機能(SAF)の概要	37
	SAF全体の管理(sacadm)	38
	サービスアクセスコントローラ(SACプログラム)	38
	SACの初期化プロセス	39
	ポートモニターサービス管理(pmadm)	39
	ttymon ポートモニター	39
	ポートの初期化プロセス	40
	発着信両用サービス	40
	TTY モニターとネットワークリスナーポートモニター	41
	TTY ポートモニター(ttymon)	41
	ttymon とコンソールポート	41
	ttymon 固有の管理コマンド(ttyadm)	42
	ネットワークリスナーサービス(listen)	42
	listen 固有の管理コマンド(nlsadmin)	42
	ttymon ポートモニターの管理	43
	▼ ttymon コンソールの端末タイプを設定する方法	43
	▼ ttymon コンソールの端末でボーレート速度を設定する方法	44
	▼ ttymon ポートモニターを追加する方法	45
	▼ ttymon ポートモニターのステータスを表示する方法	46
	▼ ttymon ポートモニターを停止する方法	47
	▼ ttymon ポートモニターを起動する方法	47
	▼ ttymon ポートモニターを無効にする方法	47
	▼ ttymon ポートモニターを有効にする方法	48
	▼ ttymon ポートモニターを削除する方法	48
	ttymon サービスの管理(タスクマップ)	49
	ttymon サービスの管理	49
	▼ サービスを追加する方法	49
	▼ TTY ポートサービスのステータスを表示する方法	50
	▼ ポートモニターサービスを有効にする方法	52
	▼ ポートモニターサービスを無効にする方法	53
	サービスアクセス機能の管理(リファレンス)	53
	SAFに関連付けられているファイル	53
	/etc/saf/_sactab ファイル	54
	/etc/saf/pmtab/_pmtab ファイル	54

サービスの状態	56
ポートモニターの状態	56
ポートの状態	57
4 システムリソースの管理 (概要)	59
システムリソース管理の新機能	59
製品名を表示する新しい prtconf オプション	59
psrinfo コマンドオプションによるチップマルチスレッド化機能の識別	60
新しい localeadm コマンド	60
システムリソースの管理 (ロードマップ)	61
5 システム情報の表示と変更 (タスク)	63
システム情報の表示 (タスクマップ)	63
システム情報の表示	64
▼ システムの物理プロセッサタイプを表示する方法	70
▼ システムの論理プロセッサタイプを表示する方法	71
▼ システムにインストールされているロケールを表示する方法	72
▼ ロケールがシステムにインストールされているかどうかを調べる方法	72
システム情報の変更 (タスクマップ)	73
システム情報の変更	74
▼ システムの日付と時間を手作業で設定する方法	74
▼ その日のメッセージを設定する方法	75
▼ システムのホスト名の変更方法	75
▼ システムにロケールを追加する方法	77
▼ システムからロケールを削除する方法	77
6 ディスク使用の管理 (タスク)	79
ディスク使用の管理 (タスクマップ)	79
ファイルとディスク容量の情報の表示	80
▼ ファイルとディスク容量の情報を表示する方法	81
ファイルサイズの確認	83
▼ ファイルサイズを表示する方法	84
▼ サイズの大きなファイルを見つける方法	85
▼ 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	86

ディレクトリサイズの確認	87
▼ 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 ...	87
▼ 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	88
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	89
▼ 最新ファイルのリストを表示する方法	90
▼ 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	90
▼ 一時ディレクトリを一度にクリアする方法	91
▼ コアファイルを見つけて削除する方法	92
▼ クラッシュダンプファイルを削除する方法	93
7 UFS 割り当て制限の管理 (タスク)	95
UFS 割り当て制限	95
UFS 割り当て制限の使用	95
UFS 割り当て制限の弱い制限値と強い制限値の設定	96
ディスクブロックとファイル制限の相違	96
UFS 割り当て制限の設定	97
UFS 割り当て制限の設定のガイドライン	97
UFS 割り当て制限の設定 (タスクマップ)	98
▼ ファイルシステムに UFS 割り当て制限を構成する方法	99
▼ ユーザーに UFS 割り当て制限を設定する方法	100
▼ 複数ユーザーに UFS 割り当て制限を設定する方法	101
▼ UFS 割り当て制限の整合性を確認する方法	101
▼ UFS 割り当て制限を有効にする方法	102
UFS 割り当て制限の管理 (タスクマップ)	103
UFS 割り当て制限のチェック	104
▼ UFS 割り当て制限を超過したかどうかをチェックする方法	104
▼ ファイルシステムの UFS 割り当て制限をチェックする方法	105
UFS 割り当て制限の変更と削除	106
▼ 弱い制限値のデフォルトを変更する方法	107
▼ 1 ユーザーの UFS 割り当て制限を変更する方法	108
▼ 1 ユーザーの UFS 割り当て制限を無効にする方法	109
▼ UFS 割り当て制限を無効にする方法	110
8 システムタスクのスケジュール設定 (タスク)	113
crontab ファイルの作成と編集 (タスクマップ)	113

システムタスクを自動的に実行する方法	114
繰り返されるジョブのスケジューリング (crontab)	115
1つのジョブのスケジューリング (at)	115
繰り返されるシステムタスクのスケジューリング (cron)	116
crontab ファイルの内容	116
cron デーモンのスケジューリング管理	117
crontab ファイルエントリの構文	118
crontab ファイルの作成と編集	119
▼ crontab ファイルを作成または編集する方法	119
▼ crontab ファイルを確認する方法	120
crontab ファイルの表示	121
▼ crontab ファイルを表示する方法	121
crontab ファイルの削除	122
▼ crontab ファイルを削除する方法	122
crontab コマンドの使用制御	123
▼ crontab コマンドの使用を拒否する方法	124
▼ crontab コマンドの使用を特定のユーザーに限定する方法	125
crontab コマンドの使用制限を確認する方法	126
at コマンドの使用 (タスクマップ)	126
1つのシステムタスクのスケジューリング (at)	127
at コマンドの説明	127
at コマンドの使用制御	128
▼ at ジョブを作成する方法	128
▼ at 待ち行列を表示する方法	129
▼ at ジョブを確認する方法	130
▼ at ジョブを表示する方法	130
▼ at ジョブを削除する方法	130
▼ at コマンドの使用を拒否する方法	131
▼ at コマンドの使用の拒否を確認する方法	132
9 システムアカウントの設定と管理(タスク)	133
システムアカウントの新機能	133
Oracle Solaris プロセスアカウントと統計の改善点	133
システムアカウント	134
システムアカウントの動作	134

システムアカウントのコンポーネント	135
システムアカウント (タスクマップ)	139
システムアカウントの設定	140
▼ システムアカウントを設定する方法	141
ユーザーへの課金	142
▼ ユーザーに課金する方法	143
アカウント情報の管理	143
壊れたファイルと wtmpx エラーを修復する	143
▼ 壊れた wtmpx ファイルの修復方法	144
tacct エラーを修復する	144
▼ tacct エラーを修復する方法	145
runacct スクリプトを再起動する	145
▼ runacct スクリプトを再起動する方法	146
システムアカウントの停止と無効	146
▼ 一時的にシステムアカウントを停止する方法	146
▼ システムアカウントを永続的に無効にする方法	147
10 システムアカウント (リファレンス)	149
runacct スクリプト	149
日次アカウントレポート	152
日次レポート	152
日次利用状況レポート	153
日次コマンドサマリー	155
月次コマンドサマリー	156
最終ログインレポート	156
acctcom による pacct ファイルの確認	157
システムアカウントファイル	159
runacct スクリプトが生成するファイル	161
11 システムパフォーマンスの管理 (概要)	163
システムパフォーマンスの管理に関する新機能	163
強化された pfiles ツール	163
CPU パフォーマンスカウンタ	164
システムパフォーマンスタスクに関する情報の参照箇所	164
システムパフォーマンスおよびシステムリソース	165

プロセスとシステムのパフォーマンス	165
システムパフォーマンスの監視	167
監視ツール	167
12 システムプロセスの管理(タスク).....	169
システムプロセスの管理に関する新機能	169
擬似カーネルプロセス	169
システムプロセスの管理(タスクマップ)	170
システムプロセスを管理するコマンド	171
ps コマンドの使用	172
/proc ファイルシステムとコマンド	173
プロセスコマンド(/proc)を使用したプロセスの管理	174
▼ プロセスを表示する方法	175
▼ プロセスに関する情報を表示する方法	176
▼ プロセスを制御する方法	177
プロセスの終了 (pkill, kill)	178
▼ プロセスを終了させる方法 (pkill)	178
▼ プロセスを終了させる方法 (kill)	179
プロセスのデバッグ (pargs, preap)	180
プロセスクラス情報の管理(タスクマップ)	181
プロセスクラス情報の管理	181
プロセスのスケジュール優先順位の変更 (priocntl)	182
▼ プロセスクラスに関する基本情報を表示する方法 (priocntl)	182
▼ プロセスのグローバル優先順位を表示する方法	183
▼ プロセスの優先順位を指定する方法 (priocntl)	183
▼ 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)	184
▼ プロセスのクラスを変更する方法 (priocntl)	185
タイムシェアリングプロセスの優先順位の変更 (nice)	186
▼ プロセスの優先順位を変更する方法 (nice)	186
システムのプロセスに関するトラブルシューティング方法	188
13 システムパフォーマンスの監視(タスク)	189
システムパフォーマンス情報の表示(タスクマップ)	189
仮想メモリーの統計情報の表示 (vmstat)	190

▼仮想メモリーの統計情報を表示する方法 (vmstat)	191
▼システムイベント情報を表示する方法 (vmstat -s)	192
▼スワップの統計情報を表示する方法 (vmstat -S)	192
▼各デバイス当たりの割り込み数を表示する方法 (vmstat -i)	193
ディスク使用状況の表示 (iostat)	193
▼ディスクの使用状況を表示する方法 (iostat)	194
▼拡張ディスク統計情報を表示する方法 (iostat -xtc)	195
ディスク容量統計情報の表示 (df)	196
▼ディスク容量情報を表示する方法 (df -k)	196
システム動作の監視(タスクマップ)	197
システム動作の監視 (sar)	199
▼ファイルアクセスをチェックする方法 (sar -a)	199
▼バッファー動作をチェックする方法 (sar -b)	200
▼システムコールの統計情報をチェックする方法 (sar -c)	201
▼ディスク動作をチェックする方法 (sar -d)	202
▼ページアウトとメモリーをチェックする方法 (sar -g)	204
カーネルメモリーの割り当て (KMA) のチェック	205
▼カーネルメモリーの割り当てをチェックする方法 (sar -k)	206
▼プロセス間通信をチェックする方法 (sar -m)	207
▼ページイン動作をチェックする方法 (sar -p)	208
▼待ち行列動作をチェックする方法 (sar -q)	209
▼未使用のメモリーをチェックする方法 (sar -r)	210
▼CPU の使用状況をチェックする方法 (sar -u)	211
▼システムテーブルのステータスをチェックする方法 (sar -v)	212
▼スワップ動作をチェックする方法 (sar -w)	213
▼端末動作をチェックする方法 (sar -y)	214
▼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)	216
システム動作データの自動収集 (sar)	216
ブート時の sadc コマンドの実行	216
sa1 スクリプトを使用した sadc コマンドの定期的な実行	217
sa2 スクリプトを使用したレポートの生成	217
データの自動収集を設定する (sar)	217
▼自動データ収集を設定する方法	219

14	ソフトウェアのトラブルシューティング(概要)	221
	トラブルシューティングの新機能	221
	共通エージェントコンテナの問題	221
	x86: システムリブート時に SMF ブートアーカイブサービスが失敗する場 合がある	222
	動的トレース機能	222
	kadb に代わる標準 Solaris カーネルデバッガ kmdb	223
	ソフトウェアのトラブルシューティングタスクの参照先	223
	システムクラッシュの問題のトラブルシューティング	224
	システムがクラッシュした場合の対処方法	224
	トラブルシューティングデータの収集	224
	システムクラッシュをトラブルシュートするためのチェックリスト	225
15	システムメッセージの管理	227
	システムメッセージの表示	227
	▼システムメッセージを表示する方法	228
	システムログローテーション	229
	システムのメッセージ記録のカスタマイズ	230
	▼システムのメッセージ記録をカスタマイズする方法	232
	リモートコンソールメッセージングを有効にする	233
	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	234
	対話型ログインセッション中に <code>consadm</code> コマンドを使用する	235
	▼補助(リモート)コンソールを有効にする方法	235
	▼補助コンソールのリストを表示する方法	236
	▼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	236
	▼補助(リモート)コンソールを無効にする方法	237
16	コアファイルの管理(タスク)	239
	コアファイルの管理(タスクマップ)	239
	コアファイルの管理の概要	240
	構成可能なコアファイルのパス	240
	拡張されたコアファイル名	240
	コアファイル名パターンの設定	241
	<code>setuid</code> プログラムがコアファイルを作成できるようにする	242
	現在のコアダンプ構成を表示する方法	242

▼ コアファイル名パターンを設定する方法	243
▼ プロセス別コアファイルパスを有効にする方法	243
▼ グローバルのコアファイルパスを有効にする方法	243
コアファイルのトラブルシューティング	244
コアファイルの調査	244
17 システムクラッシュ情報の管理 (タスク)	247
システムクラッシュ情報の管理に関する新機能	247
高速クラッシュダンプ機能	247
システムクラッシュ情報の管理 (タスクマップ)	248
システムクラッシュ (概要)	248
スワップ領域およびダンプデバイスに関する Oracle Solaris ZFS サポート	249
x86: GRUB ブート環境のシステムクラッシュ	249
システムクラッシュダンプファイル	250
クラッシュダンプの保存	250
dumpadm コマンド	251
dumpadm コマンドの動作	252
ダンプデバイスとボリュームマネージャー	252
システムクラッシュダンプ情報の管理	252
▼ 現在のクラッシュダンプ構成を表示する方法	252
▼ クラッシュダンプ構成を変更する方法	253
▼ クラッシュダンプを検査する方法	255
▼ 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 (オプション)	256
▼ クラッシュダンプの保存を無効または有効にする方法	256
18 ソフトウェアで発生するさまざまなトラブルシューティング (タスク)	259
リポートが失敗した場合の対処	259
ルートパスワードを忘れた場合の対処	260
x86: システムリポート中に SMF ブートアーカイブサービスが失敗した場合の対処	263
システムがハングした場合の対処	264
ファイルシステムが一杯になった場合の対処	265
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	266
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	266
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	266
バックアップ時のトラブルシューティング	267
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	267
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	267
現在のディレクトリが間違っていないことを確認する	267
対話型コマンド	268
Oracle Solaris OS での共通エージェントコンテナのトラブルシューティング	268
ポート番号の競合	268
▼ポート番号をチェックする方法	269
スーパーユーザーパスワードのセキュリティの危殆化	269
▼Oracle Solaris OS のセキュリティ鍵を生成する方法	270
19 ファイルアクセスでのトラブルシューティング(タスク)	271
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)	271
▼検索パスの問題を診断して解決する方法	272
ファイルアクセスの問題を解決する	274
ファイルとグループの所有権の変更	274
ネットワークアクセスで発生する問題の把握	274
20 UFS ファイルシステムの不整合解決(タスク)	275
fsck のエラーメッセージ	276
fsck の一般エラーメッセージ	277
初期化フェーズでの fsck メッセージ	279
フェーズ 1: ブロックとサイズの検査のメッセージ	282
Oracle Solaris 10: フェーズ 1B: 走査し直して DUPS メッセージを表示する	287
フェーズ 1B: 走査し直して DUPS メッセージを表示する	288
フェーズ 2: パス名の検査のメッセージ	288
フェーズ 3: 接続性の検査のメッセージ	295
フェーズ 4: 参照数の検査のメッセージ	298
フェーズ 5: シリンダグループ検査のメッセージ	301
フェーズ 5: シリンダグループ検査のメッセージ	302
fsck サマリーメッセージ	303
クリーンアップ (後処理) フェーズのメッセージ	304

21	ソフトウェアパッケージで発生するトラブルシューティング(タスク)	307
	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トラブルシューティング	307
	特定のソフトウェアパッケージのインストールエラー	308
	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	309
	索引	311

はじめに

『Solaris のシステム管理ガイド(上級編)』は、Oracle Solaris システム管理に関する重要な情報を提供するドキュメントの一部です。このガイドには、SPARC ベースおよび x86 ベースのシステムの情報が含まれています。

このドキュメントでは、Oracle Solaris オペレーティングシステム (OS) がインストールされていることを想定しています。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。

システム管理者にとって重要な Oracle Solaris リリースの新機能については、該当する章の、新機能に関するセクションを参照してください。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。このドキュメントでは、プラットフォームにより実装が異なる場合は、それを特記します。

このドキュメントの x86 に関連する用語については、次を参照してください。

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

サポートされるシステムについては、[Oracle Solaris OS: Hardware Compatibility Lists](#) を参照してください。

対象読者

このドキュメントは、Oracle Solaris 10 を実行しているシステムの管理者を対象にしています。このドキュメントを読むには、UNIX 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 システム管理マニュアルセットの構成

システム管理ガイドセットに含まれる各ガイドとその内容は、次のとおりです。

マニュアルのタイトル	トピック
『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のサポート、システムのシャットダウンとブート、管理サービス、およびソフトウェアの管理 (パッケージとパッチ)
『Solaris のシステム管理 (上級編)』	端末とモデムの設定、システムリソースの管理 (ディスク割り当て、アカウントिंग、および crontab ファイルの管理)、システムプロセスの管理、および Oracle Solaris ソフトウェアの障害追跡
『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、および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 ネットワーク管理、IPv4 と IPv6 アドレス管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチのパス化 (IPMP)、および IPQoS
『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、および LDAP のネーミングとディレクトリサービス (NIS から LDAP への移行、および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス: NIS+ 編)』	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 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と autofs)、メール、SLP、および PPP
『Solaris のシステム管理ガイド (印刷)』	印刷に関するトピックや、サービス、ツール、プロトコル、およびテクノロジーを使って印刷サービスおよびプリンタを設定および管理する方法
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティ ティ、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、および Solaris Secure Shell
『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』	リソース管理に関連する計画とタスク、拡張アカウントिंग、リソース制御、フェアシェアスケジューラ (FSS)、リソース上限デーモン (rcapd) による物理メモリーの制御、およびリソースプール (Solaris Zones ソフトウェア区分技術と lx ブランドゾーンによる仮想化)

マニュアルのタイトル	トピック
『Oracle Solaris ZFS 管理ガイド』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Oracle Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Oracle Solaris Trusted Extensions 管理の手順』	Oracle Solaris Trusted Extensions 機能固有のシステム管理
『Oracle Solaris Trusted Extensions 構成ガイド』	Solaris 10 5/08 リリース以降での、Oracle Solaris Trusted Extensions 機能の計画、有効化、および初期構成の方法

関連するサードパーティーの Web サイト情報

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

Oracle サポートへのアクセス

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

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表P-1 表記上の規則

字体	意味	例
『』	コマンド名、ファイル名、ディレクトリ名、画面上的コンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上的コンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	Placeholder: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm filename と入力します。
<i>AaBbCc123</i>	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

表P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

一般規則

このドキュメントでは次の規則が使用されています。

- このドキュメント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、このドキュメントでは絶対パス名で表記していない場合があります。ただし、それ以外のあまり一般的でないディレクトリにあるコマンドについては、このドキュメント中の例では絶対パスで表記します。

◆◆◆ 第 1 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。

- 21 ページの「端末とモデムの管理に関する新機能」
- 23 ページの「端末、モデム、ポート、サービス」
- 25 ページの「端末とモデムを管理するツール」
- 26 ページの「シリアルポートツール」
- 26 ページの「サービスアクセス機能 (SAF) の概要」

シリアルポートツールを使用して端末やモデムを設定する手順については、第 2 章「端末とモデムの設定 (タスク)」を参照してください。

サービスアクセス機能 (SAF) を使用して端末やモデムを設定する手順については、第 3 章「サービスアクセス機能によるシリアルポートの管理 (タスク)」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

このセクションでは、Oracle Solaris で端末およびモデムを管理するための新機能、または機能の変更について説明します。Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『Oracle Solaris 10 8/11 の新機能』を参照してください。

SPARC: Coherent Console

Solaris 10 8/07: Coherent Console サブシステム機能は、カーネルコンソールサブシステムの一部を実装し、コンソール出力のレンダリングを容易にします。Coherent Console では、Programmable Read-Only Memory (PROM) インタフェースではなく、Oracle Solaris のカーネルメカニズムを使用してコンソール出力がレンダリングさ

れます。これにより、コンソールレンダリングの OpenBoot PROM (OBP) に対する依存性が減少します。Coherent Console では、カーネル常駐のフレームバッファードライバを使用してコンソール出力を生成します。生成されるコンソール出力は、OBP レンダリングを使用する場合よりも効率的です。Coherent Console では、SPARC コンソール出力時のアイドルリングも回避され、ユーザー体験が向上します。

SPARC: コンソールの \$TERM 値の設定方法の変更

Solaris 10 8/07: \$TERM 値は現在、動的に派生し、コンソールが使用する端末エミュレータに依存しています。x86 ベースのシステムでは、カーネルの端末エミュレータが常に使用されるため、\$TERM 値は sun-color です。

SPARC ベースのシステムでは、\$TERM 値は次のとおりです。

sun-color	システムでカーネルの端末エミュレータが使用される場合は、\$TERM にこの値が使用されます。
sun	システムで PROM の端末エミュレータが使用される場合は、\$TERM にこの値が使用されます。

この変更は、端末タイプをシリアルポートに設定する方法には影響しません。\$TERM 値を変更するには、次の例に示すように、引き続き svccfg コマンドを使用することができます。

```
# svccfg
svc:> select system/console-login
svc:/system/console-login> setprop ttymon/terminal_type = "xterm"
svc:/system/console-login> exit
```

システムコンソールでの ttymon の起動 (SMF による管理)

Oracle Solaris 10: システムコンソールでの ttymon の起動は、SMF によって管理されます。svc:/system/console-login:default サービスにプロパティを加えることにより、svccfg コマンドで ttymon コマンドの引数を指定できるようになりました。これらのプロパティは、SMF の汎用プロパティではなく ttymon 固有のものであることに注意してください。

注 - /etc/inittab ファイルでの ttymon の起動はカスタマイズできなくなりました。

SMF で ttymon コマンド引数を指定する方法については、43 ページの「[ttymon コンソールの端末タイプを設定する方法](#)」を参照してください。

SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。SMF に関する手順の詳細については、『Solaris のシステム管理 (基本編)』の第 19 章「サービスの管理 (手順)」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、システムリソースとネットワークリソース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。このセクションでは、Oracle Solaris オペレーティングシステム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。

端末について

システムのビットマップグラフィックディスプレイは、文字端末とは異なります。英数字端末はシリアルポートに接続され、テキストのみを表示します。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。

モデムについて

モデムには、以下の 3 つの基本構成があります。

- 発信専用
- 着信専用
- 双方向

家庭用コンピュータに接続されたモデムは、「発信専用」サービスを提供するように設定されていることがあります。発信専用サービスを使用すると、自宅から他のコンピュータにアクセスできます。しかし、外部からそのマシンにアクセスすることはできません。

「着信専用」サービスは、その逆です。着信サービスを使用すると、リモート地からシステムにアクセスできます。しかし、外部への発信はできません。

「発着信両用」アクセスは、その名のとおりに、発信と着信の両方の機能を備えています。

ポートについて

「ポート」とは、デバイス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）コンポーネントからなっています。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、標準的な通信プロトコルを使用し、1バイト単位の情報を1本の回線上で1ビットずつ伝送します。

RS-232-CまたはRS-423標準に従って設計されたデバイスには、大半のモデム、文字端末、プロッタ、および一部のプリンタが含まれます。これらのデバイスは、同様に設計されたコンピュータのシリアルポートであればどれにでも、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デバイスを接続する必要がある場合、システムに「アダプタボード」を追加しなければならないことがあります。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。

サービスについて

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リソース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

サービスへアクセスする主なメカニズムは、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。

次の表に、Oracle Solaris リリースで提供されている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 1-1 ポートモニターのタイプ

マニュアルページ	ポートモニター	説明
listen(1M)	listen	Solaris 2.6 より前のリリースで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します。デフォルトの Oracle Solaris OS は、このタイプのポートモニターを使用しません。
ttymon(1M)	ttymon	モデムや文字端末が必要とするログインサービスへのアクセスを提供します。シリアルポートツールは、これらのデバイス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します。

getty と呼ばれる従来のポートモニターを使い慣れたユーザーもいます。新しい ttymon ポートモニターはより強力です。1つの ttymon ポートモニターで複数の 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。詳細は、[getty\(1M\)](#) のマニュアルページを参照してください。

端末とモデムを管理するツール

次の表に、端末とモデムを管理するためのツールの一覧を示します。

表 1-2 端末とモデムを管理するツール

端末とモデムの管理方法	ツール	詳細
管理作業全般	サービスアクセス機能 (SAF) のコマンド	26 ページの「サービスアクセス機能 (SAF) の概要」
簡易設定	Solaris Management Console のシリアルポートツール	第2章「端末とモデムの設定 (タスク)」 および Solaris Management Console のオンラインヘルプ

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。

このツールは、以下も提供します。

- 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
- 各ポートのステータスの簡易表示

サービスアクセス機能 (SAF) の概要

SAFは、端末、モデム、およびその他のネットワークデバイスの管理用のツールです。

SAFを使用すると、具体的には以下を設定できます。

- `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` デバイスに関するトラブルシューティング
- ネットワークからの印刷サービス要求に関するトラブルシューティング
- サービスアクセスコントローラに関するトラブルシューティング (`sacadm` コマンドを使用)

SAFは、`tty` デバイスやローカルエリアネットワーク (LAN) を通じたシステムリソースやネットワークリソース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 2 章

端末とモデムの設定 (タスク)

この章では、Solaris Management Console 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ステップごとの手順を説明します。

端末とモデムの概要については、第 1 章「[端末とモデムの管理 \(概要\)](#)」を参照してください。システムリソースの管理に関する概要については、第 4 章「[システムリソースの管理 \(概要\)](#)」を参照してください。

Solaris Management Console 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については、27 ページの「[端末とモデムの設定 \(タスクマップ\)](#)」を参照してください。

端末とモデムの設定 (タスクマップ)

タスク	説明	参照先
端末を設定します。	Solaris Management Console のシリアルポートツールを使用し、端末を設定します。「アクション」メニューから適切なオプションを選択し、端末を構成します。	31 ページの「端末を設定する方法」
モデムを設定します。	Solaris Management Console のシリアルポートツールを使用し、モデムを設定します。「アクション」メニューから適切なオプションを選択し、モデムを構成します。	31 ページの「モデムを設定する方法」

タスク	説明	参照先
ポートを初期化します。	ポートを初期化するには、Solaris Management Console のシリアルポートツールを使用します。「アクション」メニューから適切なオプションを選択します。	32 ページの「ポートを初期化する方法」

シリアルポートツールによる端末とモデムの設定 (概要)

Solaris Management Console 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」メニューからシリアルポートを選択し、「アクション」メニューから「構成」オプションを選択して、次の内容を構成します。

- 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」オプションは、これらのサービスを構成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」メニューから「プロパティ」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3章「サービスアクセス機能によるシリアルポートの管理(タスク)」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

次の表は、シリアルポートツール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 2-1 端末のデフォルト値

詳細	項目	デフォルト値
基本	ポート	

表 2-1 端末のデフォルト値 (続き)

詳細	項目	デフォルト値
	説明	端末
	サービスのステータス	有効
	ボーレート	9600
	端末の種類	vi925
	ログインプロンプト	ttyn login:
拡張	キャリア検出	ソフトウェア
	選択: キャリア検出時に接続	使用不可
	選択: 双方向	使用可能
	選択: 初期化のみ	使用不可
	タイムアウト (秒)	なし
	ポートモニター	zsmon
	サービスプログラム	/usr/bin/login

モデムの設定

次の表は、シリアルポートツールを使用して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 2-2 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ますが、発信はできません。
発信専用	モデムから発信はできますが、着信はできません。
発着信両用	モデムへ着信も、モデムから発信もできます。

次の表に各テンプレートのデフォルト値を示します。

表 2-3 モデム用テンプレートのデフォルト値

詳細	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	ポート名			
	説明	Modem - Dial In Only	Modem - Dial Out Only	Modem - Dial In and Out

表 2-3 モデム用テンプレートのデフォルト値 (続き)

詳細	項目	モデム-着信専用	モデム-発信専用	モデム-着信と発信
	サービスのステータス	有効	有効	有効
	ボーレート	9600	9600	9600
	ログインプロンプト	ttyn login:	ttyn login:	ttyn login:
拡張	キャリア検出	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	使用不可	使用不可	使用不可
	選択: 双方向	使用不可	使用不可	使用可能
	選択: 初期化のみ	使用不可	使用可能	使用不可
	タイムアウト (秒)	なし	なし	なし
	ポートモニター	zsmon	zsmon	zsmon
	サービスプログラム	/usr/bin/login	/usr/bin/login	/usr/bin/login

次の表では、「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 2-4 「初期化のみ」のデフォルト値

詳細	項目	デフォルト値
基本	ポート名	—
	説明	初期化のみ - 接続なし
	サービスのステータス	有効
	ボーレート	9600
	ログインプロンプト	ttyn login:
拡張	キャリア検出	ソフトウェア
	選択: キャリア検出時に接続	使用不可
	選択: 双方向	使用可能
	選択: 初期化のみ	使用可能
	タイムアウト (秒)	なし
	ポートモニター	zsmon
	サービスプログラム	/usr/bin/login

端末とモデムを設定し、ポートを初期化する方法(タスク)

▼ 端末を設定する方法

- 1 Solaris Management Console が実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris Management Console の起動方法については、『Solaris のシステム管理(基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

- 2 ナビゲーションペインで「このコンピュータ」アイコンをクリックします。
- 3 「デバイスとハードウェア」、「シリアルポート」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」メニューから「構成」→「端末」を選択します。
「シリアルポートを構成」ウィンドウが「基本」モードで表示されます。
「端末」メニュー項目については、表 2-1 を参照してください。
- 6 「了解(OK)」をクリックします。
- 7 各項目を構成するには、端末として構成されたポートを選択します。
- 8 「アクション(Action)」メニューから「プロパティ(Properties)」を選択します。
- 9 必要な場合は、テンプレートエントリの値を変更します。
- 10 「了解」をクリックしてポートを構成します。
- 11 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 ttyn
```

▼ モデムを設定する方法

- 1 Solaris Management Console が実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris Management Console の起動方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

- 2 ナビゲーションペインで「このコンピュータ」アイコンをクリックします。
- 3 「デバイスとハードウェア」、「シリアルポート」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」メニューから、次の「構成」オプションのいずれかを選択します。
 - a. 「構成」→「モデム(着信)」
 - b. 「構成」→「モデム(発信)」
 - c. 「構成」→「モデム(着信/発信)」
「シリアルポートを構成」ウィンドウが「基本」モードで表示されます。
「モデム」メニュー項目については、表 2-3 を参照してください。
- 6 「了解(OK)」をクリックします。
- 7 各項目を構成するには、モデムとして構成されたポートを選択します。
- 8 「アクション(Action)」メニューから「プロパティ(Properties)」を選択します。
- 9 必要な場合は、テンプレートエントリの値を変更します。
- 10 「了解」をクリックしてポートを構成します。
- 11 モデムのサービスが構成されたことを確認します。

```
$ pmadm -l -s tty#
```

▼ ポートを初期化する方法

- 1 Solaris Management Console が実行していなければ、起動します。

```
% /usr/sadm/bin/smc &
```

Solaris Management Console の起動方法については、『Solaris のシステム管理 (基本編)』の「Solaris 管理コンソールを起動する」を参照してください。

- 2 ナビゲーションペインで「このコンピュータ」アイコンをクリックします。

- 3 「デバイスとハードウェア」、「シリアルポート」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」→「初期化のみ-接続なし」を選択します。
「シリアルポート」ウィンドウが「基本」モードで表示されます。
「初期化のみ」メニュー項目については、表 2-4 を参照してください。
- 6 「了解 (OK)」をクリックします。
- 7 拡張機能の項目を構成するには、初期化のみで構成されたポートを選択します。次に、「アクション」メニューから「プロパティ」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」をクリックしてポートを構成します。
- 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トラブルシュートする方法

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してトラブルシューティングを始めてください。

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
 - 端末の構成が正しくない
 - 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構成を調べてトラブルシュートし続けます。端末またはモデムとの通信の正しい tty 名を調べます。端末またはモデムの設定が 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。pmdm 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にサービスを提供するように構成されていることと、正しい tty 名が関連付けられていることを確認します。例:

```
$ pmdm -l -t ttymon
```

/etc/ttydefs ファイルを調べ、ラベル定義を端末構成と照合してチェックします。sacadm コマンドを使用してポートモニターのステータスを調べます。pmdm 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ており、さらに次の条件を満たしている場合があります。

- pmdm コマンド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する。
- 端末の構成がポートモニターの構成と一致する。

この場合は、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。Oracle Solaris 10 から、コンソールでの ttymon の起動は SMF によって管理されています。コンソールの端末タイプを変更する方法については、[43 ページの「ttymon コンソールの端末タイプを設定する方法」](#)を参照してください。

ttymon と SMF の詳細については、[21 ページの「端末とモデムの管理に関する新機能」](#)を参照してください。

サービスアクセス機能によるシリアルポートの管理(タスク)

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。

さらに、サービス管理機能 (SMF) を使用してコンソールを管理する方法についても説明します。

注 - SAF と SMF は、Oracle Solaris OS で使用される 2 つの異なるツールです。Oracle Solaris 10 より、システムコンソールでの `ttymon` の起動は SMF により管理されるようになりました。SAF は、現バージョンでも端末、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として使用されます。

この章の内容は以下のとおりです。

- 37 ページの「サービスアクセス機能 (SAF) の概要」
- 38 ページの「SAF 全体の管理 (sacadm)」
- 39 ページの「ポートモニターサービス管理 (pmadm)」
- 41 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」

シリアルポートを管理する手順については、以下を参照してください。

- 36 ページの「シリアルポートの管理 (タスクマップ)」
- 49 ページの「`ttymon` サービスの管理 (タスクマップ)」

SAF の参照情報については、53 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。

シリアルポートの管理(タスクマップ)

タスク	説明	参照先
コンソール管理を行います。	次のコンソール管理タスクを行う必要があります。 <ul style="list-style-type: none"> ■ ttymon コンソールの端末タイプを設定します。 Oracle Solaris 10 からは、svccfg コマンドを使用して ttymon コンソールの端末タイプを指定する必要があります。 ■ ttymon コンソールの端末ボーレート速度を設定します。 	43 ページの「ttymon コンソールの端末タイプを設定する方法」 44 ページの「ttymon コンソールの端末でボーレート速度を設定する方法」
ttymon ポートモニターを追加します。	sacadm コマンドを使用して ttymon ポートモニターを追加します。	45 ページの「ttymon ポートモニターを追加する方法」
ttymon ポートモニターのステータスを表示します。	sacadm コマンドを使用して ttymon ポートモニターのステータスを表示します。	46 ページの「ttymon ポートモニターのステータスを表示する方法」
ttymon ポートモニターを停止します。	sacadm コマンドを使用して ttymon ポートモニターを停止します。	47 ページの「ttymon ポートモニターを停止する方法」
ttymon ポートモニターを起動します。	sacadm コマンドを使用して ttymon ポートモニターを起動します。	47 ページの「ttymon ポートモニターを起動する方法」
ttymon ポートモニターを無効にします。	sacadm コマンドを使用して ttymon ポートモニターを無効にします。	47 ページの「ttymon ポートモニターを無効にする方法」
ttymon ポートモニターを有効にします。	sacadm コマンドを使用して ttymon ポートモニターを有効にします。	48 ページの「ttymon ポートモニターを有効にする方法」
ttymon ポートモニターを削除します。	sacadm コマンドを使用して ttymon ポートモニターを削除します。	48 ページの「ttymon ポートモニターを削除する方法」

サービスアクセス機能 (SAF) の概要

Solaris Management Console のシリアルポートツールまたは SAF コマンドを使用して、端末とモデムを設定することができます。

SAF は、端末、モデム、およびその他のネットワークデバイス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。

次の表に 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。

一方、`ttymon` と `listen` のサービスは `pmadm` コマンドによって制御されます。`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 3-1 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>	シリアルポートのログイン要求を監視します
	<code>listen</code>	ネットワークのサービス要求を監視します
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ーコール	SAF がアクセスを可能にするサービス

表 3-1 SAFの制御階層 (続き)

機能	プログラム	説明
コンソールの管理	コンソールログイン	コンソールサービスはSMFサービス (svc:/system/console-login:default) で管理されます。このサービスは、ttymon ポートモニターを起動します。pmadm コマンドや sacadm コマンドを使用してコンソールを管理しないでください。詳細は、41 ページの「ttymon とコンソールポート」、43 ページの「ttymon コンソールの端末タイプを設定する方法」、および 44 ページの「ttymon コンソールの端末でボーレート速度を設定する方法」を参照してください。

SAF全体の管理 (sacadm)

sacadm コマンドは、SAFの上位レベルにあります。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。sacadm にはそれ以外に、ポートモニターの現在のステータスの表示、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監視します。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 プログラムは、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 プログラム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 プログラム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 プログラム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 プログラム管理ファイル内のコマンドを使用して、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を初めて起動したときの一連の処理を要約します。

1. SACプログラムは、SMFサービス (`svc:/system/sac:default`) によって起動されます。
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` コマンドを使用すると、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 `svctag` 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

直結モデムまたは文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。まず、SACプロセスがSMFによって起動されます。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。`ttymon` ポートモニターは起動されると、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。

英数字端末またはモデム経由でログインが試行されると、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渡します。`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ています。`ttymon` ポートモニターは、装置との通信に必要なデータ転送速度、回線制御手順、および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、ポートごとに、回線制御手順が指定されていればその手順を最初に初期化し、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 ポートモニターは、`/etc/utmpx` 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構成されている場合、ttymon ポートモニターは次のように動作します。

- 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、将来のポートモニターや他社製のポートモニターに対応するために一般的な管理方法を提供していますが、Oracle Solaris リリースでは `ttymon` と `listen` の2つのポートモニター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 `getty` ポートモニターと同じサービスを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。この管理ファイルは `pmadm` コマンドおよび `ttyadm` 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` 管理ファイルによって管理されるわけではありません。`ttymon` の呼び出しは SMF によって管理されます。そのため、`/etc/inittab` ファイルにエントリを追加し、`ttymon` を起動することはできなくなりました。タイプが `application`、名前が `ttymon` のプロパティグループが SMF サービス (`svc:/system/console-login:default`) に追加されました。このプロパティグループに含まれるプロパティは、メソッドスクリプト (`/lib/svc/method/console-login`) によって使用されます。このスクリプトは、プロパティ値を `ttymon` 起動の際の引数として使用します。通常、値が空白であったり、値がどのプロパティでも指定されていない場合は、値は `ttymon` によって使用されません。ただし、`ttymon` デバイス値が空白であったり、設定されていない場合は、`/dev/console` が `ttymon` 実行のデフォルトとして使用されます。

この SMF サービス (`svc:/system/console-login:default`) では、以下のプロパティが使用できます。

<code>ttymon/nohangup</code>	<code>nohangup</code> プロパティを指定します。 <code>true</code> に設定されている場合、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し、回線を強制的にハンガアップさせないでください。
------------------------------	--

<code>ttymon/prompt</code>	コンソールポート用のプロンプト文字列を指定します。
<code>ttymon/terminal_type</code>	コンソールのデフォルトの端末タイプを指定します。
<code>ttymon/device</code>	コンソールデバイスを指定します。
<code>ttymon/label</code>	<code>/etc/ttydefs</code> 行の TTY 名を指定します。

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` コマンドの他に `ttymon` 固有の `ttymon` 固有の情報を書式化し、そ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` コマンドおよび `pmadm` コマンドに提示する手段を提供します。

したがって、`ttymon` コマンドは `ttymon` を直接管理するわけではありません。 `ttymon` コマンドは、一般的な管理用コマンドである `sacadm` および `pmadm` を補足します。詳細は、[ttymon\(1M\)](#) 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ポートモニターの管理ファイル内に指定します。この管理ファイルは `pmadm` コマンドおよび `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。 Oracle Solaris OS 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` コマンドおよび `pmadm` コマンドのほかに、`nlsadmin` コマンドでも更新できます。 `nlsadmin` コマンドは、`listen` 固有の情報を書式化し、その情報を標準出力に書き出し、書式化された `listen` 固有のデータを `sacadm` コマンドおよび `pmadm` コマンドに提示する手段を提供します。

したがって、`nlsadmin` コマンドが `listen` を直接管理するわけではありません。このコマンドは、一般的な管理用コマンドである `sacadm` および `pmadm` を補足します。

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、あるネットワーク用に `listen` ポートモニターを構成し、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

`ttymon` のコンソール管理は、SMF で管理されるようになりました。`svccfg` コマンドを使用し、`ttymon` システムコンソールのプロパティを設定します。今後も `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 コンソールの端末タイプを設定する方法

この手順では、`svccfg` コマンドを使用してコンソールの端末タイプを変更する方法を説明します。

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

- 2 `svccfg` コマンドを実行して、変更するサービスインスタンスのプロパティを設定します。

```
# svccfg -s console-login setprop ttymon/terminal_type = "xterm"
「xterm」とは、使用する端末タイプの例です。
```

- 3 (オプション) サービスインスタンスを再起動します。

```
# svcadm restart svc:/system/console-login:default
```



注意-サービスインスタンスを即座に再起動する場合は、コンソールからログアウトされます。サービスインスタンスを即座に再起動しない場合は、プロパティの変更は、次回ログイン時にコンソールのプロンプトで適用されます。

▼ ttymon コンソールの端末でボーレート速度を設定する方法

次の手順では、ttymon コンソールの端末でボーレート速度を設定する方法を示します。x86 ベースのシステムでのコンソール速度のサポートは、特定のプラットフォームに依存します。

SPARC ベースのシステムには、次のコンソール速度がサポートされています。

- 9600 bps
- 19200 bps
- 38400 bps

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 システムタイプに適したボーレート速度を設定するには、**eeprom** コマンドを使用します。

```
# eeprom ttya-mode=baud-rate,8,n,1,-
```

たとえば、x86 ベースシステムのコンソールのボーレート速度を 38400 に変更するには、次のように入力します。

```
# eeprom ttya-mode=38400,8,n,1,-
```
- 3 **/etc/ttydefs** ファイルのコンソール行を次のように変更します。

```
console baud-rate hupcl opost onlcr:baud-rate::console
```
- 4 システムのタイプに合わせて、次の追加変更を行います。
これらの変更はプラットフォームに依存することに注意してください。
 - **SPARC** ベースのシステム: **/kernel/drv/options.conf** ファイルでボーレート速度を変更します。
ボーレートを 9600 に変更するには、次のコマンドを使用します。

```
# 9600          :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

ボーレート速度を 19200 に変更するには、次のコマンドを使用します。

```
# 19200         :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

ボーレート速度を 38400 に変更するには、次のコマンドを使用します。

```
# 38400         :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

- x86 ベースのシステム: BIOS のシリアル切り替えが有効である場合にコンソール速度を変更します。コンソール速度を変更する方法は、プラットフォームに依存します。

▼ 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るか、同等の役割を引き受けます。

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

- 2 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します。
- p mbmon をポートモニタータグとして指定します。
- t ポートモニタータイプを ttymon として指定します。
- 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します。
- v ポートモニターのバージョン番号を指定します。
- 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定義します。

▼ ttymon ポートモニターのステータスを表示する方法

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

- 2 ttymon ポートモニターのステータスを表示します。

```
# sacadm -l -p mbmon
```

```
-l   ポートモニターステータス表示用のフラグを指定します。
```

```
-p   mbmon をポートモニタータグとして指定します。
```

例3-1 ttymon ポートモニターのステータスを表示する

この例では、mbmon と呼ばれるポートモニターを表示する方法を示します。

```
# 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 であることを示します。

PMTYPE        ポートモニターのタイプが ttymon であることを示します。

FLGS          次のフラグが設定されているかどうかを示します。
d             新しいポートモニターを有効にしません。
x             新しいポートモニターを起動しません。
ダッシュ(-)   フラグを設定しません。

RCNT          戻りカウント値を示します。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します。

STATUS        ポートモニターの現在のステータスを示します。

COMMAND       ポートモニターを起動するコマンドを示します。

#TTY Ports a & b  ポートモニターを説明するコメントを示します。
```

▼ 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 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ステータスにするフラグを指定します。

-p mbmon をポートモニタータグとして指定します。

▼ 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 終了状態の 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ステータスにするフラグを指定します。

-p mbmon をポートモニタータグとして指定します。

▼ 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 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ステータスにするフラグを指定します。

-p mbmon をポートモニタータグとして指定します。

▼ 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

- 2 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ステータスにするフラグを指定します。

-p mbmon をポートモニタータグとして指定します。

▼ ttymon ポートモニターを削除する方法

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。

注-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。

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

- 2 ttymon ポートモニターを削除します。

```
# sacadm -r -p mbmon
```

-r ポートモニターステータス削除用のフラグを指定します。

-p mbmon をポートモニタータグとして指定します。

ttymon サービスの管理 (タスクマップ)

タスク	説明	参照先
ttymon サービスを追加します。	pmadm コマンドを使用して サービスを追加します。	49 ページの「サービスを追加する方法」
TTY ポートサービスのステータスを表示します。	pmadm コマンドを使用して TTY ポートのステータスを表示します。	50 ページの「TTY ポートサービスのステータスを表示する方法」
ポートモニターサービスを有効にします。	pmadm コマンドを -e オプションとともに使用し、ポートモニターを有効にします。	52 ページの「ポートモニターサービスを有効にする方法」
ポートモニターサービスを無効にします。	pmadm コマンドを -d オプションとともに使用し、ポートモニターを無効にします。	53 ページの「ポートモニターサービスを無効にする方法」

ttymon サービスの管理

pmadm コマンドを使用すると、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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 mbmon ポートモニターに、標準の端末サービスを追加します。

```
# 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。実際には Return キーを押さずに (改行なしに) 入力します。

- a ポートモニターステータス追加用のフラグを指定します。
- p mbmon をポートモニタータグとして指定します。

- s a をポートモニターサービスタグとして指定します。
- 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します。
- v ポートモニターのバージョン番号を指定します。
- m ttyadm により書式化された ttymon 固有の構成データを指定します。

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します。
- i *inactive* (無効応答) メッセージを指定します。
- l /etc/ttydefs ファイルにあるどの TTY 名を使用するか指定します。
- m サービスを起動する前にプッシュする STREAMS モジュールを指定します。
- d TTY ポートに使用する装置へのフルパス名を指定します。
- 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します。引数が必要な場合、コマンドと引数を二重引用符 (“”) で囲みます。

▼ TTY ポートサービスのステータスを表示する方法

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 ポートのステータスを表示します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 ポートモニターのいずれかのサービスをリストします。

```
# pmadm -l -p mbmon -s a
```

- l システム上のサービス情報を表示します。
- p mbmon をポートモニタータグとして指定します。
- s a をポートモニターサービスタグとして指定します。

例 3-2 1 つの TTY ポートモニターサービスのステータスを表示する

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 FLGS 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 を示します。

PMTYPE ポートモニターのタイプが ttymon であることを示します。

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します。

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指定します。

- x - サービスを有効にしません。
- u - サービス用の utmpx エントリを作成します。
- ダッシュ (-) - フラグを設定しません。

ID 起動時にサービスに割り当てられた ID を示します。この値は、pmadm -i コマンドを使用して設定されます。

<PMSPECIFIC> 情報

/dev/term/a ttyadm -d コマンドを使用して設定された、TTY ポートパス名を示します。

- 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します。

- c - キャリア検出時に接続するよう、ポートを設定します。
- 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) と設定します。
- h - 着呼が受信された直後の自動ハングアップを抑制します。
- I - ポートを初期化します。
- 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 を待機させます。
- ダッシュ (-) - フラグを設定しません。

- ttyadm -r count オプションを使用して設定された値を示します。このオプションは、ポートからデータを受信したあとに、ttymon がプロンプトを表示するタイミングを決定します。count が 0 の場合、ttymon は任意の文字を受信するまで待機します。count 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します。この例では、値は設定されていません。
<code>/usr/bin/login</code>	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します。この値は、 <code>ttyadm -s</code> コマンドを使用して設定されます。
<code>-</code>	<code>ttyadm -t</code> コマンドのタイムアウト値を示します。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します。この例では、タイムアウト値は設定されていません。
<code>contty</code>	<code>/etc/ttydefs</code> ファイル中の TTY 名を示します。この値は、 <code>ttyadm -l</code> コマンドを使用して設定されます。
<code>ldterm,ttcompat</code>	プッシュする STREAMS モジュールを示します。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ます。
<code>login: Terminal disabled</code>	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します。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ます。
<code>tvi925</code>	<code>ttyadm -T</code> コマンドを使用して設定されている場合、端末タイプを示します。この例では、端末タイプは、 <code>tvi925</code> です。
<code>y</code>	<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します。n は、ソフトウェアキャリアをオフにします。y は、ソフトウェアキャリアをオンにします。この例では、ソフトウェアキャリアはオンです。
<code>#</code>	<code>pmadm -y</code> コマンドで指定した任意のコメントを示します。この例では、コメントは存在しません。

▼ 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るか、同等の役割を引き受けます。

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

- 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 有効フラグを指定します。

-p mbmon をポートモニタータグとして指定します。

-s a をポートモニターサービスタグとして指定します。

▼ ポートモニターサービスを無効にする方法

- スーパーユーザーになるか、同等の役割を引き受けます。

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

- 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 無効フラグを指定します。

-p mbmon をポートモニタータグとして指定します。

-s a をポートモニターサービスタグとして指定します。

サービスアクセス機能の管理(リファレンス)

このセクションには、サービスアクセス機能の管理に関するリファレンスが記載されています。

SAFに関連付けられているファイル

SAFは構成ファイルを使用しますが、このファイルは `sacadm` コマンドおよび `pmadm` コマンドを使用して変更できます。構成ファイルを手動で編集する必要はありません。

ファイル名	説明
/etc/saf/_sysconfig	システムごとの構成スクリプト
/etc/saf/_sactab	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SACの管理ファイル

ファイル名	説明
<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>	<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ポートモニター <code>pmtag</code> の管理ファイル
<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` ファイルの情報は以下のとおりです。

```
# 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します。
zsmon                ポートモニター名です。
ttymon               ポートモニターのタイプです。
::                   次の2つのフラグが設定されているかどうかを示します。
d                    ポートモニターを有効にしません。
x                    ポートモニターを起動しません。この例では、どちらのフラグも設定されていません。
0                    戻りコード値を示します。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します。
/usr/lib/saf/ttymon  ポートモニターのパス名を示します。
```

`/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: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します。
ttya            サービスタグを示します。
x,u            次のフラグが設定されているかどうかを示します。
x              サービスを有効にしません。
u              サービス用の utmpx エントリを作成します。
root           サービスタグに割り当てられた ID を示します。
reserved       このフィールドは、将来の拡張のために予約されています。
reserved       このフィールドは、将来の拡張のために予約されています。
reserved       このフィールドは、将来の拡張のために予約されています。
/dev/term/a    TTY ポートパス名を示します。
/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示
                します。
:c,b,h,I,r:    次のフラグが設定されているかどうかを示します。
c              キャリア検出時に接続するよう、ポートを設定します。
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)と設定します。
h              着呼が受信された直後の自動着信処理を抑制します。
I              ポートを初期化します。
r              ttymon が 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 を待機させます。
9600           /etc/ttydefs ファイルに定義されている TTY 名を指定します。
ldterm,ttcompat プッシュする STREAMS モジュールを示します。
ttya login\:   表示するプロンプトを指定します。
:y/n:         「はい」か「いいえ」かの応答を示します。
message       任意のアクティブでない(無効な)応答メッセージを指定しま
                す。
tvi925        端末タイプを指定します。
y            ソフトウェアキャリアが設定されているかどうかを示します
                (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。次に、サービスの起こりうる状態に関する説明を一覧で示します。

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ります。

無効 デフォルト状態。ポートモニターを削除したとき、サービスは停止します。

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 ポートモニターおよび 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ます。
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ります。
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します。
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します。
起動中	中間状態 - ポートモニターの起動が進行中です。
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ますが、まだシャットダウン手続きは完了していません。ポートモニターが停止状態になるまでの途中の状態です。
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です。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ります。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ません。
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ません。

特定のポートモニターの状態を確認するには、次のコマンドを入力します。


```
# 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します。
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。この状態を指定した場合、ttymon は接続要求を受け取ると「disabled」メッセージを送信します。

システムリソースの管理 (概要)

この章では、Oracle Solaris OS のシステムリソース管理機能に関する簡単な説明と、システムリソースの管理に役立つロードマップを記載します。

これらの機能を使用すると、一般的なシステム情報を表示したり、ディスク容量監視したり、ディスク割り当て制限を設定したり、アカウントティングプログラムを使用したりできます。また、ルーチンコマンドを自動的に実行するよう cron コマンドと at コマンドをスケジュールすることもできます。

このセクションでは、柔軟な方法でシステムリソースの割り当て、監視、および制御を可能にするリソース管理機能については言及しません。

リソース管理機能を使用せずにシステムリソースを管理する手順については、61 ページの「[システムリソースの管理 \(ロードマップ\)](#)」を参照してください。

リソース管理機能を使用して、システムリソースを管理する方法については、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 1 章「[Solaris 10 リソース管理の紹介](#)」を参照してください。

システムリソース管理の新機能

このセクションでは、Oracle Solaris でシステムリソースを管理するための新機能、または機能の変更について説明します。Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『[Oracle Solaris 10 8/11 の新機能](#)』を参照してください。

製品名を表示する新しい prtconf オプション

Solaris 10 1/06: システムの製品名を表示するため、新しい `-b` オプションが `prtconf` コマンドに追加されています。このオプションは `uname -i` コマンドに似ています。ただし、`prtconf -b` コマンドは、製品のマーケティング名を判別するために特別に設計されています。

`prtconf` コマンドに `-b` オプションを使用することで表示されるファームウェアデバイスツリーのルートプロパティは、次のようになります。

- name
- compatible
- banner-name
- model

使用できる場合があるプラットフォーム固有の追加出力を表示するには、`prtconf -vb` コマンドを使用します。詳細は、[prtconf\(1M\)](#) のマニュアルページおよび [69 ページ](#) の「システムの製品名を表示する方法」を参照してください。

psrinfo コマンドオプションによるチップマルチスレッド化機能の識別

Oracle Solaris 10: `psrinfo` コマンドが変更され、仮想プロセッサの情報だけでなく、物理プロセッサの情報も返すようになりました。この拡張機能の追加によって、チップマルチスレッド化 (CMT) 機能を識別できるようになりました。新しい `-p` オプションは、システム内の物理プロセッサの合計数を報告します。`psrinfo -pv` コマンドを使用すると、システム内の全物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサも表示されます。`psrinfo` コマンドのデフォルト出力は、これまでどおりシステムの仮想プロセッサ情報を表示します。

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

この機能に関連する手順については、[70 ページ](#) の「システムの物理プロセッサタイプを表示する方法」を参照してください。

新しい localeadm コマンド

Oracle Solaris 10: 新しい `localeadm` コマンドを使用すると、OS をインストールし直したり、パッケージを手動で追加および削除したりせずに、システム上のロケールを変更できます。このコマンドを使用してシステムに照会すると、どのロケールがインストールされているかを特定することもできます。`localeadm` コマンドを実行するには、ユーザーはスーパーユーザー特権を持っているか、役割によるアクセス制御 (RBAC) を通じて同等の役割になる必要があります。

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

詳細について、このガイドでは、[第5章「システム情報の表示と変更\(タスク\)」](#)を参照してください。

Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『[Oracle Solaris 10 8/11 の新機能](#)』を参照してください。

システムリソースの管理(ロードマップ)

タスク	説明	手順
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、言語環境、日付と時間、システムのホスト名などの、システム情報を表示したり変更したりします。	第5章「システム情報の表示と変更(タスク)」
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6章「ディスク使用の管理(タスク)」
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7章「UFS 割り当て制限の管理(タスク)」
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ちます。	第8章「システムタスクのスケジュール設定(タスク)」
システムアカウントの管理	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リソースを識別します。	第9章「システムアカウントの設定と管理(タスク)」
Solaris リソース管理機能を使用したシステムリソースの管理	リソースマネージャを使用して、アプリケーションが利用可能なシステムリソースをどのように使用するかを制御したり、リソースの利用を追跡して、課金したりします。	『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の第1章「Solaris 10 リソース管理の紹介」

システム情報の表示と変更 (タスク)

この章では、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タスクを示します。

システム情報の表示と変更に関する手順については、以下を参照してください。

- 63 ページの「システム情報の表示 (タスクマップ)」
- 73 ページの「システム情報の変更 (タスクマップ)」

システムリソースの管理に関する概要については、第4章「システムリソースの管理 (概要)」を参照してください。

システム情報の表示 (タスクマップ)

タスク	説明	参照先
システムで 32 ビットまたは 64 ビットの機能が有効になっているかどうかを調べます。	isainfo コマンドを使用し、システムで 32 ビットまたは 64 ビット機能が有効になっているかどうかを確認します。x86 システムについては、isalist コマンドを使用してこの情報を表示できます。	65 ページの「システムで 32 ビットまたは 64 ビット機能が有効になっているか調べる方法」
Oracle Solaris リリース情報を表示します。	Oracle 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。	68 ページの「Oracle Solaris リリース情報を表示する方法」
一般的なシステム情報を表示します。	showrev コマンドを使用し、一般的なシステム情報を表示します。	68 ページの「一般的なシステム情報を表示する方法」

タスク	説明	参照先
システムのホスト ID 番号を表示します。	hostid コマンドを使用し、システムのホスト ID を表示します。	69 ページの「システムのホスト ID を表示する方法」
システムの製品名を表示します。	Solaris 10 1/06 リリースから、prtconf -b コマンドを使用して、システムの製品名を表示できます。	69 ページの「システムの製品名を表示する方法」
システムにインストールされているメモリーを表示します。	prtconf コマンドを使用し、システムにインストールされたメモリーに関する情報を表示します。	70 ページの「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」
システムの日付と時間を表示します。	date コマンドを使用し、システムの日付と時間を表示します。	70 ページの「日付と時間を表示する方法」
システムの物理プロセッサタイプを表示します。	psrinfo -p コマンドを使用し、システム上の物理プロセッサの合計数を表示します。 psrinfo -pv コマンドを使用すると、システム内の全物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサも表示されます。	70 ページの「システムの物理プロセッサタイプを表示する方法」
システムの論理プロセッサタイプを表示します。	psrinfo -v コマンドを使用し、システムの論理プロセッサタイプを表示します。	71 ページの「システムの論理プロセッサタイプを表示する方法」
システムにインストールされているロケールを表示します。	localeadm コマンドを使用し、システムにインストールされているロケールを表示します。	72 ページの「システムにインストールされているロケールを表示する方法」
ロケールがシステムにインストールされているかどうかを調べます。	localeadm コマンドの -q オプションとロケール名を使用して、そのロケールがシステムにインストールされているかどうかを調べます。	72 ページの「ロケールがシステムにインストールされているかどうかを調べる方法」

システム情報の表示

次の表に、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 5-1 システム情報を表示するためのコマンド

コマンド	表示されるシステム情報	マニュアルページ
date	日付と時間	date(1)
hostid	ホスト ID	hostid(1)
isainfo	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。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ます。	isainfo(1)
isalist	x86 ベースのシステムのプロセッサタイプ	psrinfo(1M)
localeadm	システムにインストールされているロケール	localeadm(1M)
prtconf	システム構成情報、インストールされたメモリー、および製品名	prtconf(1M)
psrinfo	プロセッサタイプ	psrinfo(1M)
showrev	ホスト名、ホスト ID、リリース、カーネルアーキテクチャー、アプリケーションアーキテクチャー、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	showrev(1M)
uname	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	uname(1)

▼ システムで **32** ビットまたは **64** ビット機能が有効になっているか調べる方法

- **isainfo** コマンドを使用し、システムで **32** ビットまたは **64** ビット機能が有効になっているかどうかを確認します。

```
# isainfo options
```

isainfo コマンドをオプションなしで実行すると、現在の OS バージョンでサポートされるアプリケーションのネイティブ命令セットの名前が表示されます。

- v ほかのオプションに関する詳細情報を表示します。
- b ネイティブ命令セットのアドレス空間のビット数を表示します。
- n OS の現在のバージョンでサポートされる、移植性のあるアプリケーションで使用されるネイティブ命令セットを表示します。

- k デバイスドライバや STREAMS モジュールなどの OS カーネルコンポーネントで使用される命令セットの名前を表示します。

注-x86 ベースのシステムでは、`isalist` コマンドを使用してこの情報を表示することも可能です。

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

例 5-1 SPARC: システムで 32 ビットまたは 64 ビット機能が有効になっているか調べる

旧リリースの Oracle Solaris OS を 32 ビットカーネルで実行する UltraSPARC システムの `isainfo` コマンド出力は、次のように表示されます。

```
$ 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。

最新リリースの Oracle Solaris OS で SPARC ベースのシステム用に出荷されるのは、64 ビットカーネルだけです。64 ビットカーネルを実行する UltraSPARC システムの `isainfo` コマンド出力は、次のように表示されます。

```
$ 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 Oracle Solaris OS を実行する SPARC ベース、x86 ベース、UltraSPARC システムの出力は、次のように表示されます。

```
$ isainfo -b
32
```

64 ビット Oracle Solaris OS を実行する 64 ビット UltraSPARC システムの `isainfo` コマンド出力は、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。

例 5-2 x86: システムで 32 ビットまたは 64 ビット機能が有効になっているか調べる

64 ビットカーネルを実行する x86 システムの `isainfo` コマンド出力は、次のように表示されます。

```
$ isainfo
amd64 i386
```

この出力は、システムが 64 ビットアプリケーションをサポートすることを示します。

x86 システムが 32 ビットカーネルを実行できるかどうか調べるには、`isainfo -v` コマンドを使用します。

```
$ isainfo -v
64-bit amd64 applications
    fpu tsc cx8 cmov mmx ammx a3dnow a3dnowx fxsr sse sse2
32-bit i386 applications
    fpu tsc cx8 cmov mmx ammx a3dnow a3dnowx fxsr sse sse2
```

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 Oracle Solaris OS を実行する x86 ベースシステムの出力は、次のように表示されます。

```
$ isainfo -b
32
```

64 ビット Oracle Solaris OS を実行する x86 システムの `isainfo` コマンド出力は、次のように表示されます。

```
$ isainfo -b
64
```

また、x86 システムが 32 ビットモードと 64 ビットモードのどちらで動作しているかを判別するには、`isalist` コマンドも使用できます。

```
$ isalist
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

上の例の amd64 は、システムで 64 ビットの機能が有効になっていることを示します。

▼ Oracle Solaris リリース情報を表示する方法

- 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。

```
$ cat /etc/release
Oracle Solaris 10 s10_51 SPARC
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All Rights Reserved.
Use is subject to license terms.
Assembled 28 February 2011
```

▼ 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。

```
$ showrev options
-a                利用できるすべてのシステムリビジョン情報を表示します。
-c (command)     コマンドに関するリビジョン情報を表示します。
-p                パッチに関するリビジョン情報のみを表示します。
-R (root_path)   root_path として使用するディレクトリのフルパス名を定義します。
-s (host name)   この操作を実行するホスト名を指定します。
-w                OpenWindows のリビジョン情報のみを表示します。
```

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、およびプロセッサタイプが表示されます。

```
$ uname
SunOS
$ uname -a
SunOS starbug 5.10 Generic sun4u sparc SUNW,Ultra-5_10
$
```

例 5-3 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: stonetouch
Hostid: 8099dfb9
Release: 5.10
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider:
Domain:
Kernel version: SunOS 5.10 s10_46

OpenWindows version:
Solaris X11 Version 6.6.2 19 November 2010
No patches are installed
```

▼ 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 5-4 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
```

▼ システムの製品名を表示する方法

Solaris 10 1/06: **prtconf** コマンドの **-b** オプションを使用すると、システムの製品名を表示できます。この機能の詳細については、[prtconf\(1M\)](#) のマニュアルページを参照してください。

- システムの製品名を表示するには、**-b** オプションを付けて **prtconf** コマンドを使用します。

```
% prtconf -b
```

例 5-5 システムの製品名を表示する

次に、**prtconf -b** コマンドの出力例を示します。

```
$ prtconf -b
name: SUNW,Ultra-5_10
model: SUNW,375-0066
banner-name: Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
```

次に、**prtconf -vb** コマンドの出力例を示します。

```
$ prtconf -vb
name: SUNW,Ultra-5_10
```

```
model: SUNW,375-0066
banner-name: Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIIi 333MHz)
idprom: 01800800.20a6c363.00000000.a6c363a9.00000000.00000000.405555aa.aa555500
openprom model: SUNW,3.15
openprom version: 'OBP 3.15.2 1998/11/10 10:35'
```

▼ 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容量を表示するには、**prtconf** コマンドを使用します。

例 5-6 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 コマンドの出力を示します。grep Memory コマンドは prtconf コマンドの出力内容を選別して、メモリー情報だけを表示します。

```
$ prtconf | grep Memory
Memory size: 128 Megabytes
```

▼ 日付と時間を表示する方法

- システムクロックに従った現在の日付と時間を表示するには、**date** コマンドを使用します。

例 5-7 日付と時間を表示する

次の例は、date コマンドの出力を示します。

```
$ date
Wed Jan 21 17:32:59 MST 2004
$
```

▼ システムの物理プロセッサタイプを表示する方法

- **psrinfo -p** コマンドを使用し、システム上の物理プロセッサの合計数を表示します。

```
$ psrinfo -p
1
```

psrinfo -pv コマンドを使用すると、システム上の各物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサの情報も表示されます。

```
$ psrinfo -pv
The UltraSPARC-IV physical processor has 2 virtual processors (8, 520)
```

```
The UltraSPARC-IV physical processor has 2 virtual processors (9, 521)
The UltraSPARC-IV physical processor has 2 virtual processors (10, 522)
The UltraSPARC-IV physical processor has 2 virtual processors (11, 523)
The UltraSPARC-III+ physical processor has 1 virtual processor (16)
The UltraSPARC-III+ physical processor has 1 virtual processor (17)
The UltraSPARC-III+ physical processor has 1 virtual processor (18)
The UltraSPARC-III+ physical processor has 1 virtual processor (19)
```

psrinfo -pv コマンドを x86 ベースのシステムで使用すると、以下の出力が表示されます。

```
$ psrinfo -pv
The i386 physical processor has 2 virtual processors (0, 2)
The i386 physical processor has 2 virtual processors (1, 3)
```

▼ システムの論理プロセッサタイプを表示する方法

- **psrinfo -v** コマンドを使用して、システムのプロセッサタイプに関する情報を表示します。

```
$ psrinfo -v
```

x86 ベースのシステムでは、isalist コマンドを使用し、仮想プロセッサタイプを表示します。

```
$ isalist
```

例 5-8 SPARC: システムのプロセッサタイプを表示する

この例では、SPARC システムのプロセッサタイプに関する情報を表示する方法を示します。

```
$ psrinfo -v
Status of virtual processor 0 as of: 04/16/2004 10:32:13
  on-line since 03/22/2004 19:18:27.
  The sparcv9 processor operates at 650 MHz,
  and has a sparcv9 floating point processor.
```

例 5-9 x86: システムのプロセッサタイプを表示する

この例では、x86 ベースシステムのプロセッサタイプに関する情報を表示する方法を示します。

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

▼ システムにインストールされているロケールを表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 **localeadm** コマンドを使用し、システムに現在インストールされているロケールを表示します。**-l** オプションを使用すると、システムにインストールされているロケールを表示できます。例:

```
# localeadm -l
Checking for installed pkgs. This could take a while.

Checking for Australasia region (aua)
(1of2 pkgs)
|.....|
.
.
.
The following regions are installed on concordance on Wed Dec 17 15:13:00 MST 2003

POSIX (C)

Central Europe (ceu)
[ Austria, Czech Republic, Germany, Hungary, Poland, Slovakia,
Switzerland (German), Switzerland (French) ]

Done.
```

▼ ロケールがシステムにインストールされているかどうかを調べる方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 システムにロケールがインストールされているかどうかは、**localeadm** コマンドを使用して調べます。**-q** オプションとロケール名を使用して、そのロケールがインス

トールされているかどうかをシステムに照会します。たとえば、中央ヨーロッパ地域(ceu)がシステムにインストールされているかどうかを調べるには、次のように実行します。

```
# localeadm -q ceu
locale/region name is ceu
Checking for Central Europe region (ceu)
.
.
.
The Central Europe region (ceu) is installed on this system
```

システム情報の変更(タスクマップ)

タスク	説明	参照先
システムの日付と時間を手動で設定します。	システムの日付と時間を手動で設定するには、 <code>date mmd HHMM[[cc]yy]</code> コマンド行構文を使用します。	74 ページの「システムの日付と時間を手作業で設定する方法」
その日のメッセージを設定します。	システムのその日のメッセージを設定するには、 <code>/etc/motd</code> ファイルを編集します。	75 ページの「その日のメッセージを設定する方法」
システムのホスト名を変更します。	次のファイルを編集し、システムのホスト名を変更します。 <ul style="list-style-type: none"> ■ <code>/etc/nodename</code> ■ <code>/etc/hostname.*host-name</code> ■ <code>/etc/inet/hosts</code> 注 - Solaris 3/05、1/06、6/06、または 11/06 リリースを実行している場合は、 <code>/etc/inet/ipnodes</code> ファイルの更新も必要です。Solaris 10 8/07 リリース以降、 <code>hosts</code> ファイルは OS 内で 1 つになりました。 <code>/etc/inet/hosts</code> ファイルが唯一の <code>hosts</code> ファイルであり、この中に IPv4 と IPv6 の両方のエントリが含まれます。	75 ページの「システムのホスト名の変更方法」
システムにロケールを追加します。	システムにロケールを追加するには、 <code>localeadm</code> コマンドを使用します。	システムにロケールを追加する方法

タスク	説明	参照先
システムからロケールを削除します。	システムからロケールを削除するには、 <code>localeadm</code> コマンドの <code>-r</code> オプションと削除したいロケール名を使用します。	システムからロケールを削除する方法

システム情報の変更

このセクションでは、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間を手作業で設定する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 次のように新しい日付と時間を入力します。

```
# date mmddHHMM[[cc]yy]
```

mm 月。2桁を使用します。
dd 日。2桁を使用します。
HH 時。2桁で24時間制を使用します。
MM 分。2桁を使用します。
cc 世紀。2桁を使用します。
yy 年。2桁を使用します。

詳細については、[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 5-10 システムの日付と時間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間を設定する方法を示します。

```
# date
Wed Mar 3 14:04:19 MST 2004
```

```
# date 0121173404
Thu Jan 21 17:34:34 MST 2004
```

▼ 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、システム的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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 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、リターンも含めます。
- 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 5-11 その日のメッセージを設定する

Oracle Solaris ソフトウェアのインストール時に設定されるデフォルトのその日のメッセージには、バージョン情報が含まれています。

```
$ cat /etc/motd
Oracle Corporation SunOS 5.10 Generic January 2005
```

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。

ホスト名の変更は、次の手順で行なってください。

また、`sys-unconfig` コマンドを使用しても、ホスト名を含めて、システムの再構成が可能です。詳細は、[sys-unconfig\(1m\)](#) のマニュアルページを参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[RBACの構成\(作業マップ\)](#)」を参照してください。
- 2 次のファイルにあるシステムのホスト名を変更します。
 - `/etc/nodename`
 - `/etc/hostname.*interface`
 - `/etc/inet/hosts`
 - `/etc/inet/ipnodes` (一部のリリースのみに該当します)

注 - Solaris 10 8/07 リリース以降、`hosts` ファイルは1つになりました。`/etc/inet/hosts` ファイルが唯一の `hosts` ファイルであり、この中に IPv4 と IPv6 の両方のエントリが含まれます。常に同期させる必要がある2つの `hosts` ファイルに、IPv4 エントリを保持する必要はありません。`/etc/inet/ipnodes` ファイルは、下位互換性のために、`/etc/inet/hosts` ファイルへの同名のシンボリックリンクに置き換えられています。詳細は、[hosts\(4\)](#) のマニュアルページを参照してください。

- 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 `/var/crash` ディレクトリ内のホスト名ディレクトリの名前を変更します。

```
# cd /var/crash
# mv old-host-name new-host-name
```
- 5 システムをリブートして、新しいホスト名をブートします。

```
# init 6
```

▼ システムにロケールを追加する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 **localeadm** コマンドを使用して、システムにインストールしたいロケールのパッケージを追加します。
追加したいロケールは、**-a** オプションの後ろに入力します。追加したいロケールパッケージのあるデバイスは、**-d** オプションの後ろに入力します。たとえば、中央ヨーロッパ地域(ceu)をシステムに追加するには、次のように実行します。

```
# localeadm -a ceu -d /net/install/latest/Solaris/Product
locale/region name is ceu
Devices are /net/install/latest/Solaris/Product
.
.
.
One or more locales have been added.
To update the list of locales available at
.
.
.
```

▼ システムからロケールを削除する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 **localeadm** コマンドを使用して、システムにインストールしたロケールのパッケージを削除します。
システムから削除したいロケールは、**-r** オプションの後ろに入力します。たとえば、中央ヨーロッパ地域(ceu)をシステムから削除するには、次のように実行します。

```
# localeadm -r ceu
locale/region name is ceu
Removing packages for Central Europe (ceu)
.
.
.
One or more locales have been removed.
```

To update the list of locales available
at the login screen's "Options->Language" menu,

.
.
.

ディスク使用の管理 (タスク)

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。

ディスク使用の管理に関する手順については、79 ページの「ディスク使用の管理 (タスクマップ)」を参照してください。

ディスク使用の管理 (タスクマップ)

タスク	説明	参照先
ファイルとディスク容量の情報を表示します。	df コマンドを使用して、ディスク容量の利用状況に関する情報を表示します。	81 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」
ファイルのサイズを表示します。	ls コマンドと -lh オプションを使用して、ファイルのサイズに関する情報を表示します。	84 ページの「ファイルサイズを表示する方法」
サイズの大きなファイルを探します。	ls -s コマンドを使用して、ファイルをサイズの降順にソートします。	85 ページの「サイズの大きなファイルを見つける方法」
指定されたサイズ制限を超えるファイルを見つけます。	find コマンドと -size オプションを使用し、サイズ制限の値を指定して、そのサイズ制限を超えるファイルを見つけ、名前を表示します。	86 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」
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します。	du コマンドを使用して、1つ以上のディレクトリ、サブディレクトリ、およびファイルのサイズを表示する	87 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」

タスク	説明	参照先
ローカル UFS ファイルシステムの所有状況を表示します。	<code>quot -a</code> コマンドを使用して、ファイルの所有状況を表示します。	88 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」
最新ファイルのリストを表示します。	<code>ls -t</code> コマンドを使用して、もっと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。	90 ページの「最新ファイルのリストを表示する方法」
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します。	<code>find</code> コマンドと <code>-atime</code> および <code>-mtime</code> オプションを使用して、指定された日数アクセスされていないファイルを見つけます。これらのファイルは、 <code>rm 'cat filename'</code> コマンドを使用して削除できます。	90 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」
一時ディレクトリを一度にクリアします。	一時ディレクトリを見つけ、 <code>rm -r *</code> コマンドを使用してディレクトリ全体を削除します。	91 ページの「一時ディレクトリを一度にクリアする方法」
コアファイルを見つけて削除します。	<code>find . -name core -exec rm {} \;</code> コマンドを使用し、コアファイルを見つけて削除します。	92 ページの「コアファイルを見つけて削除する方法」
クラッシュダンプファイルを削除します。	<code>rm *</code> コマンドを使用して、 <code>/var/crash/</code> ディレクトリにあるクラッシュダンプファイルを削除します。	93 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

次の表に、ファイルのサイズとディスク容量の情報を表示するコマンドを示します。

コマンド	説明	マニュアルページ
<code>df</code>	空きディスクブロック数とファイル数を表示します。	df(1M)
<code>du</code>	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します。	du(1)

コマンド	説明	マニュアルページ
<code>find -size</code>	<code>-size</code> 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します。	find(1)
<code>ls -lh</code>	ファイルのサイズを 1024 の累乗でリストします。	ls(1)

▼ 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]
```

`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します。

`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します。

`-h` ディスク容量を 1024 の累乗で表示します。

`-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します。

例 6-1 ファイルサイズと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、リストされているすべてのファイルシステムがローカルにマウントされています。

```
$ df
/ (/dev/dsk/c0t0d0s0 ): 101294 blocks 105480 files
/devices (/devices ): 0 blocks 0 files
/system/contract (ctfs ): 0 blocks 2147483578 files
/proc (proc ): 0 blocks 1871 files
/etc/mnttab (mnttab ): 0 blocks 0 files
/etc/svc/volatile (swap ): 992704 blocks 16964 files
/system/object (objfs ): 0 blocks 2147483530 files
/usr (/dev/dsk/c0t0d0s6 ): 503774 blocks 299189 files
/dev/fd (fd ): 0 blocks 0 files
/var/run (swap ): 992704 blocks 16964 files
/tmp (swap ): 992704 blocks 16964 files
/opt (/dev/dsk/c0t0d0s5 ): 23914 blocks 6947 files
/export/home (/dev/dsk/c0t0d0s7 ): 16810 blocks 7160 files
```

例 6-2 UFS ルートファイルシステムを使用したシステムのファイルサイズ情報を 1024 バイト単位で表示する

次の例では、UFS ルートファイルシステムを使用したシステムの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        249M  200M   25M    90%        /
/devices                  0K    0K    0K     0%        /devices
ctfs                      0K    0K    0K     0%        /system/contract
proc                     0K    0K    0K     0%        /proc
mnttab                    0K    0K    0K     0%        /etc/mnttab
swap                     485M  376K  485M    1%        /etc/svc/volatile
objfs                     0K    0K    0K     0%        /system/object
/dev/dsk/c0t0d0s6        3.2G  2.9G  214M   94%        /usr
fd                        0K    0K    0K     0%        /dev/fd
swap                     485M  40K   485M    1%        /var/run
swap                     485M  40K   485M    1%        /tmp
/dev/dsk/c0t0d0s5         13M  1.7M   10M   15%        /opt
/dev/dsk/c0t0d0s7         9.2M  1.0M  7.3M   13%        /export/home
```

/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。

例 6-3 ZFS ルートファイルシステムを使用したシステムのファイルサイズ情報を 1024 バイト単位で表示する

次の例では、Oracle Solaris ZFS ルートファイルシステムを使用したシステムの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。

```
Filesystem                size  used  avail capacity  Mounted on
rpool/ROOT/s1008be       67G  4.6G  58G     8%        /
/devices                  0K    0K    0K     0%        /devices
ctfs                      0K    0K    0K     0%        /system/contract
proc                     0K    0K    0K     0%        /proc
mnttab                    0K    0K    0K     0%        /etc/mnttab
swap                     1.9G  1.5M  1.9G    1%        /etc/svc/volatile
objfs                     0K    0K    0K     0%        /system/object
sharefs                   0K    0K    0K     0%        /etc/dfs/sharetab
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcapl.so.1
63G  4.6G  58G     8% /platform/sun4u-us3/lib/libc_psr.so.1
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcapl.so.1
63G  4.6G  58G     8% /platform/sun4u-us3/lib/sparcv9/libc_psr.so.1
fd                        0K    0K    0K     0%        /dev/fd
rpool/ROOT/s1008be/var   67G  73M   58G    1%        /var
swap                     1.9G  32K   1.9G    1%        /tmp
swap                     1.9G  40K   1.9G    1%        /var/run
rpool/export             67G  20K   58G    1%        /export
rpool/export/home       67G  18K   58G    1%        /export/home
```

例 6-4 ファイルシステムに割り当てられたブロックとファイルの合計数を表示する

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 ): 101294 blocks 105480 files
                  total: 509932 blocks 129024 files
/devices         (/devices      ):      0 blocks      0 files
                  total:      0 blocks      113 files
/system/contract (ctfs        ):      0 blocks 2147483578 files
                  total:      0 blocks      69 files
/proc            (proc        ):      0 blocks      1871 files
                  total:      0 blocks      1916 files
/etc/mnttab      (mnttab      ):      0 blocks      0 files
                  total:      0 blocks      1 files
/etc/svc/volatile (swap       ): 992608 blocks 16964 files
                  total: 993360 blocks 17025 files
/system/object   (objfs       ):      0 blocks 2147483530 files
                  total:      0 blocks      117 files
/usr             (/dev/dsk/c0t0d0s6 ): 503774 blocks 299189 files
                  total: 6650604 blocks 420480 files
/dev/fd          (fd          ):      0 blocks      0 files
                  total:      0 blocks      31 files
/var/run         (swap       ): 992608 blocks 16964 files
                  total: 992688 blocks 17025 files
/tmp             (swap       ): 992608 blocks 16964 files
                  total: 992688 blocks 17025 files
/opt            (/dev/dsk/c0t0d0s5 ): 23914 blocks 6947 files
                  total: 27404 blocks 7168 files
/export/home     (/dev/dsk/c0t0d0s7 ): 16810 blocks 7160 files
                  total: 18900 blocks 7168 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\(1\)](#) および [find\(1\)](#) のマニュアルページを参照してください。

注 - /var ディレクトリの容量がなくなった場合、/var ディレクトリに、ファイルシステム上の空き容量のあるディレクトリへのシンボリックリンクを設定しないでください。たとえそれが一時的な策であっても、デーモンプロセスおよびユーティリティーの一部に問題が生じることがあります。

▼ 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します。(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します。このオプションは、`-o`、`-n`、`-@`、`-g` オプションによる出力も、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よう変更します。詳細は、[ls\(1\)](#) のマニュアルページ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します。

例 6-5 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 検索したいディレクトリに移動します。
- 2 ファイルのサイズをブロック数でもっとも大きいものから降順に表示します。
 - ファイルの文字またはカラムが異なる場合、次のコマンドを使用して、ブロックサイズによりもっとも大きいものから降順で、ファイルのリストをソートします。

```
$ ls -l | sort +4rn | more
```

このコマンドは、左から4番目のフィールドにある文字により、リスト内のファイルをソートすることに注意してください。

- ファイルの文字またはカラムが同じである場合、次のコマンドを使用して、ブロックサイズによりもっとも大きいものから降順で、ファイルのリストをソートします。

```
$ ls -s | sort -nr | more
```

このコマンドは、もっとも左側の文字から始め、リスト内のファイルをソートすることに注意してください。

例6-6 サイズの大きなファイルを見つける(5番目のフィールドの文字によるソート)

```
$ cd /var/adm
$ ls -l | sort +4rn | more
-r--r--r-- 1 root  root  4568368 Oct 17 08:36 lastlog
-rw-r--r-- 1 adm   adm   697040 Oct 17 12:30 pacct.9
-rw-r--r-- 1 adm   adm   280520 Oct 17 13:05 pacct.2
-rw-r--r-- 1 adm   adm   277360 Oct 17 12:55 pacct.4
-rw-r--r-- 1 adm   adm   264080 Oct 17 12:45 pacct.6
-rw-r--r-- 1 adm   adm   255840 Oct 17 12:40 pacct.7
-rw-r--r-- 1 adm   adm   254120 Oct 17 13:10 pacct.1
-rw-r--r-- 1 adm   adm   250360 Oct 17 12:25 pacct.10
-rw-r--r-- 1 adm   adm   248880 Oct 17 13:00 pacct.3
-rw-r--r-- 1 adm   adm   247200 Oct 17 12:35 pacct.8
-rw-r--r-- 1 adm   adm   246720 Oct 17 13:15 pacct.0
-rw-r--r-- 1 adm   adm   245920 Oct 17 12:50 pacct.5
-rw-r--r-- 1 root  root   190229 Oct 5 03:02 messages.1
-rw-r--r-- 1 adm   adm   156800 Oct 17 13:17 pacct
-rw-r--r-- 1 adm   adm   129084 Oct 17 08:36 wttmpx
```

例6-7 サイズの大きなファイルを見つける(もっとも左側の文字によるソート)

次の例では、lastlog と messages ファイルが /var/adm 内でもっと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
```

directory 検索するディレクトリを指定します。

-size +nnn 512バイトブロック数です。このサイズを上回るファイルがリストされます。

例 6-8 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で 400 ブロックを超えるファイルを見つける方法を示します。-print オプションは、find コマンドの出力を表示します。

```
$ 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すると、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\(1\)](#) と [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

- du コマンドを使用して、1つ以上のディレクトリ、サブディレクトリ、およびファイルのサイズを表示するサイズは 512 バイトブロック単位で表示されます。

```
$ du [-as] [directory...]
```

du 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表示します。

-a 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します。

-s 指定したディレクトリ内の合計ブロック数を表示します。

-h ディレクトリのサイズを 1024 バイト単位のブロック数で表示します。

-H ディレクトリのサイズを 1000 バイト単位のブロック数で表示します。

[directory...] 調べたい 1 つ以上のディレクトリを指定します。複数のディレクトリを指定する場合は、コマンド行構文で空白を使用して区切ります。

例 6-9 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を示しています。各ディレクトリ内の合計ブロック数も表示しています。

```
$ du /var/adm /var/spool/lp
2    /var/adm/exacct
2    /var/adm/log
2    /var/adm/streams
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
258  /var/adm
4    /var/spool/lp/admins
2    /var/spool/lp/requests/printing....
4    /var/spool/lp/requests
4    /var/spool/lp/system
2    /var/spool/lp/fifos
24   /var/spool/lp
```

次の例は、ディレクトリのサイズを1024バイト単位のブロック数で示しています。

```
$ 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 次のように入力して、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します。

filesystem UFSファイルシステムを指定します。このファイルシステムについて、ユーザーと使用されているブロック数が表示されます。

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す。

例 6-10 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。その次に、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/rdisk/c0t0d0s0 (/):
43340  root
3150  rimmer
47    uucp
35    lp
30    adm
4     bin
4     daemon
/dev/rdisk/c0t0d0s6 (/usr):
460651 root
206632 bin
791   uucp
46    lp
4     daemon
1     adm
/dev/rdisk/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

負荷の大きい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` コマンドまたは `find` コマンドを使用して見つけることができます。詳細については、[ls\(1\)](#) および [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には、`/var/tmp` または `/var/spool` 内にある一時ディレクトリを空にしたり、コア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の方法もあります。クラッシュダンプファイルの詳細は、[第 17 章「システムクラッシュ情報の管理\(タスク\)」](#)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

- `ls -t` コマンドを使用して、もっと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します。

`directory` 検索するディレクトリを指定します。

例 6-11 最新ファイルのリストを表示する

次の例では、`ls -tl` コマンドを使用して、`/var/adm` ディレクトリ内でもっとも新しく作成または変更されたファイルを見つける方法を示します。`sulog` がもっと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 uuucp  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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 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します。このディレクトリの下にあるディレクトリも検索します。

`-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ます。

`-mtime +nnn` 指定した日数 (*nnn*) の間変更のないファイルを見つけます。

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します。

- 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ファイルです。このファイルには、使用されていないファイルのリストが入っています。

例 6-12 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

次の例は、`/var/adm` ディレクトリ内とそのサブディレクトリ内で過去 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るか、同等の役割を引き受けます。

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

- 2 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 サブディレクトリやファイルで、不要なもの、一時的なもの、または古くなったものがある他のディレクトリに移動します。
- 5 手順3を繰り返してこれらのサブディレクトリやファイルを削除します。

例 6-13 一時ディレクトリを一度にクリアする

次の例は、mywork ディレクトリを空にする方法、および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
#
```

▼ コア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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[RBACの構成\(作業マップ\)](#)」を参照してください。
- 2 コアファイルを検索したいディレクトリに移動します。
- 3 ディレクトリとサブディレクトリ内のすべてのコアファイルを見つけて削除します。

```
# find . -name core -exec rm {} \;
```

例 6-14 コアファイルを見つけて削除する

次の例は、find コマンドを使用して jones のユーザーアカウントからコアファイルを見つけて削除する方法を示します。

```
# 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。これらのファイルを保存するようシステムで設定している場合は、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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 次のように入力して、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たことを確認します。

```
# ls
```

例 6-15 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* からクラッシュダンプファイルを削除する方法、およびクラッシュダンプファイルが削除されたことを確認する方法を示します。

```
# cd /var/crash/venus
# rm *
# ls
```


UFS 割り当て制限の管理 (タスク)

この章では、ディスク容量と i ノードに対する UFS 割り当て制限を設定し管理する方法を示します。

UFS 割り当て制限の管理については、以下を参照してください。

- [98 ページの「UFS 割り当て制限の設定 \(タスクマップ\)」](#)
- [103 ページの「UFS 割り当て制限の管理 \(タスクマップ\)」](#)

Oracle Solaris ZFS 割り当て制限の管理については、『[Oracle Solaris ZFS 管理ガイド](#)』の「[ZFS の割り当て制限と予約を設定する](#)」を参照してください。

UFS 割り当て制限

UFS 割り当て制限を使用すると、システム管理者は、ファイルシステムのサイズを制御できます。割り当て制限は、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。通常、「公共な」ファイルシステムと /tmp ファイルシステムについては、ディスク割り当て制限の使用による利点はあまりありません。

UFS 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。割り当て制限やそれを超えることができる時間を変更する方法、各割り当て制限を無効にする方法、またはファイルシステムの割り当て制限を削除する方法については、[106 ページの「UFS 割り当て制限の変更と削除」](#)を参照してください。

さらに、割り当てのステータスを監視できます。UFS 割り当て制限の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、104 ページの「UFS 割り当て制限のチェック」を参照してください。

UFS 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、割り当て制限タイマーが起動します。割り当て制限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ることはできません。

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つのリソースを提供します。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。データブロックは、通常は、1K バイトブロックで構成されます。

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。

UFS 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. `/etc/vfstab` ファイルエントリに割り当て制限オプションを追加し、システムをリブートするたびに割り当て制限が適用されるようにします。また、ファイルシステムの一番上のディレクトリに `quotas` ファイルを作成します。
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. 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。

これらの手順の具体的な情報については、98 ページの「UFS 割り当て制限の設定 (タスクマップ)」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 7-1 UFS 割り当て制限を設定するコマンド

コマンド	タスク	マニュアルページ
<code>edquota</code>	各ユーザーの i ノード数とディスク容量に強い制限値と弱い制限値を設定します。	<code>edquota(1M)</code>
<code>quotacheck</code>	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ます。このあと、不整合を解決します。	<code>quotacheck(1M)</code>
<code>quotaon</code>	指定したファイルシステムの割り当て制限を有効にします。	<code>quotaon(1M)</code>
<code>quota</code>	マウントされているファイルシステムのユーザーの UFS 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します。	<code>quota(1M)</code>

UFS 割り当て制限の設定のガイドライン

UFS 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 i ノード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。

全ユーザーが同時に割り当て制限を超えるような可能性が小さい環境では、個々の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に、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したら、これをプロトタイプとして利用して、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を設定できます。

割り当て制限を有効にする前に、以下の作業を行ないます。

- まず UFS ファイルシステムの割り当て制限を構成します。
- 各ユーザーの割り当て制限を設定して、`quotacheck` コマンドを実行し、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。
- システムのリブートの頻度が低い場合は、`quotacheck` コマンドを定期的に行います。

`edquota` コマンド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効力を持ちません。割り当て制限ファイルを正しく構成すれば、システムをリブートしてファイルシステムがマウントされるたびに、割り当て制限は自動的に有効になります。

UFS 割り当て制限の設定(タスクマップ)

タスク	説明	参照先
1. ファイルシステムの割り当て制限を構成します。	<code>/etc/vfstab</code> ファイルを編集して、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します。また、 <code>quotas</code> ファイルを作成します。	99 ページの「ファイルシステムに UFS 割り当て制限を構成する方法」
2. 1 ユーザー用の UFS 割り当て制限を設定します。	<code>edquota</code> コマンドを使用して 1 ユーザーアカウント用にディスクと <code>i</code> ノードの割り当て制限を行います。	100 ページの「ユーザーに UFS 割り当て制限を設定する方法」
3.(オプション) 複数ユーザーの UFS 割り当て制限を設定します。	<code>edquota</code> 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します。	101 ページの「複数ユーザーに UFS 割り当て制限を設定する方法」

タスク	説明	参照先
4. 整合性をチェックします。	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します。	101 ページの「UFS 割り当て制限の整合性を確認する方法」
5. UFS 割り当て制限を有効にします。	quotaon コマンドを使用して、1 つまたは複数のファイルシステムの UFS 割り当て制限を有効にします。	102 ページの「UFS 割り当て制限を有効にする方法」

▼ ファイルシステムに **UFS** 割り当て制限を構成する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。
- 2 `/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 `quotas` というファイルを作成します。
`# touch quotas`
- 5 スーパーユーザーにのみ、読み取り/書き込み権限を与えます。
`# chmod 600 quotas`

例 7-1 ファイルシステムに UFS 割り当て制限を構成する

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。割り当て制限が有効であることが、`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 `/etc/vfstab` ファイルの例は、ローカルに `/work` ディレクトリがマウントされており、割り当て制限が有効 (`mount options` 列の `rq` エントリで示される) であ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
```

- 参照
- 100 ページの「ユーザーに UFS 割り当て制限を設定する方法」
 - 101 ページの「複数ユーザーに UFS 割り当て制限を設定する方法」
 - 101 ページの「UFS 割り当て制限の整合性を確認する方法」
 - 102 ページの「UFS 割り当て制限を有効にする方法」

▼ ユーザーに UFS 割り当て制限を設定する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 割り当て制限エディタを使用して、一時ファイルを作成します。このファイルには、ファイルシステムのルートディレクトリに `quotas` ファイルがあるマウント済み UFS ファイルシステム 1 つにつき 1 行の、割り当て制限情報が含まれます。
`# 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の両方を、各ファイルシステム用に指定する割り当て制限に変更します。
- 4 i ノード数の弱い制限値と強い制限値の両方を、デフォルト設定の 0 から各ファイルシステム用に指定する割り当て制限に変更します。
- 5 ユーザーの UFS 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します。
```

```
username   ディスク割り当て制限を表示するユーザー名を指定します。
```

例 7-2 ユーザーに UFS 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、上記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 UFS 割り当て制限を設定する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[RBAC の構成\(作業マップ\)](#)」を参照してください。
- 2 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です。

`username ...` 1 人以上の追加アカウントのユーザー名を指定します。複数のユーザー名を指定するには、スペースでユーザー名を区切ります。

例 7-3 複数ユーザーにプロトタイプ UFS 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` および `john` に適用する方法を示しています。

```
# edquota -p bob mary john
```

▼ UFS 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` コマンドを手動で実行するとき、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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 次のように 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          (オプション) 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します。
```

```
-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します。
```

```
filesystem  チェックするファイルシステムを指定します。
```

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

例 7-4 UFS 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)
```

▼ UFS 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

2 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します。

-a /etc/vfstab ファイル内に rq 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します。

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します。複数のファイルシステムを指定する場合は、スペースでファイルシステム名を区切ります。

例 7-5 UFS 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
```

UFS 割り当て制限の管理(タスクマップ)

タスク	説明	参照先
UFS 割り当て制限の超過をチェックします。	quota コマンドを使用して、UFS 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 UFS 割り当て制限とディスク使用量を表示します。	104 ページの「UFS 割り当て制限を超過したかどうかをチェックする方法」
ファイルシステム上の UFS 割り当て制限をチェックします。	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 UFS 割り当て制限とディスク使用量を表示します。	105 ページの「ファイルシステムの UFS 割り当て制限をチェックする方法」
弱い制限値のデフォルトを変更します。	edquota コマンドを使用して、ディスク容量または i ノードの割り当て制限をユーザーが超えることができる時間を変更します。	107 ページの「弱い制限値のデフォルトを変更する方法」

タスク	説明	参照先
ユーザーの UFS 割り当て制限を変更します。	割り当て制限エディタ <code>edquota</code> を使用し、個々のユーザーの割り当て制限を変更します。	108 ページの「1 ユーザーの UFS 割り当て制限を変更する方法」
ユーザーの UFS 割り当て制限を無効にします。	割り当て制限エディタ <code>edquota</code> を使用し、個々のユーザーの割り当て制限を無効にします。	109 ページの「1 ユーザーの UFS 割り当て制限を無効にする方法」
UFS 割り当て制限を無効にします。	<code>quotaoff</code> コマンドを使用し、UFS 割り当て制限を無効にします。	110 ページの「UFS 割り当て制限を無効にする方法」

UFS 割り当て制限のチェック

UFS 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 7-2 UFS 割り当て制限をチェックするコマンド

コマンド	タスク
<code>quota(1M)</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します。ユーザーの割り当て制限超過使用量も表示できます
<code>repquota(1M)</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します

▼ UFS 割り当て制限を超過したかどうかをチェックする方法

`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 UFS 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 次のように入力して、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します。

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) です。

例 7-6 UFS 割り当て制限を超過したかどうかをチェックする

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です。
usage 現在のブロック使用数です。
quota 弱いブロック制限値です。
limit 強いブロック制限値です。
timeleft ディスク割り当て制限タイマーの残り時間(日単位)です。
files 現在のiノード使用数です。
quota 弱いiノード制限値です。
limit 強いiノード制限値です。
timeleft ディスク割り当て制限タイマーの残り時間(日単位)です。
```

▼ ファイルシステムの UFS 割り当て制限をチェックする方法

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 UFS 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 2 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 UFS 割り当て制限を表示します。

```
# repquota [-v] -a filesystem
```

-v リソース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 UFS ディスク割り当て制限を報告します。

-a すべてのファイルシステムについて報告します。

filesystem 指定したファイルシステムについて報告します。

例 7-7 ファイルシステム上の UFS 割り当て制限をチェックする

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     0     2     3
#341  --   57    50   60    7.0 days  2    90   100
```

Block limits 定義

used 現在のブロック使用数です。

soft 弱いブロック制限値です。

hard 強いブロック制限値です。

timeleft ディスク割り当て制限タイマーの残り時間(日単位)です。

File limits ファイル制限値の定義です。

used 現在の i ノード使用数です。

soft 弱い i ノード制限値です。

hard 強い i ノード制限値です。

timeleft ディスク割り当て制限タイマーの残り時間(日単位)です。

UFS 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 7-3 UFS 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
<code>edquota</code>	<code>edquota(1M)</code>	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します。また、割り当て制限を設定した各ファイルシステムの弱い制限値を変更します。
<code>quotaoff</code>	<code>quotaon(1M)</code>	指定したファイルシステムの割り当て制限を無効にします。

▼ 弱い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 UFS 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、ディスク容量または i ノードの割り当て制限をユーザーが超えることのできる期間を変更できます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。
- 2 次のように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編集することを示します。
- 3 期間制限を、**0**(デフォルト)から指定する値に変更します。数値とキーワード `month`、`week`、`day`、`hour`、`min` または `sec` を使用します。

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 7-8 弱い制限値のデフォルトを変更する

次の例は、`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。このシステムでは `/export/home` だけが割り当て制限を持つマウント済みファイルシステムであることを示しています。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週間に変更されたあとの、上の例と同じ一時ファイルの内容を示しています。また、ファイル数の超過に対する期間制限値は16日に変更されています。

```
fs /export/home blocks time limit = 2 weeks, files time limit = 16 days
```

▼ 1 ユーザーの UFS 割り当て制限を変更する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 次のように割り当て制限エディタを使用して、一時ファイルを開きます。このファイルには、ファイルシステムのルートディレクトリに **quotas** ファイルがあるマウント済みファイルシステム1つにつき1行ずつのエントリが含まれます。

```
# 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。ただし、表示される情報がどのユーザーに属するかは示されません。混乱を避けるため、ユーザー名は1つだけ指定してください。

- 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設定します。
- 4 ユーザーの UFS 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 UFS ディスク割り当て制限情報を表示します。

username 割り当て制限をチェックしたいユーザー名を指定します。

例 7-9 1 ユーザーの UFS 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。この一時ファイルのあるシステムでは、ファイルシステムの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)
```

例 7-10 強い UFS 割り当て制限値が変更されたことを確認する

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 UFS 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。
- 2 次のように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。ただし、表示される情報がどのユーザーに属するかは示されません。混乱を避けるため、ユーザー名は 1 つだけ指定してください。

- 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の両方を **0** に変更します。
- 4 i ノード数の弱い制限値と強い制限値の両方を **0** 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。

- 5 ユーザーの UFS 割り当て制限を無効にし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 UFS ディスク割り当て制限情報を表示します。

username UFS 割り当て制限を確認したいユーザー名 (UID) を指定します。

例 7-11 1 ユーザーの UFS 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、ファイルシステムの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、UFS 割り当て制限を無効にしたあとの、上記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)
```

▼ UFS 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るか、同等の役割を引き受けます。

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

- 2 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 ...
```

-v UFS 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します。

-a 全ファイルシステムの UFS 割り当て制限を無効にします。

filesystem 指定する 1 つ以上のファイルシステムの UFS 割り当て制限を無効にします。複数のファイルシステムを指定する場合は、スペースでファイルシステム名を区切ります。

例 7-12 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
```


システムタスクのスケジュール設定 (タスク)

この章では、`crontab` コマンドおよび `at` コマンドを使用して、定型タスクや 1 度限りのシステムタスクをスケジュール設定する方法を説明します。

また、次のファイルを使用して上記のコマンドの使用を制御する方法も説明します。

- `cron.deny`
- `cron-allow`
- `at.deny`

システムタスクのスケジュール設定に関する手順については、以下を参照してください。

- 113 ページの「[crontab ファイルの作成と編集 \(タスクマップ\)](#)」
- 126 ページの「[at コマンドの使用 \(タスクマップ\)](#)」

crontab ファイルの作成と編集 (タスクマップ)

タスク	説明	参照先
crontab ファイルを作成または編集する	<code>crontab -e</code> コマンドを使用し、 <code>crontab</code> ファイルを作成または編集する	119 ページの「 crontab ファイルを作成または編集する方法 」
crontab ファイルが存在するかどうかを確認する	<code>ls -l</code> コマンドを使用し、 <code>/var/spool/cron/crontabs</code> ファイルの内容を確認する	120 ページの「 crontab ファイルを確認する方法 」
crontab ファイルを表示する	<code>ls -l</code> コマンドを使用し、 <code>crontab</code> ファイルを表示する	121 ページの「 crontab ファイルを表示する方法 」

タスク	説明	参照先
crontab ファイルを削除する	crontab ファイルには、アクセス制限がかけられている。したがって、crontab ファイルを削除するには、rm ではなく crontab -r コマンドを使用する	122 ページの「crontab ファイルを削除する方法」
crontab の使用を拒否する	ユーザーによる crontab コマンドの使用を拒否するには、/etc/cron.d/cron.deny ファイルを編集してユーザー名を追加する	124 ページの「crontab コマンドの使用を拒否する方法」
crontab の使用を特定のユーザーに限定する	crontab コマンドのユーザーの使用を許可するには、/etc/cron.d/cron.allow ファイルにユーザー名を追加する	125 ページの「crontab コマンドの使用を特定のユーザーに限定する方法」

システムタスクを自動的に実行する方法

多数のシステムタスクを自動的に実行できるよう設定できます。これらのタスクの中には、定期的な実行が必要になるタスクがあります。また、夜間や週末などの就業時間外に1回だけ実行するタスクもあります。

このセクションでは、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、定型タスク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 コマンドは、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 と at、および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 8-1 コマンドのサマリー: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	アクセスを制御するファイル
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 および /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 Management Console のスケジュールされたジョブツールを使用して、定型タスクをスケジュールすることもできます。Solaris Management Console 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)』の第2章「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` システム管理タスクには、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
- アカウンティングサマリーコマンドを実行する
- `df` コマンドおよび `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ー監視を実行する
- システムのバックアップを実行する

毎週1回の `crontab` システム管理タスクには、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` コマンドを実行して、ディスク問題があれば表示する

毎月1回の `crontab` システム管理タスクには、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、連絡事項の通知やバックアップファイルの削除などの定型的システムタスクを実行するように、`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ルする手順については、119 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)

`at` コマンドを使用すると、1つのジョブをあとで実行するように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` と同様に、`at` コマンドを使用すると定型タスクの自動実行をスケジュールできます。しかし、`crontab` ファイルとは異なり、`at` ファイルはタスクを1回だけ実行します。その後はディレクトリから削除されます。したがって、`at` コマンドが役立つのは、単純なコマンドまたはスクリプトを実行して、別ファイルに書き出した出力をあとから調べるような場合です。

`at` ジョブの実行を指定するには、コマンドを入力してから、`at` コマンド構文に従ってオプションで実行時間を指定してください。`at` ジョブの実行方法については、127 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、起動時に at ジョブをチェックし、新しく実行されるジョブを待機します。cron デーモンが at ジョブを実行すると、atjobs ディレクトリから at ジョブのファイルが削除されます。詳細は、[at\(1\)](#) のマニュアルページを参照してください。

at ジョブをスケジューリングする手順については、[128 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)

以降のセクションで、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リングします。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。このファイルの内容には、次のコマンド行が含まれています。

```
10 3 * * * /usr/sbin/logadm      (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind  (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1      (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean  (4)
```

これらのコマンド行の出力について次に説明します。

- 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。

x86 のみ `/usr/sbin/rtc` スクリプトは、**x86** ベースのシステムでのみ実行できません。

- 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、[118 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。Oracle Solaris 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。次の表を参照してください。

表 8-2 デフォルトの `crontab` ファイル

<code>crontab</code> ファイル	機能
<code>adm</code>	アカウントिंग
<code>lp</code>	印刷
<code>root</code>	一般的なシステム機能とファイルシステムの整理
<code>sys</code>	パフォーマンスデータの収集
<code>uucp</code>	一般的な <code>uucp</code> 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、作成した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の `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、以降のセクションで説明します。

cron デーモンのスケジューリング管理

`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、`/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、起動時に次のタスクを実行します。

- 新しい crontab ファイルがないかを確認する
- ファイル内のリストから実行時間を読み取る
- 正しい時間にコマンドを実行する
- 更新された crontab ファイルに関する crontab コマンドからの通知を待機する

ほとんど同様に、cron デーモンは at ファイルのスケジューリングを制御します。これらのファイルは /var/spool/cron/atjobs ディレクトリに格納されています。cron デーモンは、実行された at ジョブに関する crontab コマンドからの通知も待機します。

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ており、各コマンド行の最初の 5 つのフィールドでは、コマンドが実行される時間を指定します。これら 5 つのフィールドを、次の表で説明します。これらのフィールドはスペース (空白) で区切られます。

表 8-3 crontab 時間フィールドの値

時間フィールド	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 時間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行内に入れる必要があります。crontab ファイルは余分なキャリッジリターンを認識しません。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、システム環境で指定されたテキストエディタを起動します。システム環境のデフォルトのエディタは、`EDITOR` 環境変数で定義されます。この環境変数が設定されていない場合は、`crontab` コマンド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ている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

始める前に `root` または他のユーザーに属する `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るか、同等の役割になります。役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

自分の `crontab` ファイルを編集する場合は、スーパーユーザーになる必要はありません。

- 1 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。自分の `crontab` ファイルを作成するにはスーパーユーザー特権は必要ありませんが、`root` または別のユーザーの `crontab` ファイルを作成したり編集したりするには、スーパーユーザー特権が必要です。



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。

- 2 コマンド行を `crontab` ファイルに追加します。

118 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` ディレクトリに保存されます。

- 3 `crontab` ファイルの変更箇所を確認します。

```
# crontab -l [username]
```

例 8-1 `crontab` ファイルを作成する

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
```

次のコマンドエントリを新しい `crontab` ファイルに追加すると、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が自動的に削除されます。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` のあとに追加されます。このためコマンドが正しく実行されます。

```
# 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、次の出力はユーザー `jones` と `smith` 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。121 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同様に、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、(crontab ファイルが入っている) /var/spool/cron/crontabs ディレクトリに移動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せん。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

始める前に root または他のユーザーの crontab ファイルを表示するには、スーパーユーザーになるか、同等の役割になります。

自分の crontab ファイルを表示するには、スーパーユーザーまたは同等の役割になる必要はありません。

- 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、crontab ファイルを表示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 ファイルを表示するには、スーパーユーザー特権が必要です。



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 8-2 crontab ファイルを表示する

次の例は、crontab -l コマンドを使用してユーザーのデフォルトの crontab ファイルを表示する方法を示します。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

例 8-3 デフォルトの root の crontab ファイルを表示する

次の例は、デフォルトの root の crontab ファイルを表示する方法を示します。

```
$ suPassword:
Oracle Corporation SunOS 5.10 Generic Patch January 2005
# 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.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

例 8-4 他のユーザーの crontab ファイルを表示する

次の例は、他のユーザーの crontab ファイルを表示する方法を示します。

```
$ su
Password:
Oracle Corporation SunOS 5.10 Generic Patch January 2005
# 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 コマンドは自分自身の crontab ファイルを削除します。

このコマンドを使用するために、/var/spool/cron/crontabs ディレクトリ (crontab ファイルが入っている) に移動する必要はありません。

▼ crontab ファイルを削除する方法

始める前に 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るか、同等の役割になります。役割には、認証と特権コマンドが含まれます。

自分の crontab ファイルを削除するには、スーパーユーザーまたは同等の役割になる必要はありません。

- 1 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、crontab ファイルを削除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。

2 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 8-5 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

`/etc/cron.d` ディレクトリ内の2つのファイルを使用して、`crontab` コマンドの使用を制御できます。これらのファイルは `cron.deny` および `cron.allow` です。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` ファイルおよび `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` ファイルを使用できます。
- `cron.allow` も `cron.deny` も存在しない場合は、`crontab` コマンドの実行にスーパーユーザーの権限が必要です。

`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。

SunOS 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。

`cron.allow` のデフォルトファイルは提供されません。つまり、Oracle 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 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 `/etc/cron.d/cron.deny` ファイルを編集し、ユーザー名を 1 行に 1 つずつ追加します。`crontab` コマンドの使用を拒否するユーザーを入れます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

- 3 `/etc/cron.d/cron.deny` ファイルに新しいエントリが含まれているか確認します。

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

▼ crontab コマンド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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 `/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` ファイルに追加します。
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 ユーザー名を1行に1つずつ追加します。
`crontab` コマンドの使用を許可するユーザーを入れます。

```
root
username1
username2
username3
.
.
.
```

例 8-6 crontab コマンドの使用を特定のユーザーに限定する

次は、ユーザー `jones`、`temp`、および `visitor` に `crontab` コマンドを使用させない `cron.deny` ファイルの例です。

```
$ cat /etc/cron.d/cron.deny
daemon
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 `root`、`jones`、`lp` および `smith` 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 コマンドの使用制限を確認する方法

特定のユーザーが `crontab` コマンド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、そのファイルが表示されます。そのユーザーが `crontab` コマンドを使用できるが、`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: can't open your crontab file
```

このユーザーは、`cron.allow` ファイル (が存在する場合) に含まれているか、`cron.deny` ファイル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` ファイルの有無にかかわらず、次のメッセージが表示されます。

```
crontab: you are not authorized to use cron. Sorry.
```

このメッセージは、ユーザーが `cron.allow` ファイル (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。

at コマンドの使用(タスクマップ)

タスク	説明	参照先
at ジョブを作成する	次のような目的には、 <code>at</code> コマンドを使用する。 <ul style="list-style-type: none"> ■ コマンド行から <code>at</code> ユーティリティーを起動する ■ 実行したいコマンドまたはスクリプトを、1行に1つずつ入力する ■ <code>at</code> ユーティリティーを終了し、ジョブを保存する 	128 ページの「 at ジョブを作成する方法 」
at 待ち行列を表示する	<code>atq</code> コマンドを使用し、 <code>at</code> 待ち行列を表示する	129 ページの「 at 待ち行列を表示する方法 」

タスク	説明	参照先
at ジョブを確認する	atq コマンドを使用し、特定のユーザーの at ジョブが待ち行列に入れられたかどうかを確認する	130 ページの「at ジョブを確認する方法」
at ジョブを表示する。	at -l [job-id] コマンドを使用し、待ち行列に入れられた at ジョブを表示する。	130 ページの「at ジョブを表示する方法」
at ジョブを削除する。	at -r [job-id] コマンドを使用し、待ち行列から at ジョブを削除する	130 ページの「at ジョブを削除する方法」
at コマンドの使用を拒否する	at コマンドの使用を拒否するには、 <code>/etc/cron.d/at.deny</code> ファイルを編集する	131 ページの「at コマンドの使用を拒否する方法」

1つのシステムタスクのスケジューリング (at)

以降のセクションでは、at コマンドを使用して以下のタスクを実行する方法を説明します。

- 特定の時間にジョブ(コマンドとスクリプト)の実行をスケジュールする
- これらのジョブを表示および削除する
- at コマンドの使用を制御する

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 ジョブの実行を設定すると、ジョブ識別番号と .a 拡張子が与えられます。これがジョブのファイル名および待ち行列番号になります。

at コマンドの説明

at ジョブファイルを書き込むには、以下の手順を実行します。

1. コマンド実行時間を指定して at ユーティリティを起動します。
2. あと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 ユーザーアカウントからコア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 2004
```

at コマンドの使用制御

特定のユーザーだけが自分の 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]
```

-m ジョブ終了後にメールを送る

time ジョブをスケジュールしたい時刻を指定する。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オプション

date 月または曜日の英語名の最初の3文字以上、またはキーワード today または tomorrow を指定する

- 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** ユーティリティーを終了し、**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** ユーティリティーの終了時に表示されます。

例 8-7 at ジョブを作成する

次の例は、ユーザー `jones` が自分のバックアップファイルを 7:30 pm に削除するように作成した **at** ジョブを示しています。彼女は、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` オプションを使用しています。

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul 12 19:30:00 2004
```

彼女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時に大規模な **at** ジョブをスケジュールする方法を示しています。ジョブの出力先は `big.file` という名前のファイルです。

```
$ 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。

```
$ atq
```

このコマンドは、その使用者が作成した **at** ジョブに関するステータス情報を表示します。

▼ 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。次の例の **atq** コマンドは、**jones** の **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, 2004 19:30   jones     897355800.a   a      stdin
2nd      Jul 14, 2004 23:45   jones     897543900.a   a      stdin
3rd      Jul 17, 2004 04:00   jones     897732000.a   a      stdin
```

▼ at ジョブを表示する方法

- 自分の **at** ジョブの実行時間に関する情報を表示するには、**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、ステータスを表示したいジョブの識別番号を指定します。

例 8-8 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ステータスを表示します。

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2004
897355800.a    Thu Jul 12 19:30:00 2004
897732000.a    Tue Jul 17 04:00:00 2004
```

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 2004
```

▼ at ジョブを削除する方法

始める前に root または他のユーザーの **at** ジョブを削除するには、スーパーユーザーになるか、同等の役割になります。役割には、認証と特権コマンドが含まれます。

自分の **at** ジョブを削除するには、スーパーユーザーまたは同等の役割になる必要はありません。

- 1 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。
- 2 **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

```
$ at -l [job-id]
```

例 8-9 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るか、同等の役割を引き受けます。
 役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 **/etc/cron.d/at.deny** ファイルを編集して、**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つずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
```

```

.
.
.

```

例 8-10 at の使用を拒否する

次の例は、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` ファイルです。

```

$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith

```

▼ at コマンド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` ファイル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。ユーザー `smith` が `at` コマンドにアクセスできない場合、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。

システムアカウンティングの設定と管理 (タスク)

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。

この章の内容は以下のとおりです。

- 134 ページの「システムアカウンティング」
- 140 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の使用については、『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の第 4 章「拡張アカウンティング (概要)」を参照してください。

システムアカウンティングの手順については、139 ページの「システムアカウンティング (タスクマップ)」を参照してください。

システムアカウンティングのさまざまなレポートについての参照情報は、第 10 章「システムアカウンティング (リファレンス)」を参照してください。

システムアカウンティングの新機能

このセクションでは、Oracle Solaris のシステムアカウンティングの新機能、または機能の変更について説明します。Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『Oracle Solaris 10 8/11 の新機能』を参照してください。

Oracle Solaris プロセスアカウンティングと統計の改善点

Oracle Solaris 10: 負荷平均の内部、cpu usr/sys/idle、およびアカウンティング機能が改善されました。マイクロステートアカウンティングが従来のアカウンティング

メカニズムの代わりに使用されるようになり、デフォルトでは常に有効になっています。したがって、プロセス利用と時間の統計において、わずかな相違が見られることがあります。

マイクロステートアカウントティングの採用によって、ユーザープロセスと、さまざまな状態においてユーザープロセスにかかる時間について、より正確なデータを得られます。さらに、この情報を使用することによって、より正確な負荷平均や統計が、`/proc` ファイルシステムから生成されます。詳細は、[proc\(4\)](#) のマニュアルページを参照してください。

システムアカウントティング

Oracle Solaris OS のシステムアカウント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。

システムアカウントティングは、日次または月次ベースで使用できます。また、ユーザー単位のディスク使用量を追跡することもできます。

アカウントティングプログラムを使用すると、次のタスク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ーの維持

システムアカウントティングプログラムは、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ティングの動作

自動アカウントティングは、まずアカウントティ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、アカウントティ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ティ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ティ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ティングファイルを処理して、累積サマリーファイルと日次アカウントティ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、149 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 累積サマリー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サマリーレポートは、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ティングのコンポーネント

システムアカウントティングソフトウェアは、データからサマリー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。アカウントティングレポートは、/var/adm/acct ディレクトリにあります。

日次アカウントティングによって、次の 4 種類の監査を行うことができます。

- 接続アカウントティング
- プロセスアカウントティング
- ディスクアカウントティング
- 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
- システムのリブート回数
- 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

この接続セッション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
- ブート時間
- アカウントティングソフトウェアが有効または無効にされた回数
- 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)
- プロセスの終了

これらのレコードは、date、init、login、ttymon、acctwtmp などのシステムプログラムの出力によって生成されます。これらは /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
- デバイス名
- 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

プロセスアカウントिंग

プロセスアカウントिंगでは、システムで実行される各プロセスに関する次の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
- プロセスの開始時間と経過時間
- プロセスの CPU 時間 (ユーザー時間とシステム時間)
- プロセスによって使用されるメモリーの容量
- プロセスによって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。

ディスクアカウントिंग

ディスクアカウントिंगでは、各ユーザーがディスク上に持っているファイルについて、次のデータを収集しフォーマットできます。

- 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、/usr/lib/acct/dodisk シェルスクリプト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エントリによって決定されます。一方、dodisk スクリプトは、acctdisk コマンドと acctdusg コマンドを起動します。これらのコマンドは、ログイン名ごとのディスク使用量を収集します。



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/diskacct 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 ノードの現在のファイルサイズの値をチェックして、ファイルのサイズを決めます。

料金計算

chargefee ユーティリティーは、ユーザーに提供した特別なサービスに対する課金を、/var/adm/fee ファイルに格納します。特別なサービスとは、たとえば、ファイルの復元です。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方法については、[143 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

日次アカウントング機能がどのように動作するかの段階的なサマリーを示します。

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。

/var/adm 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	バイナリ形式
	変更	date	バイナリ形式
	リポート	acctwtmp	バイナリ形式
	シャットダウン	shutacct	バイナリ形式
pacctn	プロセス数	カーネル(プロセス終了時)	バイナリ形式
		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)	バイナリ形式
fee	特別料金	chargefee	ASCII

<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
<code>acct/nite/diskacct</code>	使用ディスク領域	<code>dodisk</code>	バイナリ形式

3. `turnacct` スクリプトが `-on` オプションで起動されて、プロセスアカウントिंगを開始します。具体的には、`turnacct` スクリプトは、`/var/adm/pacct` 引数を使用して `accton` プログラムを実行します。
4. `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。
5. `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。
6. 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。

7. `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、アカウントングファイル処理し、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。処理されるアカウントングファイルは以下のとおりです。`/var/adm/pacct n`、`/var/adm/wtmpx`、`/var/adm/fee`、および `/var/adm/acct/nite/diskacct`。
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/rprtMMDD` ファイルに日次アカウントング情報を書き込みます。
9. `monacct` スクリプトを月に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) 実行すべきです。`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」を書き込み、プロセスアカウントिंगを無効にします。

システムアカウントिंग(タスクマップ)

タスク	説明	参照先
システムアカウントिंगを設定する	次のタスクを実行し、システムアカウントिंगを設定します。 <ul style="list-style-type: none"> ■ /etc/rc0.d/K22acct ファイルと /etc/rc2.d/S22acct ファイルを作成します。 ■ crontab ファイル /var/spool/cron/crontabs/adm および /var/spool/cron/crontabs/root を変更します。 	141 ページの「システムアカウントिंगを設定する方法」
ユーザーに課金する	/usr/lib/acct/chargefee username amount コマンドを実行します。	143 ページの「ユーザーに課金する方法」
壊れた wtmpx ファイルを修復する	wtmpx ファイルの形式をバイナリから ASCII に変換します。	144 ページの「壊れた wtmpx ファイルの修復方法」
tacct エラーを修復する	prtacct スクリプトを実行し、/var/adm/acct/sum/tacctprev ファイルをチェックします。次に、最新の /var/adm/acct/sum/tacct MMDD ファイルに修正します。/var/adm/acct/sum/tacct ファイルを再作成する必要があります。	145 ページの「tacct エラーを修復する方法」
runacct スクリプトを再起動する	lastdate ファイルとロックファイルを削除します。次に、runacct スクリプトを手動で再起動します。	146 ページの「runacct スクリプトを再起動する方法」
システムアカウントिंगを一時的に停止する	adm の crontab ファイルを編集し、ckpacct、runacct、および monacct プログラムを実行しないようにします。	146 ページの「一時的にシステムアカウントिंगを停止する方法」
システムアカウントिंगを永続的に無効にする	crontab ファイルと adm ファイルの ckpacct、runacct、および monacct プログラムのエントリを削除します。	147 ページの「システムアカウントिंगを永続的に無効にする方法」

システムアカウントिंगの設定

システムアカウントिंगは、システムがマルチユーザーモード (実行レベル2) のときに実行されるように設定できます。このタスクには通常、次の内容が含まれません。

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

次の表に、デフォルトのアカウントिंगスクリプトを説明します。

表9-1 デフォルトのアカウントिंगスクリプト

目的	アカウントングスクリプト	マニュアルページ	実行頻度
/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	ckpacct	acctsh(1M)	定期的
接続、ディスク、および料金アカウントング情報を処理する。このスクリプトから、処理したくないアカウントング機能のコマンドを削除できる	runacct	runacct(1M)	日次
会計アカウントングサマリーレポートを月に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、使用したくないアカウントング機能のコマンドを削除できる	monacct	acctsh(1M)	会計期間に基づく

デフォルトでどのアカウントングスクリプトを実行するのかが選択できます。これらのエントリを crontab ファイルに追加すると、システムアカウントング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を設定する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 必要な場合は、`pkgadd` コマンドを使用して、システムに `SUNWaccr` パッケージと `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。
詳細については、[holidays\(4\)](#) のマニュアルページと、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、次のように入力して、手動でシステムアカウントをブートします。

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

例 9-1 アカウントを設定する (adm の crontab)

この編集された `adm` の `crontab` には、`ckpacct`、`runacct`、および `monacct` スクリプトのエントリが含まれています。

```
#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
```

例9-2 アカウンティングを設定する (root の crontab)

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せん。

```
#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.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 22 * * 4 /usr/lib/acct/dodisk
```

例9-3 アカウンティングを設定する (/etc/acct/holidays)

次に、/etc/acct/holidays ファイルの例を示します。

```
* @(#)holidays January 1, 2004
*
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr Prime Non-Prime
* Year Start Start
*
2004 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

要求に応じて特別なユーザーサービスを提供する場合、chargefee ユーティリティを実行してユーザーに課金することができます。特別なサービスには、ファイルの復元やリモート印刷などがあります。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。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、全体のアカウンティングレコードにマージされます。

詳細については、[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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 特別なサービスに対し、ユーザーに課金します。

```
# /usr/lib/acct/chargefee username amount
```

username 課金したいユーザーアカウントを指定します。

amount ユーザーに対する課金の単位数。この値は、ファイルの印刷や復元のようなタスクに対してユーザーに課金するために設定する任意の単位。
chargefee ユーティリティーを実行し、特定のタスク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 9-4 ユーザーへの課金

次の例では、ユーザー `print_customer` 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

このセクションでは、壊れたシステム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

システムアカウントは絶対に障害が発生しないわけではありません。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、システムアカウント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ときに、システムがマルチユーザーモードになっていると、一連の日付変更レコードが `/var/adm/wtmpx` ファイル

ルに書き込まれます。wtmpfixユーティリティーは、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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 /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 形式に変換します。

```
# /usr/lib/acct/fwtmp < wtmpx > wtmpx.ascii
```
- 4 wtmpx.ascii ファイルを編集して、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
```


詳細については、[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、システムリソースに対してユーザーに課金している場合は重要です。負の数値、重複ユーザー ID、または 65535 のユーザー ID とともに不正な tacct レコードが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 ファイルを表示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/tacct MMDD ファイルに修正します。次に、/var/adm/acct/sum/tacct ファイルを作成し直します。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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 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 形式に変換します。

```
# /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 形式からバイナリ形式に変換します。

```
# /usr/lib/acct/acctmerg -i < xtacct > tacctMMDD
```
- 6 `tacctprev` ファイルと `tacct.MMDD` ファイルをマージして `tacct` ファイルを生成します。

```
# /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。

一般的な理由には、以下のようなものがあります。

- システムクラッシュ
- `/var` ディレクトリの容量不足
- `wtmpx` ファイルの破損

`active.MMDD` 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、その呼び出しがその日の最初の呼び出しであると想定します。`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`

ファイルをオーバーライドするには、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 `/var/adm/acct/nite` ディレクトリに移動します。

```
$ cd /var/adm/acct/nite
```

- 2 `lastdate` ファイルと `lock*` ファイル(もしあれば)を削除します。

```
$ 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` スクリプトを再起動することによって、このファイルが作成し直されます。

- 3 `runacct` スクリプトを再起動する

```
$ /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` スクリプトの処理を開始させたい状態または開始点を指定する。

システムアカウントの停止と無効

システムアカウントは、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トを停止する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

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

- 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
```

- 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、**root** の **crontab** ファイルを編集します。

```
# crontab -e
#30 22 * * 4 /usr/lib/acct/dodisk
```

- システムアカウントプログラムを停止します。

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

- (オプション) 新規に追加したコメント記号を **crontab** ファイルから削除します。

- システム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を永続的に無効にする方法

- スーパーユーザーになるか、同等の役割を引き受けます。

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

- 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
```

- 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
```

- 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 システムアカウントプログラムを停止します。
`# /etc/init.d/acct stop`

システムアカウントिंग (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。

- 149 ページの「runacct スクリプト」
- 152 ページの「日次アカウントングレポート」
- 159 ページの「システムアカウントングファイル」

システムアカウントングのタスクについては、第9章「システムアカウントングの設定と管理 (タスク)」を参照してください。

runacct スクリプト

主な日次アカウントングスクリプトである runacct は、通常の業務時間帯を避けて 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を処理します。さらに、このスクリプトは、課金目的で prdaily スクリプトと monacct スクリプトが使用する、日次および累積のサマリー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。

一連の保護メカニズムにより、次のタスクが実行されます。

- エラーの認識
- インテリジェント診断の提供
- 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な処理

このスクリプトは、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 ファイルおよび lock1 ファイルが作成されます。これらのファイルは、runacct スクリプト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います。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すると、以下が実行されます。

- コンソールにメッセージが出力される
- root と adm に電子メールが送られる
- ロックが解除される
- 診断ファイルが保存される
- 実行が終了する

runacct スクリプトを再起動する手順については、146 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 スクリプトの再起動を可能にするため、処理は分割され、個別にリエントラントな状態になります。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が反映されます。そ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 10-1 runacct スクリプトの状態

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 プロセスアカウントファイル (pacct ファイルは除く) 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 プログラムおよび utmp2wtmp プログラムは、現在ログインしているユーザーのアカウントのために 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

表 10-1 runacct スクリプトの状態 (続き)

状態	説明
CONNECT	acctcon プログラムを使用して、ctacct.MMDD ファイルに接続アカウントレコードを記録する。これらのレコードは tacct.h 形式になっている。acctcon プログラムは、さらに lineuse ファイルおよび reboots ファイルを作成する。reboots ファイルは、wtmptx 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 プログラムを使用して、/var/adm/Spacctn.MMDD プロセスアカウントレコードを ptacctn.MMDD ファイルのアカウント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、プロセスアカウントレコードを接続アカウントレコードとマージして daytacct ファイルを作成する
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 スクリプトが diskacct ファイルを作成する。dodisk スクリプトが実行されて diskacct ファイルが作成されている場合、DISK プログラムはそのファイルを daytacct ファイルにマージし、diskacct ファイルを /tmp/diskacct.MMDD ファイルに移動させる
MERGETACCT	acctmerg プログラムが、daytacct ファイルを累積合計アカウントレコードファイルである 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サマリーを生成し、このデータを sum/daycms ファイルに書き込む。次に、acctcms プログラムは、sum/daycms ファイルと sum/cms 累積コマンドサマリー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 コマンドサマリーファイルを、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 /var/adm/acct/sum/loginlog ログファイルを作成する。このファイルは、各ユーザーが最後にログインした時刻をレポートする。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間を示す日付が 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)アカウントプログラムをこの時点で実行することができる。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に、最後の ptacct ファイルは不完全になるため削除してください。

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。これらのレポートは次の表のとおりです。

表10-2 日次アカウントングレポート

レポートの種類	説明
152 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
153 ページの「日次利用状況レポート」	ユーザー別のシステムリソースの利用状況を示す。ユーザー ID 順に表示される
155 ページの「日次コマンドサマリー」	コマンド別のシステムリソースの利用状況を示す。メモリー使用量が大きい順に表示される。つまり、メモリーをもっと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サマリーで該当月について報告される
156 ページの「月次コマンドサマリー」	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サマリー
156 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。

```
Jan 16 02:30 2004 DAILY REPORT FOR venus Page 1
```

```
from Mon Jan 15 02:30:02 2004
to Tue Jan 16 02:30:01 2004
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、このレポートにはシステムリブート、シャットダウン、電源異常からの回復に関するログ、および、`acctwtmp` プログラムによって `/var/adm/wtmpx` ファイルに書き込まれたその他すべて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) であったかを示します。次の一覧は、日次レポートのデータを説明したものです。

LINE	端末回線またはアクセスポート
MINUTES	アカウントング期間を通じてこの回線が使用中であった時間 (分)
PERCENT	TOTAL DURATION を MINUTES の値で割った数値
# 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ありません。以前は、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が表示されていました。)
# OFF	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します。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 <code>ttymon</code> が初めて起動されたときに発生します。# OFF が # SESS を大きく上回る場合は、マルチプレクサ、モデム、ケーブルに障害がある可能性があります。あるいは、どこかに接続の問題がある可能性があります。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す。

マシンの稼働中は、`/var/adm/wtmpx` ファイルが接続アカウントングの元になるため、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` 回線がもっと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となります。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、[144 ページの「壊れた wtmp 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

日次利用状況レポートは、システムリソース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。

UID	LOGIN NAME	CPU PRIME	(MINS) NPRIME	KCORE-PRIME	MINS NPRIME	CONNECT PRIME	(MINS) NPRIME	DISK BLOCKS	# OF PROCS	# OF SESS	# 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 10-3 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します。
CPU (MINS)	ユーザーのプロセスが CPU を使用した時間を分単位で表します。PRIME (プライムタイム時間帯) と NPRIME (プライムタイム時間帯外) の使用に分類されます。アカウントングシステムのこれらのデータのバージョンは、/etc/acct/holidays ファイルに格納されています。
KCORE-MINS	プロセスが実行中に使用する累積メモリー量を表します。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します。PRIME と NPRIME の使用に分類されます。
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します。これは「実時間」とも呼ばれます。PRIME と NPRIME の使用に分類されます。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ます。
DISK BLOCKS	acctdusg 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 (dayacct) をマージします。アカウントングの目的では、ブロックは 512 バイトです。
# OF PROCS	ユーザーが起動したプロセス数を表します。数値が大きい場合は、ユーザーのシェルプロシージャが制御できなくなった可能性があります。
# 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します。
FEE	chargefee スクリプトによってユーザーに課金された累積合計単位数を表します。多くの場合は、使用されないフィールドです。

日次コマンドサマリー

日次コマンドサマリーレポートは、コマンド別のシステムリソースの利用状況を示します。このレポートでは、使用率の高いコマンドを特定できます。それらのコマンドがどのようにシステムリソースを利用しているかに基づいて、システムの最適なチューニング方法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、システムでの消費量の計算には便利な指標です。

日次コマンドサマリーの例を、次に示します。

COMMAND NAME	NUMBER CMDS	TOTAL COMMAND SUMMARY							
		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
dtsessio	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

次の一覧は、日次コマンドサマリーのデータを説明したものです。

COMMAND NAME	コマンド名。プロセスアカウント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ので、シェルプロシージャはすべてshという名前で取り扱われる。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	このコマンドが実行された回数合計
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転送された文字の合計数。オーバフローのために負の値になることがある
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

月次コマンドサマリー

日次コマンドサマリーと月次コマンドサマリーのレポート形式は、実際は同じものです。ただし、日次コマンドサマリーは現在のアカウントング期間だけでレポートするのに対し、月次コマンドサマリー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サマリーを表します。

月次コマンドサマリーの例を次に示します。

Jan 16 02:30 2004 MONTHLY TOTAL COMMAND SUMMARY Page 1

COMMAND NAME	NUMBER CMDS	TOTAL COMMAND SUMMARY							
		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

月次コマンドサマリーで提供されるデータの説明については、[155 ページの「日次コマンドサマリー」](#)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、最終ログインレポートの例を示します。

Jan 16 02:30 2004 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 形式のレコードを持つ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
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     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
```

次の一覧は、各フィールドを説明したものです。

COMMAND NAME	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)
USER	ユーザー名
TTYNAME	tty 名 (不明の場合は ?)
START TIME	コマンド開始時間
END TIME	コマンド終了時間
REAL (SECS)	実時間 (秒単位)
CPU (SECS)	CPU 時間 (秒単位)

MEAN SIZE (K) 平均サイズ(Kバイト単位)

acctcom コマンドを使用すると、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)
- システム終了ステータス
- hog 係数
- 合計 kcore 分
- CPU 係数
- 転送文字数
- 読み取りブロック数

次の一覧では、acctcom コマンドオプションを解説します。

- 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ステータスカラムを出力する。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。このサイズがデフォルト
- 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 係数を表示する。ユーザー時間/(システム使用時間+ユーザー使用時間)
- 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
- O *sec* CPU システム時間が *sec* 秒を超えるプロセスだけを表示する
- s *time* *time* 以後に存在したプロセスを *hr[: min[: sec]]* の書式で表示する
- S *time* *time* 以後に開始されたプロセスを *hr[: min[: sec]]* の書式で表示する
- u *user* *user* 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。次の一覧は、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。

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> . <i>MMDD</i>	runacct スクリプトの実行中に生成された <i>MMDD</i> 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、および fiscal ディレクトリが設けられています。それぞれのディレクトリ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。/var/adm/acct/nite ディレクトリ内にある各ファイルの簡潔なサマリーは次のとおりです。

表 10-4 /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 ファイルと同じ
cms	prdaily スクリプトが使用する ASCII 合計コマンドサマリー
ctacct.MMDD	tacct.h 形式の接続アカウントングレコード
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で構成される (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 日次コマンドサマリー
daytacct	tacct.h 形式の 1 日分の合計アカウントングレコード
disktacct	tacct.h 形式のディスクアカウントングレコード。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 形式)
lineuse	prdaily スクリプトが使用する tty 回線利用状況レポート
lock	runacct スクリプトの逐次使用の制御に使用する
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あとは、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、累積サマリー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 10-5 /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サマリーファイル
cmsprev	最新の更新がなされていないコマンドサマリーファイル
daycms	内部サマリー書式の、当日の利用状況を表すコマンドサマリーファイル
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。
rppt.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 ファイルと同じ
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サマリー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 10-6 /var/adm/acct/fiscal ディレクトリ内のファイル

ファイル	説明
cmsn	内部サマリー書式の、会計期 <i>n</i> の合計コマンドサマリーファイル
fiscrptn	会計期 <i>n</i> の rpptn 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、runacct スクリプトによって生成される非常に有用なファイルを説明します。これらのファイルは、/var/adm/acct ディレクトリ内にあります。

表 10-7 runacct スクリプトが生成するファイル

ファイル	説明
nite/daytacct	tacct.h 形式の当日の合計アカウントングファイル

表 10-7 runacct スクリプトが生成するファイル (続き)

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。
sum/cms	このファイルは、毎日のコマンドサマリーの累積です。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/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、日次コマンドサマリー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/loginlog	runacct スクリプトは lastlogin スクリプトを呼び出し、/var/adm/acct/sum/loginlog ファイルの各ログインの最終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
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。

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。

- 163 ページの「システムパフォーマンスの管理に関する新機能」
- 164 ページの「システムパフォーマンスタスクに関する情報の参照箇所」
- 165 ページの「システムパフォーマンスおよびシステムリソース」
- 165 ページの「プロセスとシステムのパフォーマンス」
- 167 ページの「システム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

このセクションでは、Oracle Solaris でシステムパフォーマンスを管理するための新機能、または機能の変更について説明します。Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『[Oracle Solaris 10 8/11 の新機能](#)』を参照してください。

強化された `pfiles` ツール

Oracle Solaris 10: `/proc` ファイルシステムは、`/proc/pic/path` ディレクトリのファイル名情報を格納できるよう強化されました。この情報は、`pfiles` がプロセス中の各ファイル名を表示するために使います。この変更によって、プロセスの動作の新しい洞察が可能になります。詳細は、176 ページの「プロセスに関する情報を表示する方法」と、`proc(1)` のマニュアルページを参照してください。

CPU パフォーマンスカウンタ

Oracle Solaris 10: Oracle Solaris OS を実行する SPARC および x86 プラットフォームのパフォーマンス分析機能を使いやすくするため、CPU パフォーマンスカウンタ (CPC) システムが強化されました。

CPC コマンド `cpustat` および `cputrack` で、CPU 情報を指定するためのコマンド行構文が強化されました。たとえば、旧バージョンの Oracle Solaris OS では、2 つのカウンタを指定しなければなりませんでした。以下の例に示すように、今回のバージョンで指定しなければならないカウンタ数は、どちらのコマンドの構成でも 1 つだけです。

```
# cputrack -c pic0=Cycle_cnt ls -d .
time lwp      event      pic0      pic1
.
0.034  1          exit      841167
```

測定を簡素化するため、以下の例に示すように、カウンタ構成を省略することもできます。

```
# cputrack -c Cycle_cnt ls -d .
time lwp      event      pic0      pic1
.
0.016  1          exit      850736
```

`cpustat` コマンドの使用については、[cpustat\(1M\)](#) のマニュアルページを参照してください。 `cputrack` コマンドの使用については、[cputrack\(1\)](#) のマニュアルページを参照してください。

システムパフォーマンスタスクに関する情報の参照箇所

システムパフォーマンスのタスク	詳細
プロセスの管理	第 12 章「システムプロセスの管理 (タスク)」
システムパフォーマンスの監視	第 13 章「システムパフォーマンスの監視 (タスク)」
調整可能パラメータの変更	『Oracle 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タスクの管理	『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の第 2 章「プロジェクトとタスク (概要)」
FX および FS スケジューラを使用したプロセス管理	『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の第 8 章「公平配分スケジューラ (概要)」

システムパフォーマンスおよびシステムリソース

コンピュータシステムのパフォーマンスは、システムがリソースをどのように使用し、割り当てるかによって変わります。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リソースは次のとおりです。

システムリソース	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、システム上のランダムアクセスメモリー (RAM) の容量を示します。

第13章「システムパフォーマンスの監視(タスク)」は、システムの動作とパフォーマンスに関する統計を表示するツールについて説明します。

プロセスとシステムのパフォーマンス

次の表に、プロセスに関連する用語を示します。

表11-1 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
軽量プロセス (LWP)	仮想CPUまたは実行リソース。LWPは、利用できるCPUリソース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

表 11-1 プロセスに関連する用語 (続き)

用語	説明
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

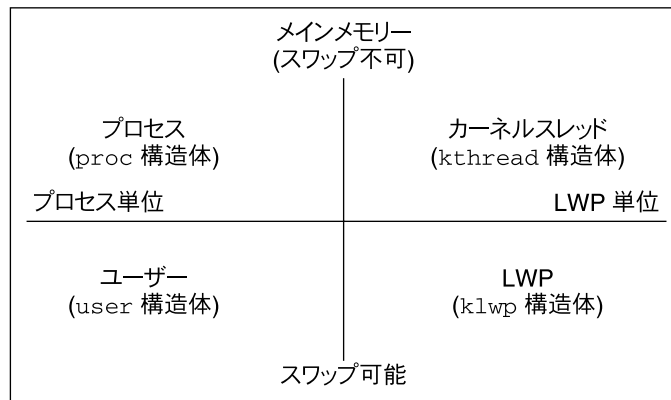
1つのプロセスは、複数のLWPと複数のアプリケーションスレッドで構成できます。カーネルはカーネルスレッド構造をスケジュールします。この構造は、SunOS環境内をスケジュールする実体です。次の表に各種プロセス構造体を示します。

表 11-2 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 11-1 プロセス構造体の関係



プロセス内のすべてのスレッドは、ほとんどのプロセスリソース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せん。

システム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

Oracle 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 11-3 パフォーマンス監視ツール

コマンド	説明	詳細
cpustat と cputrack コマンド	CPU パフォーマンスカウンタを使用し、システムのパフォーマンスまたはプロセスを監視する	cpustat(1M) および cputrack(1)
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) および nfsstat(1M)
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 12 章「システムプロセスの管理 (タスク)」

表 11-3 パフォーマンス監視ツール (続き)

コマンド	説明	詳細
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 13 章「システムパフォーマンスの監視(タスク)」
Sun Enterprise SyMON	Sun のエンタープライズ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』
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)』の第 19 章「追加スワップ空間の構成(手順)」
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 13 章「システムパフォーマンスの監視(タスク)」
cputrack コマンドと cpustat コマンド	マイクロプロセッサが提供するハードウェアパフォーマンスカウンタ機能へのアクセスを支援する	cputrack(1) および cpustat(1M) のマニュアルページ
kstat コマンドと mpstat コマンド	システムで使用可能なカーネル統計(kstats)を検査し、コマンド行で指定された基準に一致する統計を報告する。mpstat コマンドは、プロセッサ統計を表形式で報告する	kstat(1M) および mpstat(1M) のマニュアルページ

◆◆◆ 12

第 12 章

システムプロセスの管理 (タスク)

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。

システムプロセスの管理に関する手順については、以下を参照してください。

- [169 ページの「システムプロセスの管理に関する新機能」](#)
- [170 ページの「システムプロセスの管理 \(タスクマップ\)」](#)
- [181 ページの「プロセスクラス情報の管理 \(タスクマップ\)」](#)

システムプロセスの管理に関する基本情報については、以下を参照してください。

- [171 ページの「システムプロセスを管理するコマンド」](#)
- [181 ページの「プロセスクラス情報の管理」](#)

システムプロセスの管理に関する新機能

擬似カーネルプロセス

Oracle Solaris には、システム上で実行される複数のプロセスが含まれています。これらのプロセスは、特定のタスクを実行しますが、管理の必要はありません。これらのプロセスについて説明している参照用のマニュアルドキュメント (マニュアルページ) はありません。

次の表は、これらの各プロセスについて説明します。

プロセス	説明
fsflush	ページをディスクにフラッシュするシステムデーモン
init	その他のプロセスおよび SMF コンポーネントを起動および再起動する初期システムプロセス

プロセス	説明
intrad	割り込みによるシステム負荷を監視および分散するシステムプロセス
kmem_task	メモリーキャッシュのサイズを監視するシステムプロセス
pageout	ディスクへのメモリーページングを制御するシステムプロセス
sched	OS スケジューリングとプロセススワップを担当するシステムプロセス
vm_tasks	パフォーマンス向上のために仮想メモリー関連のワークロードを複数の CPU に分散する、プロセッサごとに1つのスレッドを持つシステムプロセス
zpool-pool-name	関連するプールに対応する I/O taskq スレッドを含む ZFS ストレージプールごとのシステムプロセス

システムプロセスの管理(タスクマップ)

タスク	説明	参照先
プロセスを表示する	ps コマンドを使用して、システム上のすべてのプロセスを表示する	175 ページの「プロセスを表示する方法」
プロセスについての情報を表示する	pgrep コマンドを使用して、詳細情報を表示したいプロセスのプロセス ID を取得する	176 ページの「プロセスに関する情報を表示する方法」
プロセスを制御する。	pgrep コマンドを使用して、プロセスを見つける。その後、適切な pcommand (/proc) を使用し、プロセスを制御する。(/proc) コマンドについては、表 12-3 を参照	177 ページの「プロセスを制御する方法」
プロセスを強制終了する。	プロセス名かプロセス ID を使用し、プロセスを見つける。pkill コマンドまたは kill コマンドを使用し、プロセスを終了する	178 ページの「プロセスを終了させる方法(pkill)」 179 ページの「プロセスを終了させる方法(kill)」

システムプロセスを管理するコマンド

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 12-1 プロセスを管理するためのコマンド

コマンド	説明	マニュアルページ
ps, pgrep, prstat, pkill	システム上のアクティブなプロセスのステータスをチェックします。また、プロセスについての詳細な情報を表示します。	ps(1) 、 pgrep(1) 、および prstat(1M)
pkill	pgrep と同様に機能するが、名前または他の属性によってプロセスを検索またはシグナルを送信して、プロセスを終了します。一致したプロセスのプロセス ID を出力するのではなく、kill コマンドと同様に、各プロセスにシグナルを送信します。	pgrep(1) および pkill(1) kill(1)
pargs, preap	プロセスのデバッグを支援します。	pargs(1) および preap(1)
dispadmin	デフォルトのプロセススケジューリングポリシーを表示します。	dispadmin(1M)
priocntl	プロセスに優先順位クラスを割り当てて、プロセスの優先度を管理します。	priocntl(1)
nice	タイムシェアリングプロセスの優先度を変更します。	nice(1)
psrset	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します。	psrset(1M)

Solaris Management Console 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でプロセスを管理できます。Solaris Management Console 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)』の第 2 章「Solaris 管理コンソールの操作 (手順)」を参照してください。

ps コマンドの使用

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ステータス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の設定方法を決定するときなど、各種の管理タスクに利用できます。

使用するオプションに応じて、ps コマンドは以下の情報を報告します。

- プロセスの現在のステータス
- 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 12-2 ps により出力されるフィールドのサマリー

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 -c 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、-c 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、スケジュール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

表 12-2 ps により出力されるフィールドのサマリー (続き)

フィールド	説明
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	プロセスの起動時間 (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

/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。次の表に、/proc プロセスコマンドを示します。/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。

表 12-3 プロセスコマンド (/proc)

プロセスコマンド	説明
pcred	プロセスの資格情報を表示する
pfiles	プロセス内で開いているファイルに関する fstat 情報と fcntl 情報を表示する
pflags	/proc 追跡フラグ、保留状態のシグナルと保持状態のシグナル、他のステータス情報を出力する
pldd	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
pmap	各プロセスのアドレス空間マップを表示する
psig	各プロセスのシグナルの動作とハンドラを表示する
prun	各プロセスを開始する
pstack	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
pstop	各プロセスを停止する
ptime	microstate アカウントを使用してプロセスの時間を測定する
ptree	プロセスを含むプロセスツリーを表示する
pwait	プロセス終了後のステータス情報を表示する

表 12-3 プロセスコマンド (/proc) (続き)

プロセスコマンド	説明
pwdx	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、[proc\(1\)](#) を参照してください。

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`、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。
- ユーザーがプロセスを停止または再開できるように、プロセスに対する制御を提供します。

プロセスコマンド (/proc) を使用したプロセスの管理

一部のプロセスコマンドを使用して、プロセスに関する詳細な技術情報を表示したり、アクティブなプロセスを制御したりできます。[表 12-3](#) に一部の /proc コマンドを示しています。

プロセスが無限ループに陥った場合や、実行時間が長すぎる場合は、プロセスを終了 (kill) できます。`kill` コマンドまたは `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 12 章「システムプロセスの管理 \(タスク\)」](#) 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、`xwatchpoint` 機能も提供します。この機能は、プロセスのアドレス空間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 `xwatchpoint` 機能を使用するように変更されています。つまり、`xwatchpoint` プロセス全体がより高速になっています。

`dbx` デバッグ用ツールを使用して `xwatchpoint` を設定する際の、次の制限はなくなりました。

- 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 `xwatchpoint` を設定する。
- マルチスレッド化されたプロセスに `xwatchpoint` を設定する。

詳細については、[proc\(4\)](#) および [mdb\(1\)](#) のマニュアルページを参照してください。

▼ プロセスを表示する方法

- **ps** コマンドを使用すると、システム上の全プロセスを表示できます。

```
$ 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 12-1 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** コマンドの出力を示します。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/keyserd
  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/netshvc/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
```

```

:
:
:

```

▼ 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 (/proc) コマンド。これらのコマンドについては表 12-3 を参照

pid プロセス ID

例 12-2 プロセスに関する情報を表示する

次の例は、プロセスコマンドを使用して cron 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 cron      1
4780
# pwdx 4780      2
4780: /var/spool/cron/atjobs
# ptree 4780     3
4780 /usr/sbin/cron
# pfiles 4780    4
4780: /usr/sbin/cron
Current rlimit: 256 file descriptors
 0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
   O_RDONLY|O_LARGEFILE
   /devices/pseudo/mm@0:null
 1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
   O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
   /var/cron/log
 2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
   O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
   /var/cron/log
 3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
   O_RDWR|O_LARGEFILE

```



```

/etc/cron.d/FIFO
4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
  O_RDWR|O_NONBLOCK
5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
  O_RDWR

```

1. cron プロセスのプロセスIDを表示する。
2. cron プロセスの現在の作業ディレクトリを表示する
3. cron プロセスを含むプロセスツリーを表示する
4. `fstat` と `fcntl` の情報を表示する

▼ 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 (`/proc`) コマンド。これらのコマンドについては [表 12-3](#) を参照

`pid` プロセスID

- 3 プロセスステータスを確認します。

```
# ps -ef | grep pid
```

例 12-3 プロセスを制御する

次の例は、プロセスコマンドを使用して `dtpad` プロセス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` プロセスのプロセス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, kill)

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る場合があります。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、および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、[pgrep\(1\)](#)、[pkill\(1\)](#)、および[kill\(1\)](#)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 (pkill)

- 1 (省略可能) 別のユーザーのプロセスを終了するには、スーパーユーザーになるか、同等の役割になります。

- 2 終了したいプロセスのプロセスIDを表示します。

```
$ pgrep process
```

process は、終了したいプロセスの名前です。

例:

```
$ pgrep netscape
587
566
```

プロセスIDが出力に表示されます。

注 - Sun Ray のプロセス情報を表示するには、次のコマンドを使用します。

```
# ps -fu user
```

このコマンドは、このユーザーのすべてのプロセスを表示します。

```
# ps -fu user | grep process
```

このコマンドは、ユーザーの特定のプロセスを検索します。

- 3 プロセスを終了します。

```
$ pkill [signal] process
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンド

で使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データベースプロセスや LDAP サーバープロセスなどのプロセスを終了するために使用してはならない。データが失われる可能性がある

process 停止するプロセスの名前

ヒント-pkill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。数分待ってプロセスが終了しなければ、pkill コマンドに -9 シグナルを付けて使用します。

- 4 プロセスが終了したことを確認します。

```
$ pgrep process
```

終了したプロセスは、pgrep コマンドの出力に表示されないはずです。

▼ プロセスを終了させる方法 (kill)

- 1 (省略可能)別のユーザーのプロセスを終了するには、スーパーユーザーになるか、同等の役割になります。

- 2 終了したいプロセスのプロセス ID を表示します。

```
$ ps -fu user
```

user は、プロセスを表示したいユーザーです。

プロセス ID は、出力の第 1 列目に表示されます。

- 3 プロセスを終了します。

```
$ kill [signal-number] pid
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンドで使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データベースプロセスや LDAP サーバープロセスなどのプロセスを終了するために使用してはならない。データが失われる可能性がある

pid 終了したいプロセスのプロセス ID

ヒント-kill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。数分待ってプロセスが終了しなければ、kill コマンドに -9 シグナルを付けて使用します。

- 4 プロセスが終了したことを確認します。

```
$ pgrep pid
```

終了したプロセスは、pgrep コマンドの出力に表示されないはずで

プロセスのデバッグ (pargs, preap)

pargs コマンドと preap コマンドは、プロセスのデバッグを改善します。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。ゾンビプロセスとは、その終了ステータスがまだ親に回収されていないプロセスをいいます。これらのプロセスは概して無害ですが、数が多ければシステムリソースを消費します。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、[pargs\(1\)](#)のマニュアルページを参照してください。また、[proc\(1\)](#)のマニュアルページも参照してください。

例 12-4 プロセスをデバッグする (pargs)

pargs コマンドによって、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。次の例は、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する方法を示します。

```
# 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
```

次の例は、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表示する方法を示します。

例 12-4 プロセスをデバッグする (pargs) (続き)

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

プロセスクラス情報の管理 (タスクマップ)

タスク	説明	参照先
プロセスクラスに関する基本情報を表示する	prionctl -l コマンドを使用し、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する	182 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 (prionctl)」
プロセスのグローバル優先順位を表示する	ps -ecl コマンドを使用し、プロセスのグローバル優先順位を表示する	183 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」
プロセスに優先順位を割り当てる	prionctl -e -c コマンドを使用し、割り当てた優先順位でプロセスを開始する	183 ページの「プロセスの優先順位を指定する方法 (prionctl)」
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	prionctl -s -m コマンドを使用し、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	184 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 (prionctl)」
プロセスのクラスを変更する	prionctl -s -c コマンドを使用し、プロセスのクラスを変更する	185 ページの「プロセスのクラスを変更する方法 (prionctl)」
プロセスの優先順位を変更する	/usr/bin/nice コマンドを適切なオプションとともに使用し、プロセスの優先順位を低くしたり高くしたりする	186 ページの「プロセスの優先順位を変更する方法 (nice)」

プロセスクラス情報の管理

次のリストは、システム上で構成可能なプロセススケジューリングクラスを示しています。タイムシェアリングクラスのユーザー優先順位の範囲も示しています。

プロセススケジューリングクラスの種類は次のとおりです。

- 公平配分 (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ます。次にシステムは、nice または `priocntl` (ユーザー指定) 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の変更 (`priocntl`)

プロセスのスケジュール優先順位とは、スケジュールポリシーに従って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。`dispadmin` 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。詳細は、[dispadmin\(1M\)](#) のマニュアルページを参照してください。

`priocntl` コマンドを使用すると、プロセスを優先順位クラスに割り当てたり、プロセスの優先順位を管理したりできます。`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、[183 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` コマンドを使用して、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します。

```
$ priocntl -l
```

例 12-5 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、プロセスのグローバル優先順位を表示します。

```
$ 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 12-6 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 カラム内の値は、`pageout` プロセスが最上位の優先順位を持ち、`sh` プロセス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  CMD
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るか、同等の役割を引き受けます。

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

- 2 指定した優先順位でプロセスを起動します。

```
# priocntl -e -c class -m user-limit -p pri command-name
-e                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(公平配分)、およびFX(固定優先順位)
<code>-m user-limit</code>	<code>-p</code> オプションを使用するとき、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 プロセスステータスを確認します。

```
# ps -ecl | grep command-name
```

例 12-7 プロセスの優先順位を指定する (prctl)

次の例では、ユーザーが指定でき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。

```
# prctl -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 (prctl)

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

- 2 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。

```
# prctl -s -m user-limit [-p user-priority] -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 user-limit</code>	<code>-p</code> オプションを使用するとき、優先順位を上下できる最大範囲を指定する
<code>-p user-priority</code>	優先順位を指定する

`-i xidtype xidlist` `xidtype` と `xidlist` の組み合わせを使用してプロセスを識別する。「`xidtype`」ではプロセス ID やユーザー ID など、ID のタイプを指定する。「`xidlist`」ではプロセス ID またはユーザー ID のリストを識別する

- 3 プロセスステータスを確認します。

```
# ps -ecl | grep idlist
```

例 12-8 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 -m 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) スーパーユーザーまたは同等の役割になります。
- 2 プロセスのクラスを変更する

```
# 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 (タイムシェアリング) または RT (リアルタイム) を指定して、プロセスのクラスを変更する

`-i idtype idlist` `xidtype` と `xidlist` の組み合わせを使用してプロセスを識別する。`xidtype` ではプロセス ID やユーザー ID など、ID のタイプを指定する。「`xidlist`」ではプロセス ID またはユーザー ID のリストを識別する

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。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 `priocntl -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。

- 3 プロセスステータスを確認します。

```
# ps -ecl | grep idlist
```

例 12-9 プロセスのクラスを変更する (priocntl)

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。

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

タイムシェアリングプロセスの優先順位の変更 (nice)

nice コマンドは、Solaris の旧バージョンとの下位互換性を保つためののみサポートされます。priocntl 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、そのスケジューリングポリシーと 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 priocntl 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ます。優先順位番号は、プロセスのスケジューリングクラス、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することができます。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です。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。このコマンドには、利用できるバージョンが 2 つあります。標準バージョンの /usr/bin/nice と、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)

この方法により、ユーザーがプロセスの優先順位を下げるすることができます。ただし、スーパーユーザーはプロセスの優先順位を上げたり、下げたりすることができません。

注 - このセクションでは `/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 シェルの `nice` コマンドについては、[csh\(1\)](#) のマニュアルページを参照してください。

- 1 プロセスの優先順位をユーザーとして変更するか、スーパーユーザーとして変更するかを決定します。次のいずれかの手順に従います。
 - ユーザーとして、手順2の例に従ってコマンドの優先順位を下げます。
 - スーパーユーザーとして、手順3の例に従ってコマンドの優先順位を上げたり下げたりします。
- 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 スーパーユーザーか同等の役割で、**nice** 値を変更してコマンドの優先順位を上げたり下げたりします。

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
```

参照 詳細は、[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トラブルシューティング方法

よくある問題に関する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` 値の高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、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。プロセスの開始時間 (`STIME`) と、その後の CPU 時間 (`TIME`) の累積を調べると、この問題を特定できます。

◆◆◆ 13

第 13 章

システムパフォーマンスの監視(タスク)

この章では、`vmstat`、`iostat`、`df`、および `sar` の各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。

システムパフォーマンスの監視に関する手順については、以下を参照してください。

- 189 ページの「システムパフォーマンス情報の表示(タスクマップ)」
- 197 ページの「システム動作の監視(タスクマップ)」

システムパフォーマンス情報の表示(タスクマップ)

タスク	説明	参照先
仮想メモリーの統計を表示する	<code>vmstat</code> コマンドを使用し、仮想メモリーの統計を収集する	191 ページの「仮想メモリーの統計情報を表示する方法(<code>vmstat</code>)」
システムイベント情報を表示する	<code>vmstat</code> コマンドと <code>-s</code> オプションを使用し、システムイベント情報を表示する	192 ページの「システムイベント情報を表示する方法(<code>vmstat -s</code>)」
スワップの統計を表示する	<code>vmstat</code> コマンドと <code>-S</code> オプションを使用し、スワップの統計を表示する	192 ページの「スワップの統計情報を表示する方法(<code>vmstat -S</code>)」
各デバイス当たりの割り込み数を表示する	<code>vmstat</code> コマンドと <code>-i</code> オプションを使用し、デバイス当たりの割り込み数を表示する	193 ページの「各デバイス当たりの割り込み数を表示する方法(<code>vmstat -i</code>)」
ディスクの使用状況を表示する	<code>iostat</code> コマンドを使用し、ディスクの入出力の統計を報告する	194 ページの「ディスクの使用状況を表示する方法(<code>iostat</code>)」

タスク	説明	参照先
拡張ディスク統計情報を表示する	iostat コマンドと -xtc オプションを使用し、拡張ディスクの統計情報を表示する	195 ページの「拡張ディスク統計情報を表示する方法 (iostat -xtc)」
ディスク容量情報を表示する	df -k コマンドを使用し、ディスク容量情報をキロバイト単位で表示する	196 ページの「ディスク容量情報を表示する方法 (df -k)」

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 のロード、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 コマンドの出力内のフィールドを示します。

表 13-1 vmstat コマンドからの出力

カテゴリ	フィールド名	説明
procs		次の状態を報告します
	r	ディスパッチ待ち行列内のカーネルスレッド数
	b	リソースを待機中のブロックされたカーネルスレッド数
	w	リソース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します
	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 秒当たりの単位数として表示します
	re	回収されたページ数
	mf	軽度の障害と重大な障害
	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

表 13-1 vmstat コマンドからの出力 (続き)

カテゴリ	フィールド名	説明
	de	最後にスワップインされたプロセスに必要なと予想されるメモリー
	sr	page 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、page デーモンが実行されています。
disk		最高 4 台のディスク上のデータを示す、1 秒当たりのディスク処理数を表示します
faults		トラップ/割り込み率 (1 秒当たり) を表示します
	in	1 秒当たりの割り込み数
	sy	1 秒当た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します
	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、仮想メモリの統計情報が収集されます。

```
$ vmstat n
```

n は、秒単位で表した報告間隔です。

例 13-1 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
kthr  memory          page          disk          faults          cpu
 r  b  w  swap  free  re  mf  pi  po  fr  de  sr  dd  f0  s1  --  in  sy  cs  us  sy  id
0  0  0  863160 365680 0   3  1  0  0  0  0  0  0  0  0  406 378 209  1  0  99
0  0  0  765640 208568 0  36  0  0  0  0  0  0  0  0  0  479 4445 1378  3  3  94
0  0  0  765640 208568 0   0  0  0  0  0  0  0  0  0  0  423 214 235  0  0 100
```

```

0 0 0 765712 208640 0 0 0 0 0 0 0 3 0 0 0 412 158 181 0 0 100
0 0 0 765832 208760 0 0 0 0 0 0 0 0 0 0 0 402 157 179 0 0 100
0 0 0 765832 208760 0 0 0 0 0 0 0 0 0 0 0 403 153 182 0 0 100
0 0 0 765832 208760 0 0 0 0 0 0 0 0 0 0 0 402 168 177 0 0 100
0 0 0 765832 208760 0 0 0 0 0 0 0 0 0 0 0 402 153 178 0 0 100
0 0 0 765832 208760 0 18 0 0 0 0 0 0 0 0 0 407 165 186 0 0 100

```

▼ 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あとに発生したシステムイベントの数が表示されます。

```

$ vmstat -s
  0 swap ins
  0 swap outs
  0 pages swapped in
  0 pages swapped out
522586 total address trans. faults taken
17006 page ins
  25 page outs
23361 pages paged in
  28 pages paged out
45594 total reclaims
45592 reclaims from free list
  0 micro (hat) faults
522586 minor (as) faults
16189 major faults
98241 copy-on-write faults
137280 zero fill page faults
45052 pages examined by the clock daemon
  0 revolutions of the clock hand
  26 pages freed by the clock daemon
 2857 forks
  78 vforks
 1647 execs
34673885 cpu context switches
65943468 device interrupts
 711250 traps
63957605 system calls
3523925 total name lookups (cache hits 99%)
 92590 user   cpu
 65952 system cpu
16085832 idle   cpu
 7450 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
kthr  memory          page          disk          faults          cpu
 r b w  swap free  si  so pi po fr de sr dd f0 s1 --  in sy  cs us sy id
 0 0 0 862608 364792 0 0 1 0 0 0 0 0 0 0 406 394 213 1 0 99

```


スワッピング統計情報のフィールドを次のリストに示します。その他のフィールドの説明については、表 13-1 を参照してください。

si 1秒あた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、`si` フィールドおよび `so` フィールドの出力値を切り捨てます。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。

▼ 各デバイス当た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、デバイス当たりの割り込み数が表示されます。

例 13-2 各デバイス当た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`)

`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たり、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、およびサービス時間の計測結果を表示したりできます。このコマンドの詳細については、`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)、およびCPU(cpu)の統計情報が表示されます。

例 13-3 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  100
  0  16   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  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  100
  0  16   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  100
  0  16   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  100
  0  16   3   1  23   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  100
  0  16   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  100
```

次の表に、**iostat n** コマンド出力内のフィールドを示します。

デバイスの種類	フィールド名	説明
端末	デバイスの種類	
	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	デバイスの種類	
	bps	1秒当たりのブロック数
	tps	1秒当たりのトランザクション数

デバイスの種類	フィールド名	説明
	serv	ミリ秒単位で表した平均サービス時間
CPU	デバイスの種類	
	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

▼ 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リストに示します。

```
r/s      1秒当たりの読み取り数
w/s      1秒当たりの書き込み数
kr/s     1秒当たりの読み取りキロバイト数
kw/s     1秒当た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、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できます。また、tar コマンドか cpio コマンドを使用してテープにファイルを転送することもできます。また、ファイルを削除することもできます。

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 -k)

- 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%      /
```

例 13-4 ファイルシステム情報を表示する

次の例は、df -k コマンドの出力です。

```
$ df -k
Filesystem      kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0 254966   204319 25151    90%      /
/devices        0         0       0         0%      /devices
ctfs            0         0       0         0%      /system/contract
proc           0         0       0         0%      /proc
mnttab         0         0       0         0%      /etc/mnttab
swap          496808    376   496432    1%      /etc/svc/volatile
objfs          0         0       0         0%      /system/object
/dev/dsk/c0t0d0s6 3325302 3073415 218634   94%      /usr
fd             0         0       0         0%      /dev/fd
swap          496472    40   496432    1%      /var/run
swap          496472    40   496432    1%      /tmp
/dev/dsk/c0t0d0s5 13702    1745  10587    15%      /opt
/dev/dsk/c0t0d0s7 9450     1045  7460     13%      /export/home
```

次の表に、df -k コマンドの出力を説明します。

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

システム動作の監視(タスクマップ)

タスク	説明	参照先
ファイルアクセスをチェックします。	sar コマンドと <code>-a</code> オプションを使用し、ファイルアクセス操作ステータスを表示します。	199 ページの「ファイルアクセスをチェックする方法 (sar -a)」
バッファ動作をチェックします。	sar コマンドと <code>-b</code> オプションを使用し、バッファ動作の統計情報を表示します。	200 ページの「バッファ動作をチェックする方法 (sar -b)」
システムコールの統計情報をチェックします。	sar コマンドと <code>-c</code> オプションを使用し、システムコールの統計情報を表示します。	201 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)」
ディスク動作をチェックします。	sar コマンドと <code>-d</code> オプションを使用し、ディスク動作をチェックします。	202 ページの「ディスク動作をチェックする方法 (sar -d)」
ページアウトとメモリーをチェックします。	sar コマンドと <code>-g</code> オプションを使用し、ページアウトとメモリー解放動作を表示します。	204 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)」
カーネルメモリーの割り当てをチェックします。	カーネルメモリーの割り当て (KMA) では、カーネルサブシステムが必要に応じてメモリーを割り当てたり解放したりします。sar コマンドと <code>-k</code> オプションを使用し、KMA をチェックします。	206 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)」
プロセス間通信をチェックします。	sar コマンドと <code>-m</code> オプションを使用し、プロセス間通信の動作を報告します。	207 ページの「プロセス間通信をチェックする方法 (sar -m)」
ページイン動作をチェックします。	sar コマンドと <code>-p</code> オプションを使用し、ページイン動作を報告します。	208 ページの「ページイン動作をチェックする方法 (sar -p)」

タスク	説明	参照先
待ち行列動作をチェックします。	<p>sar コマンドと <code>-q</code> オプションを使用し、以下をチェックします。</p> <ul style="list-style-type: none"> ■ 待ち行列に要求が入っている間の平均待ち行列の長さ ■ 待ち行列に要求が入っている時間の割合 	209 ページの「待ち行列動作をチェックする方法 (sar -q)」
未使用メモリーをチェックします。	<p>sar コマンドと <code>-r</code> オプションを使用し、現在使用されている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を表示します。</p>	210 ページの「未使用のメモリーをチェックする方法 (sar -r)」
CPU の使用率をチェックします。	<p>sar コマンドと <code>-u</code> オプションを使用し、CPU 使用率を表示します。</p>	211 ページの「CPU の使用状況をチェックする方法 (sar -u)」
システムテーブルのステータスをチェックします。	<p>以下の sar コマンドと <code>-v</code> オプションを使用し、システムテーブルのステータスをチェックします。</p> <ul style="list-style-type: none"> ■ プロセス ■ i ノード ■ ファイル ■ 共有メモリーレコード 	212 ページの「システムテーブルのステータスをチェックする方法 (sar -v)」
スワップ動作をチェックします。	<p>sar コマンドと <code>-w</code> オプションを使用し、スワップ動作をチェックします。</p>	213 ページの「スワップ動作をチェックする方法 (sar -w)」
端末動作をチェックします。	<p>sar コマンドと <code>-y</code> オプションを使用し、端末デバイスの動作を監視します。</p>	214 ページの「端末動作をチェックする方法 (sar -y)」
システム全体のパフォーマンスをチェックします。	<p>sar -A コマンドを使用し、すべてのオプションを指定した場合と同じように、システム全体のパフォーマンスを示す統計情報を表示します。</p>	216 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)」
データの自動収集を設定します。	<p>システムでデータを自動的に収集するよう設定し、sar コマンドを実行するには、以下を実行します。</p> <ul style="list-style-type: none"> ■ <code>svcadm enable system/sar:default</code> コマンドを実行します ■ <code>/var/spool/cron/crontabs/sys</code> ファイルを編集します 	219 ページの「自動データ収集を設定する方法」

システム動作の監視 (sar)

以下のタスクを実行するには、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します。
- 特別な要求に基づいて、システム動作データにアクセスします。
- 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します。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、特別な要求レポートも生成します。sar コマンドを設定してシステム上で実行する方法と、これらのツールの説明については、[216 ページの「システム動作データの自動収集 \(sar\)」](#)を参照してください。

このコマンドの詳細については、[sar\(1\)](#)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 balmyday 5.10 s10_51 sun4u    03/18/2004

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
```

次に、sar -a コマンドによって表示される、フィールド名とオペレーティングシステムのルーチンを示します。

iget/s ディレクトリ名検索キャッシュ (DNLC) 内に入っていない i ノードに対して出された要求数

namei/s 1 秒当たりのファイルシステムパスの検索数。namei で DNLC 内にディレクトリ名が見つからない場合は、iget が呼び出され、ファイルまたはディレクトリの i ノードが取得される。したがって、ほとんどの igets は DNLC が欠落した結果である

dirbk/s 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

これらのオペレーティングシステムルーチンに対して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、メタデータをキャッシュするために使用されます。メタデータには、iノード、シリンダグループブロック、間接ブロックなどがあります。

```
$ 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
```

例 13-5 バッファー動作をチェックする (sar -b)

次の **sar -b** コマンド出力の例は、%rcache バッファーと %wcache バッファーが処理速度を低下させてい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 balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0      94      0      0
02:00:01      0      0    100      0      0      94      0      0
03:00:00      0      0    100      0      0      92      0      0
04:00:00      0      1    100      0      1      94      0      0
05:00:00      0      0    100      0      0      93      0      0
06:00:00      0      0    100      0      0      93      0      0
07:00:00      0      0    100      0      0      93      0      0
08:00:00      0      0    100      0      0      93      0      0
08:20:00      0      1    100      0      1      94      0      0
08:40:01      0      1    100      0      1      93      0      0
09:00:00      0      1    100      0      1      93      0      0
09:20:00      0      1    100      0      1      93      0      0
09:40:00      0      2    100      0      1      89      0      0
10:00:00      0      9    100      0      5      92      0      0
10:20:00      0      0    100      0      0      68      0      0
10:40:00      0      1     98      0      1      70      0      0
11:00:00      0      1    100      0      1      75      0      0

Average      0      1    100      0      1      91      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。

フィールド名	説明
bread/s	ディスクからバッファークャッシュに投入された1秒当たりの平均読み取り数
lread/s	バッファークャッシュからの1秒当たりの平均論理読み取り数
%rcache	バッファークャッシュ内で見つかった論理読み込み数の割合 (lread/s に対する bread/s の比を 100% から差し引いた値)
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秒当た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秒当た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割合 (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秒当た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秒当たりの平均物理書き込み要求数

もっとも重要なエントリは、キャッシュヒット率 %rcache と %wcache です。これらのエントリは、システムバッファリングの有効性を測定します。%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。

▼ 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
```

例 13-6 システムコールの統計情報をチェックする (sar -c)

次の例は、sar -c コマンドからの出力を示します。

```
$ sar -c

SunOS balmyday 5.10 s10_51 sun4u 03/18/2004

00:00:04 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00 89 14 9 0.01 0.00 2906 2394
```

02:00:01	89	14	9	0.01	0.00	2905	2393
03:00:00	89	14	9	0.01	0.00	2908	2393
04:00:00	90	14	9	0.01	0.00	2912	2393
05:00:00	89	14	9	0.01	0.00	2905	2393
06:00:00	89	14	9	0.01	0.00	2905	2393
07:00:00	89	14	9	0.01	0.00	2905	2393
08:00:00	89	14	9	0.01	0.00	2906	2393
08:20:00	90	14	9	0.01	0.01	2914	2395
08:40:01	90	14	9	0.01	0.00	2914	2396
09:00:00	90	14	9	0.01	0.01	2915	2396
09:20:00	90	14	9	0.01	0.01	2915	2396
09:40:00	880	207	156	0.08	0.08	26671	9290
10:00:00	2020	530	322	0.14	0.13	57675	36393
10:20:00	853	129	75	0.02	0.01	10500	8594
10:40:00	2061	524	450	0.08	0.08	579217	567072
11:00:00	1658	404	350	0.07	0.06	1152916	1144203
Average	302	66	49	0.02	0.01	57842	55544

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がシステムコール合計の約半分を占めます。ただし、システムで実行中の動作によってこの割合は大幅に変動します。

フィールド名	説明
<code>scall/s</code>	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ユーザーから6ユーザーのシステム上で1秒当たり約30)
<code>sread/s</code>	1秒当たりの <code>read</code> システムコール数
<code>swrit/s</code>	1秒当たりの <code>write</code> システムコール数
<code>fork/s</code>	1秒当たりの <code>fork</code> システムコール数 (4ユーザーから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
<code>exec/s</code>	1秒当たりの <code>exec</code> システムコール数。 <code>exec/s</code> を <code>fork/s</code> で割った値が3より大きい場合は、効率の悪い <code>PATH</code> 変数を調べる
<code>rchar/s</code>	<code>read</code> 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
<code>wchar/s</code>	<code>write</code> 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

▼ 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
```

例 13-7 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 balmyday 5.10 s10_51 sun4u    03/18/2004

12:36:32  device          %busy  avque  r+w/s  blks/s  await  avserv
12:40:01  dad1             15     0.7    26     399     18.1   10.0
          dad1,a         15     0.7    26     398     18.1   10.0
          dad1,b         0     0.0    0      1      1.0    3.0
          dad1,c         0     0.0    0      0      0.0    0.0
          dad1,h         0     0.0    0      0      0.0    6.0
          fd0           0     0.0    0      0      0.0    0.0
          nfs1          0     0.0    0      0      0.0    0.0
          nfs2          1     0.0    1     12     0.0   13.2
          nfs3          0     0.0    0      2     0.0    1.9
          nfs4          0     0.0    0      0     0.0    7.0
          nfs5          0     0.0    0      0     0.0   57.1
          nfs6          1     0.0    6    125     4.3    3.2
          nfs7          0     0.0    0      0     0.0    6.0
          sd1           0     0.0    0      0     0.0    5.4
          ohci0,bu       0     0.0    0      0     0.0    0.0
          ohci0,ct       0     0.0    0      0     0.0    0.0
          ohci0,in       0     0.0    7      0     0.0    0.0
          ohci0,is       0     0.0    0      0     0.0    0.0
          ohci0,to       0     0.0    7      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。

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でビジーであった時間
avque	デバイスが転送要求のサービスでビジーであった時間の間の平均要求数
r+w/s	デバイスへの1秒当た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、および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、待ち行列の長さや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。

▼ 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、page 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 フィールドと pgscan/s フィールドの値が大きければ、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のかも表示されます。

例 13-8 ページアウトとメモリーをチェックする (sar -g)

次の例は、sar -g コマンドからの出力を示します。

```
$ sar -g
SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

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
```

次の表に -g オプションからの出力について説明します。

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ります。
pgfree/s	空きリストに配置された1秒当たりのページ数。
pgscan/s	page デモンによって走査された1秒当たりのページ数。この値が大きい場合は、page デ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ます。これは、メモリーを増やす必要があることを示します。
%ufs_ipf	ufs の i ノード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ります。したがって、このフィールドはページフラッシュを伴う igets の割合です。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します。

カーネルメモリーの割り当て (KMA) のチェック

KMA を使用すると、カーネルサブシステムが必要に応じてメモリーを割り当てたり、解放したりできます。

KMA では、ロードのピーク時に必要と思われる最大メモリー容量を静的に割り当てることをせず、メモリーの要求を次の3つのカテゴリに分けます。

- 「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)

KMA は、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 リソース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 リソースを使用するが、終了前には特にリソース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。

▼ 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 に関して次の動作が表示されます。

```
$ 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
```

例 13-9 カーネルメモリーの割り当てをチェックする(sar -k)

次の例は、sar -k 出力を示します。

```
$ sar -k
SunOS balmyday 5.10 s10_51 sun4u 03/18/2004

00:00:04 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 6119744 4852865 0 60243968 54334808 156 9666560 0
02:00:01 6119744 4853057 0 60243968 54336088 156 9666560 0
03:00:00 6119744 4853297 0 60243968 54335760 156 9666560 0
04:00:00 6119744 4857673 0 60252160 54375280 156 9666560 0
05:00:00 6119744 4858097 0 60252160 54376240 156 9666560 0
06:00:00 6119744 4858289 0 60252160 54375608 156 9666560 0
07:00:00 6119744 4858793 0 60252160 54442424 156 9666560 0
08:00:00 6119744 4858985 0 60252160 54474552 156 9666560 0
08:20:00 6119744 4858169 0 60252160 54377400 156 9666560 0
08:40:01 6119744 4857345 0 60252160 54376880 156 9666560 0
09:00:00 6119744 4859433 0 60252160 54539752 156 9666560 0
09:20:00 6119744 4858633 0 60252160 54410920 156 9666560 0
09:40:00 6127936 5262064 0 60530688 55619816 156 9666560 0
10:00:00 6545728 5823137 0 62996480 58391136 156 9666560 0
10:20:00 6545728 5758997 0 62996480 57907400 156 9666560 0
10:40:00 6734144 6035759 0 64389120 59743064 156 10493952 0
11:00:00 6996288 6394872 0 65437696 60935936 156 10493952 0

Average 6258044 5150556 0 61138340 55609004 156 9763900 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。

フィールド名	説明
sml_mem	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 (このプールでは、小要求は 256 バイト未満)
alloc	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

フィールド名	説明
fail	失敗した小メモリー要求数
lg_mem	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(このプールでは、大要求は512バイトから4Kバイトまで)
alloc	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
fail	失敗した大メモリー要求数
ovsz_alloc	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
fail	失敗したサイズ超過メモリー要求数

▼ 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ア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。

```
msg/s      1秒当たりのメッセージ処理(送受信)数
sema/s     1秒当たりのセマフォア処理数
```

例 13-10 プロセス間通信をチェックする (sar -m)

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 balmyday 5.10 s10_51 sun4u    03/18/2004

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
```

例 13-11 ページイン動作をチェックする (sar -p)

次の例は、sar -p コマンドからの出力を示します。

```
$ sar -p
SunOS balmyday 5.10 s10_51 sun4u      03/18/2004

00:00:04 atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00   0.09   0.00   0.00   0.78   2.02   0.00
02:00:01   0.08   0.00   0.00   0.78   2.02   0.00
03:00:00   0.09   0.00   0.00   0.81   2.07   0.00
04:00:00   0.11   0.01   0.01   0.86   2.18   0.00
05:00:00   0.08   0.00   0.00   0.78   2.02   0.00
06:00:00   0.09   0.00   0.00   0.78   2.02   0.00
07:00:00   0.08   0.00   0.00   0.78   2.02   0.00
08:00:00   0.09   0.00   0.00   0.78   2.02   0.00
08:20:00   0.11   0.00   0.00   0.87   2.24   0.00
08:40:01   0.13   0.00   0.00   0.90   2.29   0.00
09:00:00   0.11   0.00   0.00   0.88   2.24   0.00
09:20:00   0.10   0.00   0.00   0.88   2.24   0.00
09:40:00   2.91   1.80   2.38   4.61  17.62   0.00
10:00:00   2.74   2.03   3.08   8.17  21.76   0.00
10:20:00   0.16   0.04   0.04   1.92   2.96   0.00
10:40:00   2.10   2.50   3.42   6.62  16.51   0.00
11:00:00   3.36   0.87   1.35   3.92  15.12   0.00

Average   0.42   0.22   0.31   1.45   4.00   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。

フィールド名	説明
atrch/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秒当たりのページフォルト数(1秒当たりの付加数)。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ます。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です。
pgin/s	ファイルシステムがページイン要求を受信する1秒当たりの回数。
ppgin/s	ページインされる1秒当たりのページ数。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ります。
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」などがあります。通常、この数値は主に「書き込み時コピー」からなっています。
vflt/s	1秒当たりのアドレス変換ページフォルト数。このフォルトは、有効性フォルトと呼ばれます。有効性フォルトは、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します。
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秒当たりのフォルト数。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ります。システムはデータを受信するページをロックするの、別のプロセスはそれを回収して使用できません。

▼ 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、以下の情報を報告できます。
 - 待ち行列に要求が入っている間の平均待ち行列の長さ
 - 待ち行列に要求が入っている時間の割合

```
$ 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。

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	ディスパッチ待ち行列が使用されている時間の割合

swpq-sz スワップアウトされるプロセスの平均数。
 %swpocc プロセスがスワップアウトされている時間の割合。

例 13-12 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` の値が 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 system2 5.10 Generic_142909-13 sun4u    06/28/2010

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00    1.0      7     0.0      0
02:00:00    1.0      7     0.0      0
03:00:00    1.0      7     0.0      0
04:00:00    1.0      7     0.0      0
05:00:00    1.0      6     0.0      0
06:00:00    1.0      7     0.0      0

Average      1.0      7     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
```

次の表では、`-r` オプションからの出力について説明します。

<code>freemem</code>	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
<code>freeswap</code>	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 13-13 未使用のメモリーをチェックする (sar -r)

次の例は、`sar -r` コマンドからの出力を示します。

```
$ sar -r
SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 freemem freeswap
```

```

01:00:00  44717  1715062
02:00:01  44733  1715496
03:00:00  44715  1714746
04:00:00  44751  1715403
05:00:00  44784  1714743
06:00:00  44794  1715186
07:00:00  44793  1715159
08:00:00  44786  1714914
08:20:00  44805  1715576
08:40:01  44797  1715347
09:00:00  44761  1713948
09:20:00  44802  1715478
09:40:00  41770  1682239
10:00:00  35401  1610833
10:20:00  34295  1599141
10:40:00  33943  1598425
11:00:00  30500  1561959

Average   43312  1699242

```

▼ 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** コマンドと同じです。プロセッサの状態には、「ビジー」と「アイドル」があります。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。

次の表では、**-u** オプションからの出力について説明します。

%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ます。
%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ます。
%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ます。
%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ます。

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 13-14 CPU の使用状況をチェックする (sar -u)

次の例は、sar -u コマンドからの出力を示します。

```
$ sar -u

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04    %usr    %sys    %wio    %idle
01:00:00         0         0         0     100
02:00:01         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
08:20:00         0         0         0      99
08:40:01         0         0         0      99
09:00:00         0         0         0      99
09:20:00         0         0         0      99
09:40:00         4         1         0      95
10:00:00         4         2         0      94
10:20:00         1         1         0      98
10:40:00        18         3         0      79
11:00:00        25         3         0      72

Average         2         0         0      98
```

▼ システムテーブルのステータス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ステータスが表示されます。

```
$ 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
```

例 13-15 システムテーブルのステータスをチェックする (sar -v)

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 balmyday 5.10 s10_51 sun4u    03/18/2004
```

```

00:00:04 proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00 69/8010 0 3476/34703 0 0/0 0 0/0
02:00:01 69/8010 0 3476/34703 0 0/0 0 0/0
03:00:00 69/8010 0 3476/34703 0 0/0 0 0/0
04:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
05:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
06:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
07:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:20:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:40:01 69/8010 0 3494/34703 0 0/0 0 0/0
09:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
09:20:00 69/8010 0 3494/34703 0 0/0 0 0/0
09:40:00 74/8010 0 3494/34703 0 0/0 0 0/0
10:00:00 75/8010 0 4918/34703 0 0/0 0 0/0
10:20:00 72/8010 0 4918/34703 0 0/0 0 0/0
10:40:00 71/8010 0 5018/34703 0 0/0 0 0/0
11:00:00 77/8010 0 5018/34703 0 0/0 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。

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ありません。ここからオーバーフローすることもあります。
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 として表示されます。
ov	各テーブルのサンプルポイント間で発生しているオーバーフロー。
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 として表示されます。

▼ 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

```

次に、`sar -w` コマンド出力の対象となる値と説明を示します。

swpin/s	メモリーに転送される 1 秒当たりの軽量プロセス数
bswin/s	スワップイン用に転送される 1 秒当たりのブロック数。 /* (float)PGTOBLK(xx->cvmi.pgswpin) / sec_diff */.
swpot/s	メモリーからスワップアウトされる 1 秒当た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 秒当たりのブロック数
pswch/s	1 秒当た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 13-16 スワップ動作をチェックする (sar -w)

次の例は、sar -w コマンドからの出力を示します。

```
$ sar -w
SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00  0.00    0.0    0.00    0.0    132
02:00:01  0.00    0.0    0.00    0.0    133
03:00:00  0.00    0.0    0.00    0.0    133
04:00:00  0.00    0.0    0.00    0.0    134
05:00:00  0.00    0.0    0.00    0.0    133
06:00:00  0.00    0.0    0.00    0.0    133
07:00:00  0.00    0.0    0.00    0.0    132
08:00:00  0.00    0.0    0.00    0.0    131
08:20:00  0.00    0.0    0.00    0.0    133
08:40:01  0.00    0.0    0.00    0.0    132
09:00:00  0.00    0.0    0.00    0.0    132
09:20:00  0.00    0.0    0.00    0.0    132
09:40:00  0.00    0.0    0.00    0.0    335
10:00:00  0.00    0.0    0.00    0.0    601
10:20:00  0.00    0.0    0.00    0.0    353
10:40:00  0.00    0.0    0.00    0.0    747
11:00:00  0.00    0.0    0.00    0.0    804

Average  0.00    0.0    0.00    0.0    198
```

▼ 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 mdmin/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回線がないかどうかを判別できます。次に、記録される動作を示します。

rawch/s 1秒当たりの入力文字数 (raw 待ち行列)。
 canch/s 標準待ち行列で処理される1秒当たりの入力文字数。
 outch/s 1秒当たりの出力文字数 (出力待ち行列)。
 rcvin/s 1秒当たりの受信側ハードウェア割り込み数。
 xmtin/s 1秒当たりの送信側ハードウェア割り込み数。
 mdmin/s 1秒当たりのモデム割り込み数。

1秒当た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秒当た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 13-17 端末動作をチェックする (sar -y)

次の例は、sar -y コマンドからの出力を示します。

```
$ sar -y
SunOS balmyday 5.10 s10_51 sun4u 03/18/2004

00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0
02:00:01      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
08:20:00      0      0      0      0      0      0
08:40:01      0      0      0      0      0      0
09:00:00      0      0      0      0      0      0
09:20:00      0      0      0      0      0      0
09:40:00      0      0      1      0      0      0
10:00:00      0      0     37      0      0      0
10:20:00      0      0      0      0      0      0
10:40:00      0      0      3      0      0      0
11:00:00      0      0      3      0      0      0

Average      0      0      1      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、システム全体のパフォーマンス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、3つのコマンドを使用します。sadc、sa1、およびsa2です。

sadc データ収集ユーティリティーは、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る) 間隔でサンプルデータを *n* 回収集します。このコマンドは次に、2進形式の *ofile* ファイルまたは標準出力に書き込みます。

ブート時の sadc コマンドの実行

カウンタが0にリセットされたとき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、`svcadm enable system/sar:default` コマンドで日次データファイルにレコードを書き込みます。

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た **sadc** コマンドの定期的な実行

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/crontabs/sys` ファイルの次の行をコメント解除するのがもっとも簡単な方法です。

```
# 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、月曜から金曜までの午前8時から午後5時までは20分ごとに、それ以外では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/saradd` 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たレポートの生成

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ない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次動作ファイルに収集された情報を報告できます。

`sar` コマンドの書式は次のとおりです。

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvwy] [-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` コマンドは以前に記録されたファイルからデータを抽出します。このファイルは、`-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日次動作ファイル `/var/adm/sa/sadd` です。
- `-s` オプションと `-e` オプションでは、レポートの開始時間と終了時間を定義します。開始時間と終了時間の書式は `hh[:mm[:ss]]` です (この場合、`hh`、`mm`、`ss` 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、日次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 13-2 `sar` コマンドのオプション

オプション	アクション
<code>-a</code>	ファイルアクセス操作をチェックする
<code>-b</code>	バッファ動作をチェックする
<code>-c</code>	システムコールをチェックする
<code>-d</code>	各ブロックデバイスの動作をチェックする
<code>-g</code>	ページアウトとメモリーの解放をチェックする
<code>-k</code>	カーネルメモリーの割り当てをチェックする
<code>-m</code>	プロセス間通信をチェックする
<code>-nv</code>	システムテーブルのステータスをチェックする
<code>-p</code>	スワップとディスパッチ動作をチェックする
<code>-q</code>	待ち行列動作をチェックする
<code>-r</code>	未使用メモリーをチェックする
<code>-u</code>	CPU の使用率をチェックする
<code>-w</code>	ボリュームのスワッピングと切り替えをチェックする

表 13-2 sar コマンドのオプション (続き)

オプション	アクション
-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、sar コマンドを `-u` オプションを指定して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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 `svcadm enable system/sar:default` コマンドを実行します。
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` を編集します。

注 - `crontab` ファイルは直接編集しないでください。既存の `crontab` ファイルを変更するときは、代わりに `crontab -e` コマンドを使用してください。

```
# crontab -e sys
```

- 4 次の行のコメントを解除します。

```
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
```

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

ソフトウェアのトラブルシューティング (概要)

この章では、ソフトウェアのトラブルシューティングについての概要を説明します。システムクラッシュのトラブルシューティング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。

- 221 ページの「トラブルシューティングの新機能」
- 223 ページの「ソフトウェアのトラブルシューティングタスクの参照先」
- 224 ページの「システムクラッシュの問題のトラブルシューティング」
- 225 ページの「システムクラッシュをトラブルシュートするためのチェックリスト」

トラブルシューティングの新機能

このセクションでは、今回のリリースで新たに追加または変更されたトラブルシューティング情報について説明します。

Oracle Solaris 10 リリースのトラブルシューティングの新機能と変更された機能については、以下を参照してください。

- 222 ページの「動的トレース機能」
- 223 ページの「kadb に代わる標準 Solaris カーネルデバッグ kmdb」

Oracle Solaris の新機能の完全な一覧や各 Oracle Solaris リリースの説明については、『Oracle Solaris 10 8/11 の新機能』を参照してください。

共通エージェントコンテナの問題

Solaris 10 6/06: 共通エージェントコンテナは、Oracle Solaris OS に含まれているスタンドアロンの Java プログラムです。このプログラムは、Java 管理アプリケーション用のコンテナを実装します。共通エージェントコンテナは、JMX (Java Management

Extensions) および Java DMK (Java Dynamic Management Kit) に基づく機能のために設計された管理インフラストラクチャーを提供します。このソフトウェアは SUNWcacaort パッケージによってインストールされ、`/usr/lib/cacao` ディレクトリに置かれます。

通常、このコンテナは表示されません。ただし、次の2つの場合は、コンテナデーモンとの対話が必要になることがあります。

- 共通エージェントコンテナ用に予約されているネットワークポートを別のアプリケーションが使用しようとする可能性がある。
- 証明書ストアが危殆化した場合に、共通エージェントコンテナ証明書鍵の再生成が必要になることがある。

これらのトラブルシューティング方法については、268 ページの「Oracle Solaris OS で の共通エージェントコンテナのトラブルシューティング」を参照してください。

x86: システムリブート時に SMF ブートアーカイブ サービスが失敗する場合がある

Solaris 10 1/06: GRUB ベースのブート環境でシステムクラッシュが発生した場合、システムのリブート時に SMF サービス (`svc:/system/boot-archive:default`) が失敗する場合があります。この問題が発生した場合は、システムをリブートし、GRUB ブートメニューでフェイルセーフアーカイブを選択します。プロンプトの指示に従い、ブートアーカイブを再構築します。アーカイブの再構築後、システムをリブートします。ブートプロセスを継続するには、`svcadm` コマンドを使用して `svc:/system/boot-archive:default` サービスをクリアします。GRUB ベースのブートの詳細については、『Solaris のシステム管理 (基本編)』の「GRUB を使用して x86 システムをブートする (作業マップ)」を参照してください。

動的トレース機能

Oracle Solaris 動的トレース (DTrace) 機能は、Solaris のカーネルプロセスやユーザープロセスの監視に新たな可能性を実現する、包括的な動的追跡機能です。DTrace では、OS のカーネルプロセスとユーザープロセスを動的に計測し、プローブと呼ばれる関心のある場所で指定のデータを記録して、システム理解に役立てることができます。それぞれのプローブは、新しい D プログラム言語で記述されたカスタムプログラムに関連付けられます。DTrace のすべての計測は完全に動的であり、実稼働システムで使用可能です。詳細は、`dttrace(1M)` のマニュアルページと『Solaris 動的トレースガイド』を参照してください。

kadb に代わる標準 Solaris カーネルデバugga kmdb

kmdb が kadb に代わって、標準的な「既定の」Solaris カーネルデバuggaとして使用されます。

kmdb は、mdb の機能と柔軟性を集約し、カーネルのライブデバuggaを行います。kmdb は以下をサポートします。

- デバuggaコマンド (dcmds)
- デバuggaモジュール (dmods)
- カーネルタイプのデータへのアクセス
- カーネル実行制御
- 検査
- 修正

詳細は、[kmdb\(1\)](#) のマニュアルページを参照してください。kmdb を使用してシステムをトラブルシューティングする手順については、『Solaris のシステム管理 (基本編)』の「カーネルデバugga (kmdb) を使ってシステムをブートする方法」および『Solaris のシステム管理 (基本編)』の「GRUB ブート環境でカーネルデバugga (kmdb) を使ってシステムをブートする方法」を参照してください。

ソフトウェアのトラブルシューティングタスクの参照先

トラブルシューティングのタスク	詳細
システムクラッシュ情報の管理	第 17 章「システムクラッシュ情報の管理(タスク)」
コアファイルの管理	第 16 章「コアファイルの管理(タスク)」
リブート失敗、バックアップ問題などのソフトウェアのトラブルシューティング	第 18 章「ソフトウェアで発生するさまざまなトラブルシューティング(タスク)」
ファイルアクセスのトラブルシューティング	第 19 章「ファイルアクセスでのトラブルシューティング(タスク)」
印刷のトラブルシューティング	『Solaris のシステム管理ガイド (印刷)』の第 13 章「Oracle Solaris OS での印刷に関する問題のトラブルシューティング(タスク)」
UFS ファイルシステムの不整合の解決	第 20 章「UFS ファイルシステムの不整合解決(タスク)」
ソフトウェアパッケージのトラブルシューティング	第 21 章「ソフトウェアパッケージで発生するトラブルシューティング(タスク)」

システムクラッシュの問題のトラブルシューティング

Oracle Solaris OS 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

もっとも重要なことは、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、システムを再稼働させるのを最優先に考えがちです。しかし、システムをリポートする前に、コンソール画面のメッセージを確認してください。これらのメッセージは、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ポートして、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については、[228 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、その原因を特定できない場合は、システムコンソールまたは `/var/adm/messages` ファイルからできるだけ多くの情報を収集し、購入先に問い合わせます。購入先に問い合わせるために収集しておくトラブルシューティング情報の完全なリストについては、[224 ページの「システムクラッシュの問題のトラブルシューティング」](#)を参照してください。

システムクラッシュ後に、リポートが失敗する場合は、[第 18 章「ソフトウェアで発生するさまざまなトラブルシューティング\(タスク\)」](#)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
```

システムクラッシュ後に、リポートが失敗する場合は、[第 18 章「ソフトウェアで発生するさまざまなトラブルシューティング\(タスク\)」](#)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せん。クラッシュダンプについては、[第 17 章「システムクラッシュ情報の管理\(タスク\)」](#)を参照してください。

トラブルシューティングデータの収集

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。[225 ページの「システムクラッシュをトラブルシュートするためのチェックリスト」](#)を使用し、クラッシュしたシステムのトラブルシューティングデータを収集します。

表 14-1 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
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)を行ったり、通常よりも高いロード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トラブルシューティング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	

システムクラッシュをトラブルシュートするためのチェックリスト

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	

◆◆◆ 15

第 15 章

システムメッセージの管理

この章では、Oracle Solaris OS のシステムメッセージ機能について説明します。

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*]

例:

```
[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。例:

```
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
```

まれに、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、さまざまなシステムの警告やエラーをメッセージファイルに自動的に記録します。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。詳しくは、230 ページの「システムのメッセージ記録のカスタマイズ」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。もっとも新しいメッセージは、`/var/adm/messages` (および `messages.*`) であり、もっと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む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、このタスクは自動化できます。このタスクの自動化については、93 ページの「クラッシュダンプファイルを削除する方法」および第 8 章「システムタスクのスケジュール設定(タスク)」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ポ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
```

例 15-1 システムメッセージの表示

次の例は、`dmesg` コマンドからの出力を示します。

```
$ dmesg
Jan  3 08:44:41 starbug genunix: [ID 540533 kern.notice] SunOS Release 5.10 ...
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
.
.
.
```

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

システムログローテーション

システムログファイルは、`root`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ローテーションが毎週実行されることを示しています。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、コピーが保存されます。

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
```

詳細は、[logadm\(1M\)](#)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` ファイルに追加すれば、`logadm` コマンドを使用する権限がユーザー `andy` に与えられます。

```
andy::::profiles=Log Management
```

または、Solaris Management Console 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。

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、さまざまなシステムプロセスが生成するさらに多くの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、[228 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 15-1](#) を参照。level は、記録する状態の重要度や優先順位を示します。優先レベルについては [表 15-2](#) を参照

同じ機能の2つのエントリは、それぞれの優先順位が異なる場合、同じ行に入力しないでください。syslog ファイルに優先順位を入力すると、この優先順位以上のすべてのメッセージが記録され、最後のメッセージが優先されます。指定の機能とレベルに対し、syslogd はそのレベル以上のすべてのメッセージを記録します。

action 動作フィールドは、メッセージが転送される場所を示します。

次の例は、デフォルトの `/etc/syslog.conf` ファイルのサンプルを示します。

```

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。

注- エントリを個別の行に入力すると、`/etc/syslog.conf` ファイルでログの対象が複数回指定された場合に、メッセージのログ順が変わることがあります。単独行のエントリに複数のセクターを指定できます。その際、セクターはセミコロンで区切ります。

一般的なエラー状態の送信元を次の表に示します。一般的な優先順位を、重要度順に表 15-2 に示します。

表 15-1 `syslog.conf` メッセージの送信元の機能

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 15-2 `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

表 15-2 syslog.conf メッセージの優先レベル (続き)

優先順位	説明
crit	クリティカルなエラー
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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 `/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 15-2 システムのメッセージ記録のカスタマイズ

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トラブルシューティングしやすくなります。

- `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` コマンドを使用すると、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\(IM\)](#) 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、[235](#) 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、新しい仮想デバイス `/dev/sysmsg` に書き込まれる、カーネルメッセージと `syslog` メッセージからなります。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。メッセージ出力先を補助デバイスに変更したい場合は、`/dev/console` を参照しているプログラムで `syslog()` または `strlog()` を使用するように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` コマンドや `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、システムリブート後も同じ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た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、[234 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。

補助コンソールの有効化については、[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 15-3 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 15-4 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
```

or
 - 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 15-5 補助(リモート)コンソールを無効にする

```
# consadm -d /dev/term/a
# consadm
```


◆◆◆ 第 16 章

コアファイルの管理 (タスク)

この章では、`coreadm` コマンドを使って、コア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手順については、239 ページの「コアファイルの管理 (タスクマップ)」を参照してください。

コアファイルの管理 (タスクマップ)

タスク	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	242 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 プロセス別コアファイルのパスを有効にする グローバルのコアファイルのパスを有効にする	243 ページの「コアファイル名パターンを設定する方法」 243 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」 243 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	244 ページの「コアファイルの調査」

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場合に生成されます。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。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コア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、コアファイルがデフォルトで現在のディレクトリに作成されます。グローバルコアファイルのパスが有効になっていると、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つはグローバルコアファイルの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コアファイルを相互に区別できます。

変数名	変数の定義
%d	実行ファイルのディレクトリ名。最大文字数は MAXPATHLEN
%f	実行ファイルの名前。最大文字数は MAXCOMLEN
%g	実効グループ ID
%m	マシン名 (uname -m)
%n	システムノード名 (uname -n)
%p	プロセス ID
%t	time(2) の 10 進数
%u	実効ユーザー ID
%z	プロセスが実行されているゾーン名 (zonename)
%%	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、次のコア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、ゾーン別、またはプロセス別に設定できます。さらに、システムリブート後も有効なプロセス別デフォルトを設定できます。

たとえば、次の `coreadm` コマンドでは、デフォルトのプロセス別コアファイルパターンを設定します。この設定は、デフォルトのコアファイルパターンを明示的に上書きしていないプロセスに対して適用されます。この設定はシステムリブート後も有効です。

```
# coreadm -i /var/core/core.%f.%p
```

次の `coreadm` コマンドでは、任意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 `$HOME/.profile` または `$HOME/.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、次のパス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` プログラムがコア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` プロセスがコア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を無効にすると、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。

詳細は、[coreadm\(1M\)](#) 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:
global core file content: default
      init core file pattern: core
      init core file content: default
      global core dumps: disabled
```

```
per-process core dumps: enabled
global setid core dumps: disabled
per-process setid core dumps: disabled
global core dump logging: disabled
```

▼ コアファイル名パターンを設定する方法

- 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. スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
 - c. グローバルコアファイル名パターンを設定します。

```
# 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るか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 プロセス別コアファイルのパスを有効にする

```
# coreadm -e process
```
- 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るか、同等の役割を引き受けます。

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

- 2 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 -e global -g /var/core/core.%f.%p

# coreadm
  global core file pattern: /var/core/core.%f.%p
  global core file content: default
  init core file pattern: core
  init core file content: default
  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
```

コアファイルのトラブルシューティング

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` を削除します。次に `coreadm` コマンドを使って、グローバル setuid コアファイルパスを有効にします。

コアファイルの調査

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 16-1 `proc` ツールによるコアファイルの調査

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
```

例 16-1 proc ツールによるコアファイルの調査 (続き)

```
core './core' of 19305: ./a.out
000108c4 main      (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
00010880 _start    (0, 0, 0, 0, 0, 0) + b8
```


システムクラッシュ情報の管理 (タスク)

この章では、Oracle Solaris OS 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、248 ページの「システムクラッシュ情報の管理 (タスクマップ)」を参照してください。

システムクラッシュ情報の管理に関する新機能

このセクションでは、Oracle Solaris でシステムリソースを管理するための新機能、または機能の変更について説明します。

高速クラッシュダンプ機能

Oracle Solaris 10 9/10: この機能強化により、システムではより短時間に、少ない容量でクラッシュダンプを保存できるようになりました。クラッシュダンプが完了するまでの所要時間は、プラットフォームに応じて2倍から10倍高速化されています。クラッシュダンプを `savecore` ディレクトリ内に保存するのに必要なディスク容量は、同じ比率で減少しています。クラッシュダンプファイルの作成と圧縮を高速化するため、高速クラッシュダンプ機能は、大規模システムの使用頻度が低い CPU を利用します。新しいクラッシュダンプファイルの `vmdump.n` は、`vmcore.n` ファイルと `unix.n` ファイルの圧縮されたバージョンです。圧縮されたクラッシュダンプは、より迅速にネットワーク上を移動し、オフサイトで分析することができます。ダンプファイルを `mdb` ユーティリティなどのツールで使うためには、最初に圧縮解除する必要があることに注意してください。ダンプファイルはローカルで、またはリモートから `savecore` コマンドを使用して、圧縮解除することができます。

新しいクラッシュダンプ機能をサポートするため、`dumpadm` コマンドに `-z` オプションが追加されました。このオプションを使用して、ダンプを圧縮または非圧縮のどちらの形式で保存するかを指定します。デフォルトは圧縮した形式です。

詳細については、[dumpadm\(1M\)](#) および [savecore\(1M\)](#) のマニュアルページを参照してください。

システムクラッシュ情報の管理(タスクマップ)

次のタスクマップは、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。

タスク	説明	参照先
1. 現在のクラッシュダンプ構成を表示する	<code>dumpadm</code> 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	252 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	<code>dumpadm</code> 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い容量を指定する	253 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調べる	<code>mdb</code> 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	255 ページの「クラッシュダンプを検査する方法」
4.(オプション)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、 <code>savecore</code> ディレクトリに十分な空き容量がなくても、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	256 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(オプション)」
5.(オプション)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	256 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。

クラッシュダンプ情報は圧縮した形式で `vmdump.n` ファイルに書き込まれます。この `n` は、クラッシュダンプ識別用の整数です。その後、同じシステムまたは別のシステムで `savecore` コマンドを呼び出して、圧縮されているクラッシュダンプを、`unix.n` および `vmcore.n` という名前の 1 組のファイルに展開できます。リブート時にクラッシュダンプが保存されるディレクトリも、`dumpadm` コマンドを使用して構成できます。

UFS ルートファイルシステムがあるシステムの場合、デフォルトのダンプデバイスは適切なスワップパーティションとして構成されます。スワップパーティションは、オペレーティングシステム用の仮想メモリーのバックアップストレージとして予約されているディスクパーティションです。このため、クラッシュダンプによって上書きされることになるスワップ内に、永続的な情報は置かれません。Oracle Solaris ZFS ルートファイルシステムがあるシステムの場合、スワップとダンプの領域用には専用の ZFS ボリュームが使用されます。詳細については、[249 ページの「スワップ領域およびダンプデバイスに関する Oracle Solaris ZFS サポート」](#) を参照してください。

スワップ領域およびダンプデバイスに関する Oracle Solaris ZFS サポート

Oracle Solaris ZFS ルートファイルシステムをインストールするか、Oracle Solaris Live Upgrade プログラムを使用して UFS ルートファイルシステムから ZFS ルートファイルシステムに移行する場合、スワップデバイスおよびダンプデバイスは 2 つの ZFS ボリューム上に作成されます。たとえば、デフォルトのルートプール名 `rpool` では、`/rpool/swap` ボリュームと `/rpool/dump` ボリュームが自動的に作成されます。スワップボリュームとダンプボリュームのサイズを新しいサイズに調整できます。ただし、システムの動作をサポートするサイズを選択する必要があります。詳細は、『[Oracle Solaris ZFS 管理ガイド](#)』の「[スワップデバイスおよびダンプデバイスの ZFS サポート](#)」を参照してください。

インストール後に ZFS スワップデバイスやダンプデバイスを変更する必要がある場合は、以前のリリース同様、`swap` または `dumpadm` コマンドを使用します。

ダンプデバイスの管理について、このドキュメントでは、[252 ページの「システムクラッシュダンプ情報の管理」](#) を参照してください。

x86: GRUB ブート環境のシステムクラッシュ

GRUB ブート環境の x86 ベースのシステムでシステムクラッシュが発生した場合、GRUB ブートアーカイブ (`svc:/system/boot-archive:default`) を管理する SMF サービスが、次のシステムリブート時に失敗する可能性があります。GRUB ベースのブートの詳細については、『[Solaris のシステム管理 \(基本編\)](#)』の「[GRUB を使用して x86 システムをブートする \(作業マップ\)](#)」を参照してください。

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。以前のリリースでは、システムを手動で有効にして物理メモリ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、システムがリブ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。詳しくは、[251 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の使用法については、[mdb\(1\)](#) のマニュアルページを参照してください。

さらに、`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

Oracle Solaris OS でシステムクラッシュダンプ情報を管理するには、`dumpadm` コマンドを使用します。

- 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、システムをブートする際、`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 Oracle Solaris OS 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トラブルシュート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` 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、 <code>savecore</code> 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
<code>savecore</code> 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 <code>savecore</code> ディレクトリに必要な最小空き容量。空き容量を構成しないと、デフォルトで 1M バイトになる

詳細は、`dumpadm(1M)` のマニュアルページを参照してください。

ダンプ構成パラメータは、`dumpadm` コマンドで管理します。

dumpadm コマンドの動作

`dumpadm` コマンドは、システム起動時に `svc:/system/dumpadm:default` サービスによって呼び出されて、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、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ー

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ー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ー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か、同等の役割になる必要があります。
- 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。
- 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に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るか、同等の役割を引き受けます。

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

- 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
Saved compressed: on
```

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
- クラッシュダンプを圧縮した形式で保存する

▼ クラッシュダンプ構成を変更する方法

- スーパーユーザーになるか、同等の役割を引き受けます。

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

- 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
Save compressed: on
```

この出力は、Oracle Solaris 10 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表しています。

- クラッシュダンプ構成を変更します。

```
# /usr/sbin/dumpadm [-nuy] [-c content-type] [-d dump-device] [-m mink | minm | min%]
[-s savecore-dir] [-r root-dir] [-z on | off]
```

-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プライマリスワップデバイス
- `-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、[256 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\(オプション\)」](#)を参照してください。
- `-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` が有効でないと、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
- `-u` `/etc/dumpadm.conf` ファイルの内容に基づいてカーネルダンプ構成を強制的に更新します。
- `-y` リブート時に自動的に `savecore` コマンドを実行するようにダンプ構成を変更します。このダンプ設定では、このコマンドの自動実行がデフォルトです。
- `-z on | off` リブート時の `savecore` コマンドの動作を制御するために、ダンプ構成を変更します。`on` 設定では、圧縮した形式でのコアファイルの保存が有効になります。`off` 設定では、クラッシュダンプファイルを自動的に圧縮解除します。クラッシュダンプファイルはサイズが非常に大きくなる場合があり、圧縮した形式で保存すれば必要なシステム領域が小さくなるため、デフォルトは `on` です。

例 17-1 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
  Save compressed: on
# 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
  Save compressed: on
```

▼ 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るか、同等の役割を引き受けます。

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

- 2 クラッシュダンプを検査するには、`mdb`ユーティリティーを使用します。

```
# /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 クラッシュステータス情報を表示します。

```
# /usr/bin/mdb file-name
```

```
> ::status
```

```
.
```

```
.
```

```
.
```

```
> ::system
```

```
.
```

```
.
```

```
.
```

例 17-2 クラッシュダンプを検査する

次の例は、mdb ユーティリティーの出力例を示します。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ている調整可能パラメータが含まれています。

```
# /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: 5.10 Generic (sun4u)
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsiz=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(オプション)

ここでは、システムがクラッシュしたが、十分な空き容量が `savecore` ディレクトリに残っておらず、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ら、スーパーユーザーとしてログインするか、同等の役割になります。
- 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。
 - 別の方法として、`savecore` コマンドを手作業で実行し、十分なディスク容量がある代替ディレクトリを指定することができます。

```
# 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るか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 2 システム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 17-3 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
  Save Compressed: on
```

例 17-4 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
  Save compressed: on
```


ソフトウェアで発生するさまざまなトラブルシューティング(タスク)

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。ソフトウェアで発生するさまざまなトラブルシューティングは、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、次のセクションで説明します。

この章で説明する情報は次のとおりです。

- 259 ページの「リブートが失敗した場合の対処」
- 263 ページの「x86: システムリブート中に SMF ブートアーカイブサービスが失敗した場合の対処」
- 264 ページの「システムがハングした場合の対処」
- 265 ページの「ファイルシステムが一杯になった場合の対処」
- 266 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」
- 267 ページの「バックアップ時のトラブルシューティング」
- 268 ページの「Oracle Solaris OS での共通エージェントコンテナのトラブルシューティング」

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> を見つけれられない	SPARC システムの PROM 内の <code>boot-device</code> 設定を変更します。デフォルトのブートデバイスの変更については、『Solaris のシステム管理 (基本編)』の「ブート PROM を使用してデフォルトのブートデバイスを変更する方法」を参照してください。
<p>Oracle Solaris 10: 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。次のメッセージが表示される。</p> <p>Not a UFS filesystem.</p> <p>Solaris 10 1/06: GRUB ブートアーカイブが壊れている。または、SMF ブートアーカイブサービスが失敗した。svcs -x コマンドを実行すると、エラーメッセージが表示される</p>	<p>Oracle Solaris 10: Configuration Assistant (構成用補助) フロッピーディスク (ブートフロッピーとも呼ぶ) 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します。</p> <p>Solaris 10 1/06: フェイルセーフアーカイブをブートします。</p>
<code>/etc/passwd</code> ファイル内に無効なエントリが存在する	無効な <code>passwd</code> ファイルの復元については、『Solaris のシステム管理 (基本編)』の第 12 章「Oracle Solaris 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	<p>ハードウェアの接続を確認します。</p> <ul style="list-style-type: none"> ■ 装置が接続されていることを確認します。 ■ すべてのスイッチが適切に設定されていることを確認します。 ■ すべてのコネクタおよびケーブル (Ethernet ケーブルも含む) を検査します。 ■ すべて異常がなければ、システムの電源を切り、10 秒 - 20 秒程待って、もう一度電源を投入します。

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。

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。
- **Oracle Solaris 10:** ブートサーバーやインストールサーバー、またはローカル CD-ROM からシステムをブートする。
- 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。

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。
- **Solaris 10 1/06** リリース以降: x86 ベースのシステムでは、Solaris フェイルセーフアーカイブでシステムをブートする。
- **Oracle Solaris 10:** ブートサーバーやインストールサーバー、またはローカル CD-ROM からシステムをブートする。
- 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。

この手順については、『[Solaris のシステム管理 \(基本編\)](#)』の第 12 章「[Oracle Solaris システムのブート \(手順\)](#)」に詳述されています。

注-GRUB ベースのブートは、今回のリリースでは、SPARC ベースのシステムには使用できません。

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。

例 18-1 SPARC: ルートパスワードを忘れた場合の対処

次の例は、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--Press Stop A keys to stop the system)
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

例 18-2 x86: ルートパスワードを忘れた場合に GRUB ベースのブートを実行する

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。

```
GNU GRUB version 0.95 (637K lower / 3144640K upper memory)
+-----+
| be1
| be1 failsafe
| be3
| be3 failsafe
| be2
| be2 failfafa
+-----+
    Use the ^ and v keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, or 'c' for a command-line.

Searching for installed OS instances...

    An out of sync boot archive was detected on /dev/dsk/c0t0d0s0.
    The boot archive is a cache of files used during boot and
    should be kept in sync to ensure proper system operation.

    Do you wish to automatically update this boot archive? [y,n,?] n
Searching for installed OS instances...

Multiple OS instances were found. To check and mount one of them
read-write under /a, select it from the following list. To not mount
any, select 'q'.

    1 pool10:13292304648356142148      ROOT/be10
    2 rpool:14465159259155950256      ROOT/be01

Please select a device to be mounted (q for none) [?,??,q]: 1
mounting /dev/dsk/c0t0d0s0 on /a
starting shell.
.
.
.
# cd /a/etc
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# reboot
```

例 18-3 x86: ルートパスワードを忘れた場合にシステムをブートする

Oracle Solaris 10: 次の例は、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。

```
Press any key to reboot.
Resetting...
```

例 18-3 x86: ルートパスワードを忘れた場合にシステムをブートする (続き)

```

.
.
Initializing system
Please wait...

          <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                  to boot with defaults

          <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Use is subject to license terms.
DEBUG enabled
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is example.com
Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
Entering System Maintenance Mode
.
.
.
# mount /dev/dsk/c0t0d0s0 /a
.
.
# cd /a/etc
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6

```

x86: システムリブート中に SMF ブートアーカイブサービスが失敗した場合の対処

Solaris 10 1/06: このリリースでは、システムがクラッシュした場合、システムのリブート時に、ブートアーカイブ SMF サービスである `svc:/system/boot-archive:default` が失敗する場合があります。ブートアーカイブサービスが失敗した場合、`svcs -x` コマンドを実行すると、次のようなメッセージが表示されます。

```
svc:/system/boot-archive:default (check boot archive content)
State: maintenance since Fri Jun 03 10:24:52 2005
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
See: http://sun.com/msg/SMF-8000-K5
See: /etc/svc/volatile/system-boot-archive:default.log
Impact: 48 dependent services are not running. (Use -v for list.)

svc:/network/rpc/gss:default (Generic Security Service)
State: uninitialized since Fri Jun 03 10:24:51 2005
Reason: Restarter svc:/network/inetd:default is not running.
See: http://sun.com/msg/SMF-8000-5H
See: gssd(1M)
Impact: 10 dependent services are not running. (Use -v for list.)

svc:/application/print/server:default (LP print server)
State: disabled since Fri Jun 03 10:24:51 2005
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
See: lpsched(1M)
Impact: 1 dependent service is not running. (Use -v for list.)
```

問題を修正するには、次の処置を行います。

1. システムをリブートし、GRUB ブートメニューからフェイルセーフアーカイブのオプションを選択します。
2. システムが、ブートアーカイブを再構築するかを尋ねるプロンプトを表示したら、y キーを押します。
ブートアーカイブの再構築後、システムはブートの準備が完了します。
3. ブートを継続するには、次のコマンドを使用して SMF ブートアーカイブサービスをクリアします。

```
# svcadm clear boot-archive
```

このコマンドを実行するには、スーパーユーザーまたは同等の役割になっている必要があります。

GRUB ブートアーカイブの再構築の詳細については、『Solaris のシステム管理 (基本編)』の「x86 システムをフェイルセーフモードでブートする方法」および [bootadm\(1M\)](#) のマニュアルページを参照してください。

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-h 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。コア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、スーパーユーザーまたは同等の役割になり、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「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。次のセクションでは、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 6 章「ディスク使用の管理 (タスク)」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、大きなコア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するか、同等の役割になり、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します。コアファイルの削除については、 92 ページの「コア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 <code>/tmp</code> ディレクトリにコピーしたり復元したりすると、ACL属性が消失する。 <code>/tmp</code> 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

バックアップ時のトラブルシューティング

このセクションでは、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トラブルシューティング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
-f 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をルート (/) ファイルシステムの <code>/dev</code> 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送信されず、 <code>/dev/rmt/st0</code> として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。ほかのコマンドで書き込まれたテープを `ufsrestore` コマンドで復元しようとした場合、テープが `ufsdump` 形式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: Fri Jan 30 10:13:46 2004
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 >
```

Oracle Solaris OS での共通エージェントコンテナのトラブルシューティング

このセクションでは、共通エージェントコンテナ共有コンポーネントで発生する可能性のある問題について説明します。この Oracle Solaris OS リリースでは、Oracle Solaris OS 内に共通エージェントコンテナ Java プログラムが含まれています。このプログラムは、Java 管理アプリケーション用のコンテナを実装します。通常、このコンテナはユーザーに表示されません。

発生する可能性のある問題は、次のとおりです。

- ポート番号の競合
- スーパーユーザーパスワードのセキュリティの危殆化

ポート番号の競合

共通エージェントコンテナは、デフォルトでは次のポート番号を占有します。

- JMX ポート (TCP) = 11162
- SNMP アダプタポート (UDP) = 11161
- トラップ用 SNMP アダプタポート (UDP) = 11162
- コマンドストリームアダプタポート (TCP) = 11163
- RMI コネクタポート (TCP) = 11164

注 - Oracle Solaris Cluster のインストールをトラブルシューティングする場合は、ポートの割り当てが異なります。

インストール時にこれらのポート番号のいずれかがすでに予約されている場合は、次の手順に従って共通エージェントコンテナの占有するポート番号を変更してください。

▼ ポート番号をチェックする方法

この手順では、ポートを確認する方法を示します。

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

- 2 共通エージェントコンテナの管理デーモンを停止します。

```
# /usr/sbin/cacaoadm stop
```

- 3 次の構文を使用してポート番号を変更します。

```
# /usr/sbin/cacaoadm set-param param=value
```

たとえば、SNMPAdaptor の占有するポートをデフォルトの 11161 から 11165 に変更するには、次のように入力します。

```
# /usr/sbin/cacaoadm set-param snmp-adaptor-port=11165
```

- 4 共通エージェントコンテナの管理デーモンを再起動します。

```
# /usr/sbin/cacaoadm start
```

スーパーユーザーパスワードのセキュリティの危殆化

Java ES が稼働しているホストで、セキュリティ鍵の再生成が必要になる場合があります。たとえば、スーパーユーザーパスワードが公開されたり危殆化した危険がある場合は、セキュリティ鍵を再生成するべきです。共通エージェントコンテナサービスで使用される鍵は、`/etc/cacao/instances/instance-name/security` ディレクトリに格納されます。次のタスクでは、Oracle Solaris OS のセキュリティ鍵を生成する方法を示します。

▼ Oracle Solaris OSのセキュリティー鍵を生成する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solarisのシステム管理\(セキュリティーサービス\)](#)』の「[RBACの構成\(作業マップ\)](#)」を参照してください。
- 2 共通エージェントコンテナの管理デーモンを停止します。

```
# /usr/sbin/cacaoadm stop
```
- 3 セキュリティー鍵を再生成します。

```
# /usr/sbin/cacaoadm create-keys --force
```
- 4 共通エージェントコンテナの管理デーモンを再起動します。

```
# /usr/sbin/cacaoadm start
```

注 - Oracle Sun Cluster ソフトウェアの場合は、クラスタ内のすべてのノードにこの変更を伝達する必要があります。

ファイルアクセスでのトラブルシューティング(タスク)

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。

この章でトラブルシュートするトピックの一覧は次のとおりです。

- 271 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)」
- 274 ページの「ファイルアクセスの問題を解決する」
- 274 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」のメッセージは、以下のどれかを意味します。

- 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。

間違っただ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。例:

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

間違っただ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。例:

```
$ 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、間違っただ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 次の点を確認します。

- 検索パスは正しいか
- 検索パスは、別バージョンのコマンドが見つかったほかの検索パスの前にリストされているか
- 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	注意事項
Bourne と Korn	<code>\$HOME/.profile</code>	<code>\$ PATH=\$HOME/bin:/sbin:/usr/local/bin ...</code> <code>\$ export PATH</code>	パス名はコロンの区切る

シェル	ファイル	構文	注意事項
C	\$HOME/.cshrc or \$HOME/.login	<i>hostname%</i> set path=(<i>~bin /sbin /usr/local/bin ...</i>)	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	.profile	\$./.profile
C	.cshrc	<i>hostname%</i> source .cshrc
	.login	<i>hostname%</i> source .login

- 5 新しいパスを確認します。

```
$ which command
```

例 19-1 検索パスの問題を診断して解決する

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 印刷デーモン) が見つからない場合、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

ファイルとディレクトリの所有権は、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、変更されることが頻繁にあります。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、またはユーザーの属するグループが `/etc/group` データベースから削除されたときにも発生します。

アクセスに問題のあるファイルのアクセス権または所有権を変更する方法については、『Solaris のシステム管理 (セキュリティサービス)』の第6章「ファイルアクセスの制御 (作業)」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に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に伴う問題、および `AutoFS` を通じたシステムへのアクセスでの問題については、『Solaris のシステム管理 (ネットワークサービス)』の「NFS のトラブルシューティングの方法」を参照してください。

UFS ファイルシステムの不整合解決 (タスク)

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応答できる内容について説明します。

注 - Solaris 10 6/06 リリースからは、fsck コマンドの実行時に表示されるエラーメッセージが変更されています。この章では、変更された fsck エラーメッセージが含まれます。Solaris 10 6/06 リリース以降が稼働していないシステム上で fsck コマンドを実行する場合に当てはまる情報については、「Oracle Solaris 10」と記載されているセクションのエラーメッセージを参照してください。今回のリリースで実施されたすべての fsck の改善点の詳細は、『Solaris のシステム管理 (デバイスとファイルシステム)』を参照してください。

この章の内容は次のとおりです。

- 277 ページの「fsck の一般エラーメッセージ」
- 279 ページの「初期化フェーズでの fsck メッセージ」
- 282 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」
- 288 ページの「フェーズ 1B: 走査し直して DUPS メッセージを表示する」
- **Oracle Solaris 10:** 287 ページの「Oracle Solaris 10: フェーズ 1B: 走査し直して DUPS メッセージを表示する」
- 288 ページの「フェーズ 2: パス名の検査のメッセージ」
- 295 ページの「フェーズ 3: 接続性の検査のメッセージ」
- 298 ページの「フェーズ 4: 参照数の検査のメッセージ」
- 301 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」
- **Oracle Solaris 10:** 302 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」
- 303 ページの「fsck サマリーメッセージ」
- **Oracle Solaris 10:** 304 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、それを使用してファイルシステムの整合性を検査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)』の第 20 章「UFS ファイルシステムの整合性検査 (手順)」を参照してください。

fsckのエラーメッセージ

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 2a - 重複している名前の検査
- フェーズ 2b - パス名の検査
- フェーズ 3 - 接続状態の検査
- フェーズ 3b - シャドウ/ACL の確認
- 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

この後の各セクションでは、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、[277 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ごとにアルファベット順で掲載されています。

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 20-1 エラーメッセージの省略形

省略形	意味
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間
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。このドキュメントでは、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

このセクション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 disk-block-number (CONTINUE)
```

Oracle Solaris 10:

```
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 *disk-block-number* への移動要求に失敗しました。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します。

Oracle Solaris 10: ファイルシステム内で、指定されたブロック番号 *block-number* への移動要求に失敗しました。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します。

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します。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します。

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、この問題は継続します。もう一度fsckを実行してファイルシステムをチェックします。

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てください。

CANNOT READ: DISK BLOCK *disk-block-number*: I/O ERROR
CONTINUE?

Oracle Solaris 10:

CANNOT READ: DISK BLOCK *block-number*: I/O ERROR
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 *disk-block-number* の読み取り要求に失敗しました。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します。

Oracle Solaris 10: ファイルシステム内で、指定されたブロック番号 *block-number* の読み取り要求に失敗しました。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します。

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ー番号のリストを表示します。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します。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ます。

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、この問題は継続します。もう一度fsckを実行してファイルシステムをチェックします。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てください。

CANNOT WRITE: BLK *disk-block-number* (CONTINUE)

Oracle Solaris 10:

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 *disk-block-number* への書き込み要求に失敗しました。

ファイルシステムのチェックを続けると、fsckは書き込みを再試行し、書き込みなかったセクタ番号のリストを表示します。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します。

Oracle Solaris 10: ファイルシステム内で、指定されたブロック番号 *block-number* への書き込み要求に失敗しました。

ファイルシステムのチェックを続けると、fsckは書き込みを再試行し、書き込みなかったセクタ番号のリストを表示します。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します。

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ります。ドライブ上で書き込み保護ロックをチェックします。ディスクにハードウェア障害がある場合、問題は解決しません。もう一度 fsck を実行してファイルシステムをチェックします。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、ご購入先に問い合わせてください。

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。

このセクションのメッセージは、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、ファイルのステータス、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、ロギングされている 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ませんでした。

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ます。この場合、fsckは、通常と同様の検査を実行し、フェーズ5で次の質問を表示します。

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します。次回にロギングを有効にしてファイルシステムをマウントすると、ログが再作成されます。

noと応答すると、ログを保持したまま終了しますが、ファイルシステムはマウントできません。

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ました。fsckは終了します。

対処方法

ご購入先に問い合わせてください。

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ました。fsckは終了します。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します。ほかのプロセスが大量のシステムリソースを使用していると、このエラー条件が発生することがあります。

対処方法

ほかのプロセスを終了すると問題を解決できることがあります。解決できない場合は、ご購入先に問い合わせてください。

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*) を開いて読み込めません。fsckは終了します。

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します。

Can't open *filename*

エラーの発生原因

fsckはファイルシステム *filename* を開けませんでした。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ます。

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します。

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ました。fsckは終了します。

対処方法

このメッセージは、重大なシステム障害を示します。ご購入先に問い合わせてください。

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ました。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ます。

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します。

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ませんでした。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ますが、fsckは何も修正しません。

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します。通常ファイル名の可能性があります。

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パーセント未満です。

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* と入力します。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* と入力します。

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ます。

対処方法

次のいずれかのメッセージが表示されます。

```

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 を再実行します。最初にブロック 32 を指定することをお勧めします。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ます。-N オプションは必ず指定してください。指定しないと、newfs は既存のファイルシステムを上書きします。

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 でもありません。

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 を入力します。このエラー条件を無視するには、n と入力します。

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* が入っています。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ります。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ます。

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ノードのステータスがすべて、ファイルシステムの損傷を示す1に設定されています。このメッセージは、fsck -y の実行後に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ありません。

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します。

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* を示します。fsck は即座に終了します。

対処方法

ご購入先に問い合わせてください。

fragment-number DUP I=*inode-number*

Oracle Solaris 10:

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 *fragment-number* が入っています。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ります。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ます。

Oracle Solaris 10: *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* が入っています。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ります。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ます。

対処方法
なし

DUP TABLE OVERFLOW (CONTINUE)

エラーの発生原因

fsck は、重複フラグメントを追跡するためのメモリーを割り当てることができませんでした。-o p (preen、修復) オプションが指定されていると、プログラムは終了します。

Oracle Solaris 10: 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ありません。-o p (preen、修復) オプションが指定されていると、プログラムは終了します。

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* と入力します。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ません。別の重複フラグメントが見つかると、このエラー条件が再発します。使用可能な仮想メモリーの容量を (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します。プログラムを終了するには *n* と入力します。

Oracle Solaris 10: プログラムを続行するには、CONTINUE プロンプトから *y* と入力します。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ません。別の重複ブロックが見つかると、このエラー条件が再発します。使用可能な仮想メモリーの容量を (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します。プログラムを終了するには *n* と入力します。

EXCESSIVE BAD FRAGMENTS I=*inode-number* (CONTINUE)

Oracle Solaris 10:

EXCESSIVE BAD BLOCKS I=*inode-number* (CONTINUE)

エラーの発生原因

無効なディスクアドレスを示しているフラグメント数が多すぎます (通常は 10 を超える数)。-o p (preen、修復) オプションを指定すると、プログラムは終了します。

Oracle Solaris 10: *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ます (通常は 10 を超える数)。-o p (preen、修復) オプションを指定すると、プログラムは終了します。

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* と入力します。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ません。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ります。プログラムを終了するには *n* と入力します。

EXCESSIVE DUP BLKSDUPLICATE FRAGMENTS I=*inode-number* (CONTINUE)

Oracle Solaris 10:

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フラグメント数が多すぎます (通常は 10 を超える数)。-o p (preen、修復) オプションが指定されていると、プログラムは終了します。

Oracle Solaris 10: 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ます (通常は 10 を超える数)。-o p (preen、修復) オプションが指定されていると、プログラムは終了します。

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* と入力します。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ません。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ります。プログラムを終了するには *n* と入力します。

INCORRECT DISK BLOCK COUNT I=*inode-number* (*number-of-BAD-DUP-or-missing-blocks* should be *number-of-blocks-in-filesystem*) (CORRECT)

Oracle Solaris 10:

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* のディスクブロック数が不正です。修復 (preen) の場合、fsck は数を訂正します。

Oracle Solaris 10: *i* ノード *inode-number* のブロック数は *number-of-BAD-DUP-or-missing-blocks* ですが、*number-of-blocks-in-filesystem* であるべきです。修復 (preen) の場合、fsck は数を訂正します。

対処方法

i ノード *inode-number* のディスクブロック数を *number-of-blocks-in-file* に修正するには、CORRECT プロンプトから *y* と入力します。

Oracle Solaris 10: 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* と入力します。プログラムを終了するには *n* と入力します。

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ありません。-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ります。

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* と入力します。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します。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ません。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ります。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 を実行し直します。プログラムを終了するには *n* と入力します。

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ありません。-o p (preen、修復) オプションを指定すると、この i ノードは消去されます。

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* と入力します。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ります。このエラー条件を無視するには、*n* と入力します。応答しなくてよいのは、この問題をほかの手段で解決しようとする場合のみです。

PARTIALLY TRUNCATED INODE I=*inode-number* (SALVAGE)

エラーの発生原因

fsck で、割り当てられたフラグメント数よりも短い i ノード *inode-number* が見つかりました。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のみです。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します。

Oracle Solaris 10: 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* が見つかりました。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のみです。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します。

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と入力します。このエラー条件を無視するには、nと入力します。

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、キャラクタ型デバイス、ブロック型デバイス、通常ファイル、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します。-o p (preen、修復) オプションを指定すると、このiノードは消去されます。

Oracle Solaris 10: 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します。-o p (preen、修復) オプションを指定すると、このiノードは消去されます。

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と入力します。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 が生成されます。このエラー条件を無視するには、nと入力します。

Oracle Solaris 10: フェーズ 1B: 走査し直して DUPS メッセージを表示する

このセクションでは、今回のリリースのフェーズ 1B の fsck メッセージについて説明します。

ファイルシステム内で重複フラグメントが見つかると、次のメッセージが表示されます。

```
fragment 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フラグメント番号 *fragment-number* が入っています。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ます。重複フラグメントを持つ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ます。これは、fsckの実行時に生成される重複フラグメントレポートを使うと簡単になります。

対処方法

重複ブロックが見つかったら、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*ノードが検索されます。

フェーズ 1B: 走査し直して DUPS メッセージを表示する

このセクションでは、Oracle Solaris 10 リリースおよびその他のサポート対象 Solaris リリースで表示される fsck メッセージについて説明します。

ファイルシステム内で重複ブロックが見つかったら、次のメッセージが表示されません。

```
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* が入っています。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ます。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ます。

対処方法

重複ブロックが見つかったら、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*ノードが検索されます。

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ます。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*ノードモードとステータス
- 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* が渡されました。fsckは終了します。

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てください。

```
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* が見つかりました。

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* と入力します。「.」のiノード番号を変更しない場合は、*n* と入力します。

```
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

ディレクトリ *inode-number* が見つかりましたが、このディレクトリの「..」のiノード番号が *inode-number* の親と等しくありません。

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* を入力します。(ルートiノード内の「..」は、それ自体を指す)。「..」のiノード番号を変更しない場合は、*n* と入力します。

```
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* が返されました。fsckは終了します。

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てください。

```
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* が割り当てられました。fsckは終了します。

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てください。

```
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* が割り当てられました。fsckは終了します。

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てください。

```
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* が見つかりました。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、およびディレクトリ名 *filename* が表示されます。

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* と入力します。このディレクトリを無視するには *n* と入力します。

```
DIRECTORY filename: LENGTH file-size NOT MULTIPLE OF disk-block-size (ADJUST)
```

Oracle Solaris 10:

```
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 *disk-block-size* の倍数でないディレクトリ *filename* が見つかりました。

Oracle Solaris 10:

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* が見つかりました。

対処方法

長さを適切なディスクブロックサイズに切り上げるには、*y* と入力します。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します。この条件を無視するには *n* と入力します。

Oracle Solaris 10:

長さを適切なブロックサイズに切り上げるには、*y* と入力します。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します。この条件を無視するには *n* と入力します。

```
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りました。

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トリを放棄するには、SALVAGE プロンプトから *y* と入力します。この処置によって、最高で42個のエントリを放棄できます。この処置は、他の回復作業に失敗した場合にのみ実行します。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* と入力します。

```
DUP/BAD I=inode-number OWNER=0 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トリ *filename*、*i* ノード *inode-number* に関連付けられた重複フラグメントまたは不良フラグメントが見つかりました。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、ディレクトリまたはファイル名 *filename* が表示されます。*-op* (*preen*、修復) オプションを指定すると、重複または不良フラグメントが削除されます。

Oracle Solaris 10:

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トリ *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りました。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、ディレクトリまたはファイル名 *filename* が表示されます。*-op* (*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ます。

対処方法

ディレクトリまたはファイルのエントリ *filename* を削除するには、REMOVE プロンプトから *y* と入力します。このエラー条件を無視するには、*n* と入力します。

```
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 (*i* ノード番号20) に、重複フラグメントまたは不良フラグメントが見つかりました。

Oracle Solaris 10:

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りました。

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* と入力します。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ます。ルートの割り当てに失敗すると、fsckは次のメッセージを表示して終了します。「CANNOT ALLOCATE ROOT INODE」。*n* と入力すると、CONTINUE プロンプトが表示されま

す。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チェックを続行するには、CONTINUE プロンプトに対して *y* と入力します。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ります。n の場合は、プログラムを終了します。

```
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

「.」のエントリが複数個入っているディレクトリ *inode-number* が見つかりました。

対処方法

「.」の余分なエントリを削除するには、FIX プロンプトから *y* と入力します。問題のディレクトリを変更しない場合は、n と入力します。

```
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

ディレクトリ *inode-number* が見つかりましたが、このディレクトリは複数個の「..」(親ディレクトリ)エントリを持っています。

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 プロンプトに *y* を入力します。問題のディレクトリを変更しない場合は、n と入力します。

```
hard-link-number IS AN EXTRANEIOUS 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* にエラーが見つかりました。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します。

対処方法

エラーのあるエントリ *hard-link-number* を削除するには、REMOVE プロンプトから *y* と入力します。このエラー条件を無視するには、n と入力します。

```
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* が付いています。-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ます。

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* と入力します。このエラー条件を無視するには、n と入力します。

```
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* が見つかりました。

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 と入力します。問題のディレクトリを変更しない場合は、n と入力します。

```
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* が見つかりました。fsckはこの問題を解決できません。

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てください。

```
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

最初のエントリが「.」でないディレクトリ *inode-number* が見つかりました。fsckは問題を解決できません。

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てください。

```
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のエントリが割り当てられていないディレクトリ *inode-number* が見つかりました。

対処方法

「..」を、i ノード番号が *inode-number* の親に等しくなるように構築するには、FIX プロンプトから y と入力します。(ルート i ノード内の「..」は、それ自体を指す)。問題のディレクトリを変更しない場合は、n と入力します。

```
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のエントリが *filename* となっているディレクトリ *inode-number* が見つかりました。fsckはこの問題を解決できません。

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てください。

```
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

ディレクトリ *inode-number* が見つかりましたが、このディレクトリの第2のエントリは「..」(親ディレクトリ)ではありません。fsckはこの問題を解決できません。

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てください。

```
NAME TOO LONG filename
```

エラーの発生原因

長すぎるパス名が見つかりました。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します。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ります。

対処方法

循環リンクを削除します。

```
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ありません。

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* と入力します。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ます。ルートの割り当てに失敗すると、fsck は次のメッセージを表示して終了します。「CANNOT ALLOCATE ROOT INODE」。プログラムを終了するには *n* と入力します。

```
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* ノードではありません。

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* と入力します。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ます。ルートの

割り当てに失敗すると、fsckは次のメッセージを表示して終了します。「CANNOT ALLOCATE ROOT INODE」。fsckにFIXプロンプトを表示させるには、nと入力します。

```
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* を指しています。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、およびファイル名 *filename* が表示されます。

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と入力します。このエラー条件を無視するには、nと入力します。

```
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になっています。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、およびディレクトリ名 *filename* が表示されます。

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と入力します。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ます。このエラー条件を無視するには、nと入力します。

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(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

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* が渡されました。fsckは終了します。

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てください。

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します。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ます。

対処方法

なし

DIRECTORY *filename* LENGTH *file-size* NOT MULTIPLE OF *disk-block-size* (ADJUST)

Oracle Solaris 10:

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* が見つかりました。(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ります。)

対処方法

長さを適切なディスク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* と入力します。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します。このエラー条件を無視するには、*n* と入力します。

Oracle Solaris 10:

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* と入力します。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します。このエラー条件を無視するには、*n* と入力します。

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ありません。

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* と入力します。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ます。*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ます。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ます。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* と入力します。

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` ディレクトリがありません。修復するときに、`fsck` は `lost+found` ディレクトリを作成しようとします。

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` と入力します。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ります。`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ます。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ます。消失 `i` ノードへのリンク試行を中止するには、`n` と入力します。

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ません。修復するときに、`fsck` は `lost+found` ディレクトリを拡張します。

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` と入力します。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ます。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ます。`lost+found` ディレクトリ内で不要なエントリを削除します。修復中にこのエラーが起きると、`fsck` は終了します。消失 `i` ノードへのリンク試行を中止するには、`n` と入力します。

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ませんでした。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`、および変更時間 `modification-time` が表示されます。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` ノードを接続し直します。それ以外の場合、`fsck` はディレクトリ `i` ノードを消去します。

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` と入力します。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ます。それ以外の場合には、`lost+found` エラーメッセージのいずれかが表示されます。このエラー条件を

無視するには、nと入力します。このエラーにより、フェーズ4でUNREFエラー条件が発生します。

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フラグメントまたは重複フラグメント

Oracle Solaris 10:

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フラグメントまたは不良フラグメントが見つかりました。iノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時間 *modification-time* が表示されます。

Oracle Solaris 10:

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りました。iノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時間 *modification-time* が表示されます。

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と入力します。このエラー条件を無視するには、nと入力します。

(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ません。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ません。

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* と入力します。直前のエラー条件を無視するには、*n* と入力します。

```
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* になっていますが、*corrected-link-count* でなければなりません。*i* ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時間 *modification-time* が表示されます。*-o p* (*preen*、修復) オプションを指定すると、参照数が増えない限り、リンク数が調整されます。この条件は、ハードウェア障害が存在する場合以外は発生しません。修復中に参照数が増加する場合、fsck は LINK COUNT INCREASING メッセージを表示して終了します。

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* と入力します。このエラー条件を無視するには、*n* と入力します。

```
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ありません。

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* と入力します。lost+found による以前の *i* ノード参照は消去されません。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ます。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ます。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ます。消失 *i* ノードへのリンク試行を中止するには、*n* と入力します。

```
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 ディレクトリがありません。修復するときに、fsck は lost+found ディレクトリを作成しようとします。

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` と入力します。`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ます。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ます。消失 `i` ノードへのリンク試行を中止するには、`n` と入力します。

`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ありません。修復するとき、`fsck` は `lost+found` ディレクトリを拡張します。

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` と入力します。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ます。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ます。`lost+found` ディレクトリ内で不要なエントリを削除します。修復 (`-op` オプション) が有効なときは、このエラーによって `fsck` が終了します。消失 `i` ノードへのリンク試行を中止するには、`n` と入力します。

`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ませんでした。`i` ノード `inode-number` の所有者 `UID`、モード `file-mode`、サイズ `file-size`、および変更時間 `modification-time` が表示されます。`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ます。それ以外の場合は再び接続されます。

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` と入力します。`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ります。このエラー条件を無視するには、`n` と入力します。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ます。

`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ませんでした。`i` ノード

inode-number の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時間 *modification-time* が表示されます。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ます。それ以外の場合は再び接続されます。

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* と入力します。このエラー条件を無視するには、*n* と入力します。

```
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 になっています。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時間 *modification-time*、およびディレクトリ名 *filename* が表示されます。

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* と入力します。このエラー条件を無視するには、*n* と入力します。

フェーズ 5: シリンダグループ検査のメッセージ

このセクションでは、Oracle Solaris 最新リリースのフェーズ 5 の fsck メッセージについて説明します。

このフェーズでは、空きフラグメントと使用済み 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
- 空きフラグメントマップから欠落している空きフラグメント
- 使用済み i ノードマップ内の空き i ノード
- 不正な合計空きフラグメント数
- 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 (アルファベット順) が表示される可能性があります。

```
FRAG BITMAP WRONG (CORRECTED)
```

エラーの発生原因

シリンダグループのフラグメントマップから空きフラグメントがいくつか欠落しています。修復中に、fsck はマップを作成し直します。

対処方法

空きフラグメントマップを作成し直すには、SALVAGE プロンプトから *y* と入力します。このエラー条件を無視するには、*n* と入力します。

CG *cg-number*: BAD MAGIC NUMBER

エラーの発生原因

シリンダグループ *cg-number* のマジック番号が間違っています。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します。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ます。ファイルシステムを修復している場合は、`fsck` が終了します。

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てください。

CORRECT GLOBAL SUMMARY (SALVAGE)

エラーの発生原因

サマリー情報が間違っています。修復していると、`fsck` はサマリー情報を計算し直します。

対処方法

サマリー情報を作成し直すには、`SALVAGE` プロンプトから `y` と入力します。このエラー条件を無視するには、`n` と入力します。

フェーズ 5: シリンダグループ検査のメッセージ

このセクションでは、Solaris 10 の最初の 3/05 リリースにおけるフェーズ 5 の `fsck` メッセージについて説明します。

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 (アルファベット順) が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ます。修復中に、`fsck` はマップを作成し直します。

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` と入力します。このエラー条件を無視するには、`n` と入力します。

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ます。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します。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ます。ファイルシステムを修復している場合は、fsckが終了します。

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てください。

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ません。-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ます。

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 と入力します。このエラー条件を無視するには、n と入力します。

SUMMARY INFORMATION BAD (SALVAGE)

エラーの発生原因

サマリー情報が間違っています。修復していると、fsckはサマリー情報を計算し直します。

対処方法

サマリー情報を作成し直すには、SALVAGE プロンプトから y と入力します。このエラー条件を無視するには、n と入力します。

fsck サマリーメッセージ

このセクションでは、Oracle Solaris 最新リリースの fsck サマリーメッセージについて説明します。Solaris 10 6/06 リリース以降が稼働していない場合、これらのメッセージはクリーンアップフェーズで表示されます。詳細については、[304 ページ](#)の「[クリーンアップ\(後処理\)フェーズのメッセージ](#)」を参照してください。

ファイルシステムのチェックが終わると、サマリーメッセージがいくつか表示されます。

```
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 fscck によって変更されたことを示します。このメッセージが表示された場合は、fscck を再び実行する必要はありません。このメッセージは、fscck の修正処置に関する情報を示すだけのものです。

クリーンアップ(後処理)フェーズのメッセージ

このセクションでは、Oracle Solaris 10 リリースの、クリーンアップフェーズの fscck メッセージについて説明します。現在のリリースでは、類似のメッセージが fscck サマリーフェーズで表示されます。詳細については、[303 ページの「fscck サマリーメッセージ」](#)を参照してください。

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、次のステータス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 fscck 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ブートします。ファイルシステムがマウントされている場合、マウント解除して再び fscck を実行する必要があることがあります。そうしないと、fscck 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る可能性があります。

```
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。

ソフトウェアパッケージで発生するトラブルシューティング(タスク)

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。「特定のソフトウェアパッケージのインストールエラー」セクションでは、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」セクションでは、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。

- 308 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」
- 309 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)』の第 20 章「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トラブルシューティング

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、このために、パッケージによってこの新しい `pkgadd` の動作に準拠する場合とし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. Oracle 提供のパッケージの場合は、[My Oracle Support](#) に電話して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します。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します。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します。しかし pkgadd は、そのためのアクセス権を持っていません。	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りません。詳細は、『Solaris のシステム管理 (基本編)』の第 20 章「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
<p>Solaris 2.5 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、既知の問題が存在します。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ますが、ユーザーの応答は無視されることがあります。</p>	<p>次の環境変数を設定して、パッケージを追加し直します。</p> <pre>NONABI_SCRIPTS=TRUE</pre>

索引

A

- acct.h 形式ファイル, 157, 158
- acctcms コマンド, 150, 161
- acctcom コマンド, 157, 158
- acctcon コマンド, 143, 150, 159
- acctdusg コマンド, 136, 154, 159
- acctprc コマンド, 150
- acctwtmp コマンド, 135, 137, 152
- active.MMDD ファイル, 145, 159
- active ファイル, 145, 159
- active ファイル, 149
- at.deny ファイル, 128, 132
 - 説明, 114
- atjobs ディレクトリ, 118
 - 説明, 114
- at コマンド, 127, 128, 132
 - l オプション (リスト), 130
 - m オプション (メール), 128, 129
 - エラーメッセージ, 132
 - 概要, 114, 115, 127
 - 自動スケジューリング, 118
 - 使用の拒否, 131-132
 - 使用の制御, 128, 132
 - 概要, 114
- at ジョブファイル, 127, 131
 - 書き込み, 127
 - 削除, 131
 - 作成, 128, 129
 - 説明, 115
 - 場所, 116
 - 表示, 130

C

- chargefee スクリプト, 137, 154
 - ユーザーへの課金, 142
- ckpacct スクリプト, 138, 140, 141
- closewtmp コマンド, 150
- cmsprev ファイル, 160
- consadm コマンド, 235-236
 - 補助コンソールのリストを表示する (方法), 236
 - 補助コンソールを無効にする, 237
 - 補助コンソールを有効にする, 235-236
 - システムリブート後, 236-237
- coreadm コマンド, 240
 - コアダンプ構成の表示, 242
 - コアファイルの管理, 240
 - コアファイル名パターンの設定, 243
- CPU (中央処理装置)
 - 情報の表示
 - 使用時間, 154, 172, 188
 - 大量に使用するプロセス, 188
- cron.allow ファイル, 123, 124, 125
- cron.deny ファイル, 123, 124
 - デフォルト, 123
- crontab コマンド, 124
 - cron デーモンと, 117
 - e オプション (編集), 119, 120
 - l オプション (リスト), 121
 - r オプション (削除), 122, 123
 - /var/adm 保守と, 228
 - エラーメッセージ, 126
 - 概要, 114, 115
 - 使用されるファイル, 117

crontab コマンド (続き)

- 使用の制御, 123, 124, 125
 - overview, 123
 - 概要, 114, 124
 - 使用の拒否, 123, 124
 - 使用を特定のユーザーに限定する, 123, 124, 125
- スケジューリング, 117
- によって実行されるアカウンティングスクリプト, 140, 142
- 変更を保存せずに終了, 120
- 毎日のタスク, 115

crontab ファイル

- 構文, 118, 119
- 削除, 122, 123
- 作成, 119, 120
- 作成と編集, 113-114
- 使用の拒否, 124
- 説明, 117, 118
- デフォルト, 117
- 場所, 117
- 表示, 121
- 編集, 119, 120

cron デーモン, 116, 117

ctacct.MMDD ファイル, 150, 159

ctmp ファイル, 159

D

- date コマンド, アカウンティングデータと, 135
- date コマンド, アカウンティングデータと, 137
- daytacct ファイル
 - runacct スクリプトと, 150, 161
 - /var/adm/acct/nite ディレクトリに設けられた, 160
 - 日次利用状況レポートと, 154
- df コマンド, 196
 - h オプション, 82
 - k オプション (キロバイト), 196
 - t オプション (ブロックの合計数), 83
 - 概要, 80, 196
 - 例, 81, 196
- disktacct.MMDD ファイル, 150
- disktacct ファイル, 138

- disktacct ファイル, 136, 150, 159
- dispadmin コマンド, 概要, 182
- dmesg コマンド, 228-229
- dodisk スクリプト, 136
 - 概要, 136, 138
 - 注意, 136
 - によって作成されるファイル, 136, 138, 150, 159
 - を実行する crontab エントリ, 142
- dtmp ファイル, 159
- DTrace 機能, 222
- dumpadm, システムクラッシュ情報の管理, 251
- du コマンド, 87, 89

E

edquota コマンド

- p オプション (プロトタイプ), 101
- t オプション (期間制限), 108
- 概要, 97, 98, 106
- 個々のユーザーの割り当て制限を無効にする, 110
- ユーザー割り当て制限を設定, 101

eeprom コマンド, ttymon 端末でのボーレートの設定に使用, 44

- /etc/acct/holidays ファイル, 141, 142
- /etc/cron.d/at.deny ファイル, 128, 132
- /etc/cron.d/cron.allow ファイル, 123, 124, 125
- /etc/cron.d/cron.deny ファイル, 123, 124
- /etc/init.d/acct ファイル, 141
- /etc/syslog.conf ファイル, 230
- /etc/utmpx ファイル, 40
- /etc/vfstab ファイル, 99

F

- fcntl 情報, 174, 176
- fd2log ファイル, 145, 149, 159
- fee ファイル, 137, 143, 150, 159
- find コマンド
 - コアファイル, 92
 - サイズ制限を超えるファイルの検索, 86
 - 古い/使用されていないファイル, 90
 - 古い/使用されていないファイル, 89

fiscriptn ファイル, 161
 fsck コマンド, 115
 fstat 情報, 174, 176

G

getty, 25
 GRUB ベースのブート
 SMF ブートアーカイブサービスの失敗のトラブルシューティング, 222
 システムクラッシュ
 失敗した SMF ブートアーカイブサービス, 249

H

holidays ファイル, 142
 hostid コマンド, 64

I

iostat コマンド
 概要, 193
 基本情報の表示, 194

K

klwp 構造体, 166
 kmdb ユーティリティ, 262
 kmdb ユーティリティー, 262–263
 kthread 構造体, 166

L

lastdate ファイル, 150, 159
 lastlogin コマンド, 150
 lineuse ファイル, 「/var/adm/acct/nite/lineuse ファイル」を参照
 localeadm コマンド, 60
 lock1 ファイル, 150

lock ファイル, 145, 150
 log.MMDD ファイル, 159
 loginlog ファイル, 150, 160, 161
 log ファイル, 159
 ls コマンド, 83, 84, 90
 -l オプション (バイト単位のサイズ), 84
 -s オプション (ブロック単位のサイズ), 85
 -t オプション (最新ファイル), 90
 ディレクトリサイズのチェック, 84
 LWP (軽量プロセス)
 構造体, 166
 情報の表示, 174
 定義, 165
 プロセスおよび, 165, 166

M

mdb ユーティリティー, 255, 256
 messages.n ファイル, 228
 messages ファイル, 230
 messages ファイル, 224
 monacct スクリプト
 runacct スクリプトと, 138, 149
 月次コマンドサマリーと, 155, 156
 実行する crontab エントリ, 141
 実行をスケジュールする, 140
 によって使用/生成されるファイル, 161
 MOTD (その日のメッセージ) 機能, 75
 motd ファイル, 75
 motd ファイル, 75

N

nice コマンド, 186, 188
 nice 値, 172, 186
 nlsadmin コマンド, 42–43

O

Oracle Solaris プロセスアカウントティングと統計の改善点, 133–134
 owtmptx ファイル, 160

P

pacctn ファイル
概要, 137, 150, 159
サイズの監視, 138, 149
表示, 157, 158

perf ファイル, 216

pfiles コマンド, 174, 176

pflags コマンド, 174

pkill コマンド, 174, 178

pldd コマンド, 174

pmadm コマンド
ttymon サービスの追加, 49
ttymon サービスの有効化, 53
ttymon サービスのリスト作成, 50
ttymon サービスを無効にする, 53
説明, 39

pmap コマンド, 174

prdaily スクリプト
runacct スクリプトと, 149, 161
回線の利用状況レポートと, 161
概要, 149
によって使用されるファイル, 159, 160

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), 185
-i オプション (ID タイプ), 184, 185
-l オプション (スケジューリングクラスの表示), 182
-m オプション (最高/最低の優先順位), 184
-p オプション (優先順位の指定), 184
-s オプション (優先順の上限/変更), 184
-s オプション (優先順の上限/優先順位の変更), 185
概要, 182

PROCFS (プロセスファイルシステム), 173

proc 構造体, 166, 172

proc ツール, コアファイルの検査, 244

/proc ディレクトリ, 173

prtconf コマンド, 64, 70
システムの製品名を表示する, 69-70

psig コマンド, 174

psrinfo コマンドオプションによるチップマルチスレッド化機能の識別, psrinfo -p, 60

pstack コマンド, 174

ps コマンド, 172, 175
-c オプション (スケジューリングクラス), 172, 188
-ecl オプション (グローバル優先順位), 183
-ef オプション (完全な情報), 175
概要, 172
報告されるフィールド, 172

ptacctn.MMDD ファイル, 152

ptime コマンド, 174

ptree コマンド, 174, 176

pwait コマンド, 174

pwdx コマンド, 174, 176

Q

quotacheck コマンド, 97, 102

quotaon コマンド, 97, 103

quotas ファイル, 97, 99

quot コマンド, 89

R

reboots ファイル, 150, 159

repquota コマンド, 104, 105-106, 106

rm コマンド, 91, 92

rprrt.MMDD ファイル, 138, 161

rpt.MMDD ファイル, 150, 160

RS-232-C, 「シリアルポート」を参照

runacct スクリプト, 144, 149
monacct スクリプトと, 149
prdaily スクリプトと, 149, 161
エラー保護, 149, 150
エラーメッセージ, 145
概要, 138
壊れたファイルの修復, 143, 144, 150
再起動, 145, 146, 150, 152
最後に実行された時間, 159
実行をスケジュールする, 140
失敗, 145
状態, 150
診断ファイル, 149
進捗ファイル, 149
によって使用/生成されるファイル, 159

runacct スクリプト (続き)
 によって使用/生成されるファイル, 161
 ユーザー料金の計算と, 142, 154
 を実行する crontab エントリ, 149

S

sa1 コマンド, 216
 sa2 コマンド, 216, 217
 SAC, 「サービスアクセスコントローラ」を参照
 sacadm コマンド, 48
 ttymon ポートモニターの起動, 47
 ttymon ポートモニターの追加, 45
 ttymon ポートモニターの停止, 47
 説明, 38
 sadc コマンド, 216, 217
 sadd ファイル, 217
 SAF, 「サービスアクセス機能」を参照
 SAFを使用したシリアルポートの管理, タスク
 マップ, 36
 sar コマンド, 199, 217
 -A オプション (全体のパフォーマンス), 216,
 218
 -a オプション (ファイルアクセス), 199, 200
 -b オプション (バッファ), 200
 -c オプション (システムコール), 202
 -e オプション (終了時間), 218
 -f オプション (データを抽出するファイ
 ル), 218
 -i オプション (間隔), 218
 -m オプション (プロセス間通信), 207
 -p オプション (ページイン/ページ障害), 208
 -q オプション (待ち行列), 209, 210
 -r オプション (未使用のメモリー), 210
 -s オプション (開始時間), 218
 -u オプション (CPU 使用状況), 211
 -v オプション (システムテーブル), 212
 -y オプション (端末デバイス), 214
 オプションのリスト, 218
 概要, 199, 217
 すべてのオプションの説明, 218
 shutacct スクリプト, 137, 139
 shutdown コマンド, 139
 Spacctn.MMDD ファイル, 150, 159

startup コマンド, acct, 137
 statefile ファイル, 145, 150, 159
 svcadm enable system/sar:default コマンド, 216
 sys crontab, 217
 syslog.conf ファイル, 230
 syslogd デーモン, 227

T

tacct.MMDD ファイル, 145, 150, 160
 tacctn ファイル, 161
 tacctprev ファイル, 160
 tacct ファイル, 145, 150, 160, 161
 /tmp/disktacct.MMDD ファイル, 150
 tmpwtmp ファイル, 150, 159, 161
 ttyadm コマンド, 42
 ttymon コンソールの端末でのボーレートの設定,
 方法, 44-45
 ttymon サービス
 追加する, 49
 表示する, 50
 無効にする, 53
 有効化, 53
 ttymon ポートモニター, 48
 起動, 47
 (図), 39
 追加, 45
 停止, 47
 発着信両用モデムサービス, 40
 tty 回線
 回線のトラブルシューティング, 153
 tty 回線, 利用状況の監視, 152
 tty 回線
 利用状況の監視, 135, 153, 161
 turnacct switch スクリプト, 137, 138
 turnacct switch スクリプト, 150

U

UFS quota コマンド, 98
 UFS ファイルシステム, 情報の表示, 89
 UFS 割り当て制限, 105-106, 107-108
 個々のユーザーに対して無効にする, 110

UFS 割り当て制限 (続き)

個々のユーザーに対する変更, 109
 コマンド, 104
 チェック, 104
 要件, 98
 弱い制限値の期間
 変更, 108

UFS 割り当て制限の使用, 95-96
 UFS 割り当て制限の初期化, 98
 UNIX システム (クラッシュ情報), 250
 /usr/adm/messages ファイル, 224
 /usr/bin/mdb ユーティリティ, 255
 /usr/proc/bin ディレクトリ, 173, 174
 utmp2wtmp コマンド, 150

V

/var/adm/acct/fiscal ディレクトリ, 159
 /var/adm/acct/nite/active.MMDD ファイル, 149, 159
 /var/adm/acct/nite/active ファイル, 145, 149, 159
 /var/adm/acct/nite/cms ファイル, 150
 /var/adm/acct/nite/cms ファイル, 159
 /var/adm/acct/nite/ctacct.MMDD ファイル, 150, 159
 /var/adm/acct/nite/ctmp ファイル, 159
 /var/adm/acct/nite/daycms ファイル, 150, 159, 161
 /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, 150
 /var/adm/acct/nite/disktacct ファイル, 138
 /var/adm/acct/nite/disktacct ファイル, 136, 138, 150, 159
 /var/adm/acct/nite/fd2log ファイル, 145, 149, 159
 /var/adm/acct/nite/lastdate ファイル, 150, 159
 /var/adm/acct/nite/lineuse ファイル, 150, 159, 161
 /var/adm/acct/nite/lock1 ファイル, 150
 /var/adm/acct/nite/lock ファイル, 145, 150, 159
 /var/adm/acct/nite/log.MMDD ファイル, 159
 /var/adm/acct/nite/log ファイル, 159
 /var/adm/acct/nite/owtmpx ファイル, 160
 /var/adm/acct/nite/reboots ファイル, 150, 159

/var/adm/acct/nite/statefile ファイル, 145, 150, 159
 /var/adm/acct/nite/tmpwtmp ファイル, 150, 159, 161
 /var/adm/acct/nite/wtmp.MMDD ファイル, 150, 160
 /var/adm/acct/nite/wtmperror.MMDD ファイル, 159
 /var/adm/acct/nite/wtmperror ファイル, 159
 /var/adm/acct/nite ディレクトリ, 159
 /var/adm/acct/sum/cmsprev ファイル, 160
 /var/adm/acct/sum/cms ファイル, 150
 /var/adm/acct/sum/cms ファイル, 160, 161
 /var/adm/acct/sum/daycms ファイル, 150, 160, 161
 /var/adm/acct/sum/loginlog ファイル, 150, 160, 161
 /var/adm/acct/sum/rprt.MMDD ファイル, 161
 /var/adm/acct/sum/rprtMMDD ファイル, 138
 /var/adm/acct/sum/rprt.MMDD ファイル, 150
 /var/adm/acct/sum/tacct.MMDD ファイル, 150, 160
 /var/adm/acct/sum/tacctMMDD ファイル, 145
 /var/adm/acct/sum/tacctprev ファイル, 145, 160
 /var/adm/acct/sum/tacct ファイル, 150
 /var/adm/acct/sum/tacct ファイル, 145, 160, 161
 /var/adm/acct/sum ディレクトリ, 138, 159, 160
 /var/adm/acct ディレクトリ, 159
 /var/adm/dtmp ファイル, 159
 /var/adm/fee ファイル, 137, 143, 150, 159
 /var/adm/messages.n ファイル, 228
 /var/adm/messages ファイル, 224, 230
 /var/adm/sa/sadd ファイル, 217
 /var/adm/Spacctn.MMDD ファイル, 150, 159
 /var/adm ディレクトリ
 raw アカウンティングデータ, 137
 サイズの制御, 91
 説明, 159
 /var/spool/cron/atjobs ディレクトリ, 114, 116, 118
 /var/spool/cron/crontabs/root ファイル, 116, 136
 /var/spool/cron/crontabs/sys crontab, 217
 /var/spool/cron/crontabs ディレクトリ, 117
 vfstab ファイル, 割り当て制限および, 99

vmstat コマンド

- 概要, 190
- レポートのフィールド, 190

W

- Watchdog reset! メッセージ, 227
- wtmp.MMDD ファイル, 150, 160
- wtmperror.MMDD ファイル, 159
- wtmperror ファイル, 159
- wtmpfix コマンド, 143, 150, 159
- wtmpx ファイル, 144
 - 概要, 137, 143, 150
 - 壊れたファイルの修復, 143, 144, 150
 - シャットダウンと, 139
 - 日次レポートと, 152

X

- x86 ベースのシステムのリブート, ブートアーカイブ SMF サービスが失敗する, 222

Z

- ZFS, スワップ領域およびダンプボリュームの要件, 249

あ

- アカウンティング, 144, 146, 161
 - 「ユーザーへの課金」も参照
- raw データ, 137
- 維持する, 146
- 概要, 134
- 壊れたファイルの修復
 - tacct ファイル, 145
 - wtmpx ファイル, 143, 144, 150
- 自動的に実行するよう設定する (方法), 141
- 接続, 135
 - runacct の状態と, 150
 - /var/adm/acct/nite/directory と, 159

アカウンティング, 接続 (続き)

- /var/adm/wtmpx, 153
- タイプ, 142
- 停止, 146-147
- ディスク, 136, 138
 - acctdusg プログラム, 154
- 日次, 137, 161
 - 「アカウンティング、レポート」も参照
 - 手順のサマリー, 139
- ファイル, 159, 161
- ブート, 141
- プロセス, 136, 137, 153, 154
- 無効化, 147-148
- ユーザー料金計算, 137
 - 「ユーザーへの課金」も参照
- レポート, 152
 - 概要, 152
 - 合計コマンドサマリー (月次), 156, 160, 161
 - 最終ログインレポート, 156
 - 日次コマンドサマリー, 155, 161
 - 日次利用状況レポート, 153, 154
 - 日次レポート (tty 回線の利用状況), 152, 153
- アダプタボード (シリアルポート), 24
- アドレス空間マップ, 174
- アプリケーションスレッド, 165, 167

い

- 一時ディレクトリ, 89, 92
- 印刷, ユーザー料金の計算, 142

え

- エラーメッセージ
 - at コマンド, 132
 - crontab コマンド, 126
 - runacct スクリプト, 145
 - 格納場所の指定, 227, 230
 - クラッシュ関連の, 227
 - クラッシュメッセージ, 228
 - の記録をカスタマイズする, 230
 - の送信元, 230
 - の優先順位, 231

エラーメッセージ (続き)

- のログファイル, 224

- ログファイル, 227

か

カーネルスレッド

- 構造体, 166, 172

- スケジューリングと, 172

回線制御, 39

回線の利用状況

- /var/adm/acct/nite/lineuse ファイル, 162

- 接続アカウントと, 135

- 日次レポートと, 152

回線の利用状況の監視, 153

確認

- UFS 割り当て制限, 104

- 割り当て制限, 108

カスタマイズ

- システムのメッセージ記録, 230

- システムのメッセージ記録 (方法), 232

き

共通エージェントコンテナ

- Oracle Solaris OS でのトラブルシューティング, 221-222

- トラブルシューティング, 268-270

共通エージェントコンテナ共有コンポーネント

- ポート番号 (チェック方法), 269

問題の種類

- スーパーユーザーパスワードに関連するセキュリティ, 268-270

- ポート番号の衝突, 268-270

共有共通エージェントコンテナ, 共有コンポーネント, 268-270

共有メモリー, プロセスの仮想メモリー, 167

く

クラッシュ, 230, 265

- クラッシュ後のリブートの失敗, 259-260

クラッシュ (続き)

- クラッシュダンプ情報の保存, 250

- クラッシュダンプの検査, 255, 256

- 購入先と, 224, 250

- 対処方法, 265

- 他のシステム情報を保存する, 228

- によって生成されたシステム情報の表示, 227, 256

- の後の対処方法, 224

- クラッシュダンプ情報の保存, 250

- 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, 256

- 繰り返されるシステムタスク, 124

- グローバルコアファイルのパス, coreadm を使用した設定, 240

グローバル優先順位

- 定義, 182

- 表示, 183

け

- 警告メッセージの優先順位 (syslogd), 231

- 月次コマンドサマリー, 156

検索

- および古い/使用されていないファイルの削除「削除」を参照

- サイズ制限を超えるファイル, 86

- サイズの大きなファイル, 85

- 検索パス, 設定するためのファイル, 272

こ

- コアダンプ構成, coreadm を使用した表示, 242

コアファイル

- coreadm を使用した管理, 240

コアファイル

- proc ツールを使用した検査, 244

- 検索と削除, 93

コアファイル

- 自動的に削除する, 128

- コアファイルの検査, proc ツールを使用した, 244

- コアファイル名パターン, coreadm で設定, 241

- 合計コマンドサマリー, 156, 160

購入先

クラッシュ情報の送信, 224

クラッシュダンプ解析, 250

コマンド, 利用状況の監視, 159

コマンドが見つかりませんエ

ラーメッセージ, 271

コンソール

補助

システムリブート後も有効にする, 236-237

コンソールの端末, ボーレートの設定方法, 44-45

コンソールの端末のボーレート, eeprom コマンド

を使用した設定, 44

さ

サービスアクセス機能

概要, 26, 37

関連プログラム(表), 37

によって制御されるサービス

の状態(表), 56

用途, 26, 37

サービスアクセスコントローラ, 38, 39

再開, プロセス, 174

再起動

runacct スクリプト, 145, 146, 150, 152

最終ログインレポート, 156

最小, nice 値, 186

サイズ

ディレクトリ, 87, 89

ファイル, 83, 84, 87, 89

サイズの大きなファイル, 85

最大

nice 値, 186

最大サイズを超えるファイルを見つける, 86

削除

at ジョブ, 131

crontab ファイル, 122, 123

一時ファイル, 92

コアファイル, 93

古い/使用されていないファイルの検索と削

除, 89

古い/未使用のファイル, 115

ログファイル, 120

作成

at ジョブ, 128

at ジョブ, 129

crontab ファイル, 119, 120

し

時間

CPU 時間が大量に増えているプロセス, 188

CPU 使用, 154, 172

CPU の使用, 188

システムアカウントिंग, タスク

マップ, 139-140

システムクラッシュ情報, dumpadm を使用し

た, 251

システムクラッシュ情報の管理, dumpadm を使用し

た, 251

システムクラッシュのトラブルシューティング

GRUB

リブート時にブートアーカイブサービスが

失敗する, 249

システムタスク

「crontab コマンド、at コマンド」も参照

自動的にスケジュールを設定する, 114

スケジューリング

1 回限りのタスク, 115, 127

繰り返されるタスク, 115, 116

システムタスクの自動実行, 114

1 回限りのタスク, 127, 128, 132

繰り返されるタスク, 124, 125

システム動作

追跡されるシステム動作のリスト, 167

データの自動収集, 216

データの手動収集, 217

システム動作データの自動収集, 216

システム動作レポートの自動生成, 216, 217

システムの製品名, prtconf コマンドを使用して表

示する, 69-70

システムの物理プロセッサタイプ, psrinfo

-p, 70-71

システムのメッセージ記録(カスタマイズ), 230

システムメッセージ

格納場所の指定, 227

ログのカスタマイズ(方法), 232

システムリソース

アカウントティング

概要, 134

概要, 165

監視, 128

UFS 割り当て制限, 106

アカウントティング, 146

アカウントティングシステム, 161

クラッシュ, 230, 265

自動, 128

システムリソースの管理, ロードマップ, 59

失敗した SMF ブートアーカイブサービス, GRUB

ベースのブートのトラブルシューティング, 249

失敗した SMF ブートアーカイブサービスのトラブルシューティング

x86

GRUB のフェイルセーフアーカイブ, 263-264

失敗した x86 ベースのシステムのリブート, SMF
ブートアーカイブサービス, 222

シャットダウン

監視, 137, 139, 152

修復, 144

壊れた tacct ファイル, 145

壊れた wtmpx ファイル, 143, 144

終了, プログラムを強制終了, 265

状態, (runacct スクリプト), 150

シリアルポート

アダプタボード, 24

定義, 24

新機能

CPU パフォーマンスカウンタ, 164

svcadm enable system/sar:default コマンド, 216

強化された pfiles ツール, 163

す

スーパーユーザーパスワードに関連するセキュリティー

共通エージェントコンテナ共有コンポーネント
トラブルシューティング, 268-270スーパーユーザー (ルート) パスワード、忘れた
x86, 262スーパーユーザー (ルート) パスワードを忘れた,
SPARC, 261

スケジューリング

「crontab コマンド、at コマンド」も参照

1 回限りのシステムタスク, 115, 127

繰り返されるシステムタスク, 115, 116

スケジューリングクラス, 181

指定, 184

情報の表示, 172, 182, 183

変更, 185

優先順位の変更, 184, 186

優先順位レベルと, 182, 184

スワップボリュームのサイズ, ZFS ルートファイル
システムを使用するシステムの, 249

せ

制御

使用 at コマンド, 114, 128, 132

使用 crontab コマンド, 124, 125

概要, 114

プロセス, 177

制御がきかなくなったプロセス, 188

製品名情報を表示する, prtconf コマンド, 69-70

セキュリティ

at コマンド, 128

crontab コマンド, 124

接続アカウントティング, 「アカウントティング、接
続」を参照

設定, coreadm を使用したコアファイル名パターンの, 243

そ

その日のメッセージ (MOTD) 機能, 75

ソフトウェアパッケージ, のインストールに伴う
トラブルシューティング, 307

た

- 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, 184
- 優先順位
 - 概要, 182
 - 範囲, 182
 - 変更, 184, 186
- ダンプボリュームのサイズ, ZFS ルートファイルシステムを使用するシステムの, 249
- 端末, 31
 - 回線のトラブルシューティング, 153
 - 回線の利用状況
 - /var/adm/acct/nite/lineuse ファイル, 162
 - 接続アカウントと, 135
 - 日次レポートと, 152, 153
 - シリアルポートツール, 28
 - シリアルポートツールの項目の解説, 28
 - タイプ間の相違, 23
 - 定義, 23
 - プロセス制御, 172
 - 文字端末, 23
- 端末とモデムの設定, タスクマップ, 27-28

ち

- 着信専用モデムサービス, 23
- チューニング, 日次コマンドサマリーと, 155

つ

- ツール
 - システムパフォーマンス監視, 167
 - プロセス, 174
 - プロセス情報の表示, 173

て

- 定型タスクを自動的に実行する (概要), 114
- 停止
 - システムアカウント, 146-147
 - プロセスを一時的に, 174

ディスクアカウンティング, 「アカウント
ング、ディスク」を参照

ディスクドライブ

- 情報の表示
 - 空きディスク容量, 196
 - 古い/使用されていないファイルの検索と削除, 120

ディスクブロックとファイル制限, 相違点, 96

ディスク容量

- サイズ制限を超えるファイルの検索, 86
- サイズの大きなファイルを見つける, 85

情報の表示

- df コマンド, 196
- ディレクトリサイズ, 87, 89
- ファイルサイズ, 83, 84, 87
- マウントポイント, 196
- ユーザーの所有するディスク容量, 89
- 古い/使用されていないファイルの検索と削除, 89
- 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, 93

ディレクトリ

- 一時, クリア, 89
- 一時, クリア, 92
- サイズ, 87, 89
- 情報の表示, 83, 84, 87, 89
- プロセスの現在の作業ディレクトリ, 174

デフォルト

- nice 値, 186
- その日のメッセージ, 75
- 弱い制限値の期間, 108
- 割り当て制限の, 107-108

電源異常からの回復, 152

電源の再投入, 265

と

トラブルシューティング

- tty 回線, 153
- 共通エージェントコンテナ, 221-222
- 共通エージェントコンテナ共有コンポーネント問題の種類, 268-270
- ソフトウェアパッケージのインストールまたは削除, 307

トラブルシューティング (続き)

プロセス, 188

トラブルシューティングのタスク, 参照

先, 223-224

に

日次アカウントिंग, 「アカウントिंग、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, 274

ネットワークアクセスで発生する問題の把握, 274

は

パスワードセキュリティの衝突、スーパーユーザー、共通エージェントコンテナ, 268-270

発信専用モデムサービス, 23

発着信両用サービス, 23

発着信両用モデムサービス, 40

パニックメッセージ, 227

パフォーマンス

監視ツール, 167

システム動作の監視, 167, 199, 216

追跡される動作, 167

動作データの自動収集, 216

動作データの手動収集, 199, 217

ファイルアクセス, 199, 200

プロセス管理, 165, 174, 186

レポート, 199

ひ

表示

acct.h 形式ファイル, 157, 158

at ジョブ, 130

表示 (続き)

coreadm を使用したコアダンプ構成の, 242

crontab ファイル, 121

LWP 情報, 174

pacctn ファイル, 157, 158

クラッシュ情報, 227, 256

システム情報

コマンド, 64, 70

システム動作情報, 199, 217

システムにインストールされているメモリー, 70

実行中のプロセス, 175

スケジューリングクラス情報, 172, 182, 183

ディスク割り当て制限の情報, 97

ディレクトリ情報, 83, 84, 87

日付と時間, 70

ファイルシステム情報, 89

ファイル情報

du コマンドを使用, 87

最新をリスト, 90

ファイルサイズ, 83, 85

ファイルのサイズ, 84

ブートメッセージ, 228-229

プロセス, 175

プロセス情報 (方法), 176-177

ホスト ID, 69

優先順位情報, 172, 183

リンクされたライブラリ, 174

割り当て制限, 104-105

割り当て制限情報, 104, 105

ふ

ファイル

fstat および fcntl 情報の表示, 174, 176

fstat と fcntl 情報の表示, 174

アカウントिंग, 159, 161

アクセス操作のチェック, 199, 200

検索パスを設定するための, 272

壊れたファイルの修復

wtmpx ファイル, 150

サイズ, 83, 84, 87, 89

サイズ制限を超えるファイルを見つける, 86

サイズの表示, 84

ファイル (続き)

削除

「削除」を参照

使用状況の監視, 136

情報の表示

サイズ, 83, 84, 87, 89

リスト, 83, 84

利用状況の監視, 154

ファイルシステム

回復, 142

ディスク容量の使用状況, 196

復元, 154

マウントポイント, 196

ファイルまたはグループの所有権, ファイルアクセスの問題の解決, 274

ブート

中に生成されたメッセージの表示, 228-229

ブート時の `sadc` コマンドの実行, 216

ブートアーカイブ, リポート時の SMF サービスの失敗, 222

ブートアーカイブサービスが失敗した場合の対処
x86

フェイルセーフアーカイブの

ブート, 263-264

ブートアーカイブサービスの失敗

x86

GRUB のトラブルシューティング, 263-264

復元, 対応コマンドを使用する, 267

フラグの追跡, 174

フルクラッシュダンプディレクトリからの復元, 256

プログラム

ディスク依存度, 200

の実行を強制終了, 265

割り込み, 265

プログラムの割り込み, 265

プログラムを強制終了, 265

プロセス

`nice` 値, 172, 186, 188

`proc` ツールコマンド, 173

`proc` ツールコマンドを使用した情報の表示, 174

`proc` ツールを使用した情報の表示, 173

`trees`, 174

プロセス (続き)

アカウントिंगユーティリティー, 136, 137, 153, 154

アドレス空間マップ, 174

アプリケーションスレッドおよび, 165, 167

一時的に停止, 174

現在の作業ディレクトリ, 174, 176

構造体, 166, 172

再開, 174

シグナル動作, 174

終了, 174, 178

情報の表示, 172

`acctcom` コマンド, 157, 158

`LWP`, 174

`priocntl` コマンド, 182

`ps` コマンド, 172, 175, 183

実行中のプロセスの表示, 175

終了したプロセス, 157

日次利用状況レポート, 153, 154

プロセスの表示, 175

情報の表示 (方法), 176-177

スケジューリングクラス, 181

指定, 184

情報の表示, 172, 182, 183

変更, 185

優先順位の変更, 184, 186

優先順位レベルと, 182, 184

スタックトレース, 174

制御, 177

制御がきかない, 188

ツールコマンド, 174

ツリー, 174, 176

定義, 165

トラブルシューティング, 188

開いているファイルの `fstat` と `fcntl` の情報, 174, 176

フラグの追跡, 174

優先順位, 186

概要, 182, 186

グローバル優先順位, 182, 183

指定, 184

指定する, 184

情報の表示, 172, 183

スケジューリングクラスおよび, 182

プロセス, 優先順位 (続き)
スケジューリングクラスと, 184
タイムシェアリングプロセスの優先順位の変更, 184, 186
変更, 184, 186
ユーザーモードの優先順位, 182
用語, 165, 167
リンクされたライブラリ, 174
プロセスアカウンティング, 136, 137, 153, 154
理由レコード, 139
プロセスの終了, 174, 178
プロセスファイルシステム (PROCFS), 173
プロセス別コアファイルのパス, `coreadm` を使用した設定, 240

へ 変更

`crontab` ファイル, 119
個々のユーザーの UFS 割り当て制限, 109
システムのホスト名, 75-76
スケジューリングクラス, 185
その日のメッセージ, 75
日付, 74
優先順位, 184, 186
タイムシェアリングプロセス, 186
弱い制限値の期間, 108

編集

`crontab` ファイル, 119, 120

ほ

ポート, 32-33
初期化プロセス, 40
定義, 24
の状態(表), 57
ポート番号 (チェック方法)
共通エージェントコンテナ共有コンポーネント
`cacao`, 269
ポート番号の衝突
共通エージェントコンテナ共有コンポーネント
トラブルシューティング, 268-270

ポートモニター
`ttymon` と `listen` (定義), 41-43
`ttymon` と `listen` (定義), 25
定義, 24
の状態(表), 56

ポーレート

`eeeprom` コマンドを使用した設定方法, 44
`ttymon` 端末での設定方法, 44-45
補助 (リモート) コンソール, 233
ホスト名, 変更, 75-76

ま

毎週のタスク (`crontab` を使用したスケジューリング), 115
毎月のタスク (`crontab` を使用したスケジューリング), 115
毎日のタスク (`crontab` を使用したスケジューリング), 115

む

無効化, システムアカウンティング, 147-148
無効にする
個々のユーザーの割り当て制限, 110
補助コンソールを `consadm` コマンドで, 237

め

メモリー

仮想

プロセス, 167

共有

プロセスの仮想メモリー, 167

情報の表示例, 70

に関する情報を表示するコマンド, 64
プロセス構造体および, 166

も

文字端末, 「端末」を参照

モデム, 31-32

- さまざまな使用方法, 23
- シリアルポートツールの概要, 28
- シリアルポートモデムのモデム用テンプレート, 29
- 着信専用サービス, 23
- 定義, 23
- 発信専用サービス, 23
- 発着信両用サービス, 23, 40

ゆ

有効にする

- システムリブート後の補助コンソール, 236-237

補助コンソールを `consadm` コマンドで, 235-236

ユーザー構造体, 166

ユーザーのディスク容量の所有状況, 89

ユーザープロセス

- CPU 使用, 154
- 優先順位, 182
- 優先順位の変更, 186

ユーザーへの課金, 142

「`chargefee` スクリプト」も参照

ユーザーモードの優先順位, 182

ユーザー料金, 137, 154

「ユーザーへの課金」も参照

ユーザーログイン

- 最終ログインの監視, 150, 156, 161
- 時間の監視, 135, 150, 154
- ログイン回数, 154

ユーザー割り当て制限, 104-105

- 個々のユーザーに対して無効にする, 110
- 個々のユーザーに対する変更, 109
- 設定, 101

優先順位(プロセス)

- 概要, 182, 186
- グローバル
 - 定義, 182
 - 表示, 183
- 指定, 184
- 情報の表示, 172, 183
- スケジューリングクラスと, 184
- 変更, 184, 186

優先順位(プロセス), 変更(続き)

- タイムシェアリングプロセス, 184, 186
- ユーザーモードの優先順位, 182

よ

- 要件, UFS 割り当て制限, 98
- 弱い制限値の期間, 変更, 107-108

り

リアルタイムプロセス, クラスの変更, 185

リスト

ファイルとディレクトリ, 83, 84, 90

リブート

- クラッシュ後の失敗, 259-260
- 接続アカウンティングと, 135
- と `/var/adm/wtmpx` ファイル, 137
- 日次レポートと, 152

リモート印刷, ユーザー料金の計算, 142

理由レコード, プロセスアカウンティング, 139

料金(ユーザー), 154

料金, ユーザー, 137, 142

る

ルート `crontab` ファイル, 136

ルートパスワードの回復

- SPARC, 261
- x86, 262

ルートパスワード, 忘れた

- x86, 262
- GRUB ベースのブート, 262

ルートパスワードを忘れた

- SPARC, 261
- x86, 262

ろ

ログイン監視

最終ログイン, 150, 156, 161

ログイン監視 (続き)

使用時間, 135, 138, 154

ログイン回数, 154

ログファイル, 自動的に削除, 120

わ

忘れた、ルートパスワードを, SPARC, 261

割り当て制限

概要, 95

確認, 97, 104, 108

削除, 106

情報を表示する, 104

整合性の検査, 102

設定, 97, 102

超過をチェック, 104-105

強い制限値の設定, 96

表示, 104-105

複数ユーザーのプロトタイプ, 101

変更, 106

有効化と無効化, 98

有効にする, 97

有効にする、例, 103

ユーザー

個々のユーザーに対する変更, 109

設定, 101

超過をチェック, 105

ユーザー割り当て制限の超過をチェック, 105

弱い制限値の設定, 96

割り当て制限、UFS

使用, 95-96

初期化, 98

ファイルシステムのチェック, 106

弱い制限値のデフォルトの変更, 107-108

割り当て制限の設定, 102

割り当て制限を自動的に有効にする, 97

割り当て制限を無効にする, 98

割り当て制限を有効にする, 98

割り当て制限を有効にする、例, 103